

Curso de Vue JS

SYS4SOFT (João Ribeiro)

https://www.youtube.com/watch?v=fnPXT1z0Ctw&list=PLXik_5Br-zO_xQHAB9GrNR1gAefYWaKxz

Resumo do curso feito por Roberto Pinheiro

Aula 01 - Introdução

- Como editor de textos utilize o Visual Studio Code.
- Instale o plugin Live Server e o utilize como servidor.

aula-01/index.html

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Olá Mundo com Vue JS</title>
</head>
<body>

  <div id="app">
    <h2>Olá mundo! Meu nome é {{nome}}.</h2>
  </div>

  <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
  <script src="../js/script.js"></script>
</body>
</html>
```

aula-01/js/script.js

```
const app = new Vue({
  el: "#app",
  data: {
    nome: "Roberto"
  }
});
```

The screenshot shows a browser window with the title "Ola Mundo com Vue JS". The address bar indicates the page is at `127.0.0.1:5500/aula-01/index.html`. The developer tools console tab is active, showing the following output:

```
Download the Vue Devtools extension for a better development experience:  
https://github.com/vuejs/vue-devtools  
You are running Vue in development mode.  
Make sure to turn on production mode when deploying for production.  
See more tips at https://vuejs.org/guide/deployment.html  
> app.nome  
< "Roberto"
```

Aula 05 - A instância Vue JS

aula-05/index.html

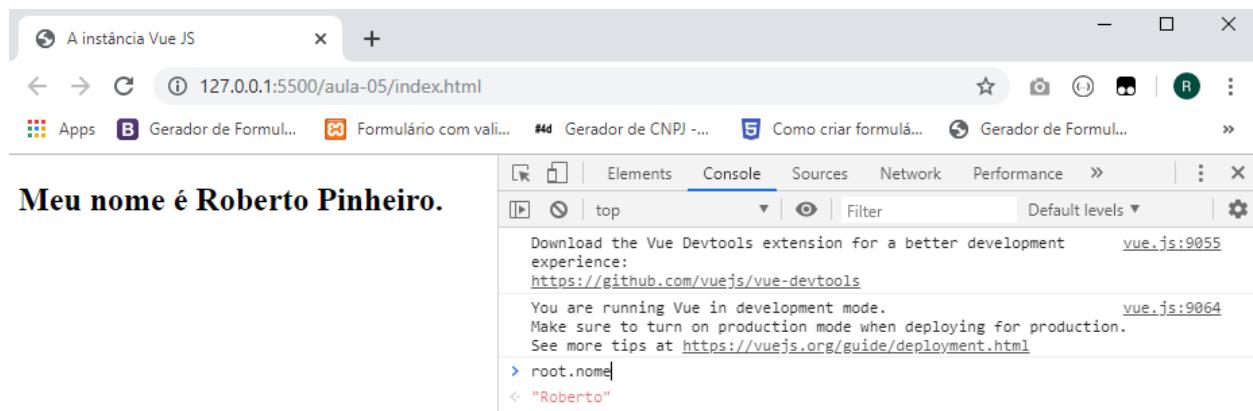
```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>A instância Vue JS</title>
</head>
<body>

  <div id="root">
    <h2>Meu nome é {{nome}} {{sobrenome}}.</h2>
  </div>

  <script src="../js/vue.js"></script>
  <script src="js/script.js"></script>
</body>
</html>
```

aula-05/js/script.js

```
const root = new Vue({
  el: "#root",
  data: {
    nome: "Roberto",
    sobrenome: "Pinheiro"
  }
});
```



A instância Vue JS

127.0.0.1:5500/aula-05/index.html

Apps Gerador de Formul... Formulário com vali... Gerador de CNPJ ... Como criar formulá... Gerador de Formul...

Meu nome é João Ribeiro.

Elements Console Sources Network Performance

Download the Vue Devtools extension for a better development experience: [vue.js:9055](https://github.com/vuejs/vue-devtools)

You are running Vue in development mode. Make sure to turn on production mode when deploying for production. See more tips at <https://vuejs.org/guide/deployment.html> [vue.js:9064](#)

```
> root.nome = "João"
< "João"
> root.sobrenome = "Ribeiro"
< "Ribeiro"
```

Aula 06 - A instância Vue JS

aula-06/index.html

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Introdução a diretivas</title>
</head>
<body>

  <div id="root">
    <h2>{{nome}}</h2>
    <input type="text" v-model="nome"/>
    <div v-if="visivel">
      <h4>DIV 1</h4>
    </div>
    <div v-else>
      <h4>DIV 2</h4>
    </div>
  </div>

  <script src="../js/vue.js"></script>
  <script src="js/script.js"></script>
</body>
</html>
```

aula-06/js/script.js

```
const root = new Vue({
  el: "#root",
  data: {
    nome: "Roberto",
    visivel: false
  }
});
```

A screenshot of a browser window titled "Introdução a diretivas". The address bar shows the URL "127.0.0.1:5500/aula-06/index.html". The page content displays a form with two input fields: one containing "Roberto" and another containing "DIV 2". The developer tools' "Console" tab is selected, showing the following output:

```
Download the Vue Devtools extension for a better development experience: https://github.com/vuejs/vue-devtools
You are running Vue in development mode. Make sure to turn on production mode when deploying for production. See more tips at https://vuejs.org/guide/deployment.html
```

A screenshot of a browser window titled "Introdução a diretivas". The address bar shows the URL "127.0.0.1:5500/aula-06/index.html". The page content displays a form with two input fields: one containing "João" and another containing "DIV 1". The developer tools' "Console" tab is selected, showing the following output:

```
Download the Vue Devtools extension for a better development experience: https://github.com/vuejs/vue-devtools
You are running Vue in development mode. Make sure to turn on production mode when deploying for production. See more tips at https://vuejs.org/guide/deployment.html
> root.nome = "João"
< "João"
> root.visivel = true
< true
```

Aula 07 - Introdução aos métodos

aula-07/index.html

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <link rel="stylesheet" href="../css/bootstrap.min.css">
    <title>Introdução aos métodos</title>
</head>
<body>
    <div id="root">
        <div class="container">
            <br>
            <button class="btn btn-primary" v-on:click="mostrarEsconder()">Click</button>
            <br><br>
            <div v-if="visivel">
                <h4>DIV 1</h4>
            </div>
            <div v-else>
                <h4>DIV 2</h4>
            </div>
        </div>
    </div>
    <script src="../js/vue.js"></script>
    <script src="js/script.js"></script>
</body>
</html>
```

aula-07/js/script.js

```
const root = new Vue({
    el: "#root",
    data: {
        visivel: true
    },
    methods: {
        mostrarEsconder: function(){
            this.visivel = !this.visivel;
            console.log(this.visivel);
        }
    }
});
```

Clicando no botão azul:

The screenshot shows a browser window with the URL `127.0.0.1:5500/aula-07/index.html`. At the top, there are two buttons labeled "Click". The left button is highlighted with a blue border, while the right one is green. Below the buttons, the text "DIV 1" is displayed. The browser's developer tools are open, specifically the "Console" tab. The console output includes:

```
Download the Vue Devtools extension for a better development experience: https://github.com/vuejs/vue-devtools
You are running Vue in development mode. Make sure to turn on production mode when deploying for production. See more tips at https://vuejs.org/guide/deployment.html
```

Clicando nos dois botões:

The screenshot shows a browser window with the URL `127.0.0.1:5500/aula-07/index.html`. The two buttons from the previous screenshot are now both green. The text "Seja bem-vindo!" is displayed above "DIV 1". The browser's developer tools console tab is open, showing:

```
false
true
```

Clicando novamente nos dois botões:

The screenshot shows a browser window with the URL `127.0.0.1:5500/aula-07/index.html`. The two buttons are now both green. The text "DIV 2" is displayed below them. The browser's developer tools console tab is open, showing:

```
false
true
false
```

Aula 08 - Exemplos com diretivas v-for

aula-08/index.html

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <link rel="stylesheet" href="../css/bootstrap.min.css">
  <title>Diretiva v-for</title>
</head>
<body>

  <div id="root">
    <ul>
      <li v-for="nome in nomes">{{nome}}</li>
    </ul>
    <br>
    <ul>
      <li v-for="cliente in clientes">O cliente chama-se {{cliente.nome}} e tem {{cliente.idade}} anos de idade.</li>
    </ul>
  </div>

  <script src="../js/vue.js"></script>
  <script src="js/script.js"></script>
</body>
</html>
```

aula-08/js/script.js

```
const root = new Vue({
  el: "#root",
  data: {
    nomes: [
      "João",
      "Ana",
      "Carlos",
      "Maria",
      "Francisco",
      "Solange",
      "Antônio",
      "Sandra"
    ],
  }
})
```

```
clientes: [
    { nome: "João", idade: 54},
    { nome: "Ana", idade: 46},
    { nome: "Carlos", idade: 23},
    { nome: "Maria", idade: 55},
    { nome: "Francisco", idade: 32},
    { nome: "Solange", idade: 17},
    { nome: "Antônio", idade: 29},
    { nome: "Sandra", idade: 37},
]
});
```



-
- João
 - Ana
 - Carlos
 - Maria
 - Francisco
 - Solange
 - Antônio
 - Sandra
-
- O cliente chama-se João e tem 54 anos de idade.
 - O cliente chama-se Ana e tem 46 anos de idade.
 - O cliente chama-se Carlos e tem 23 anos de idade.
 - O cliente chama-se Maria e tem 55 anos de idade.
 - O cliente chama-se Francisco e tem 32 anos de idade.
 - O cliente chama-se Solange e tem 17 anos de idade.
 - O cliente chama-se Antônio e tem 29 anos de idade.
 - O cliente chama-se Sandra e tem 37 anos de idade.

Aula 09 - Exemplo prático com diretivas e métodos

aula-09/index.html

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <link rel="stylesheet" href="../css/bootstrap.min.css">
    <title>Diretivas e métodos</title>
</head>
<body>

    <div id="root">
        <div class="container">
            <br>
            <input v-model="novaFruta">
            <button v-on:click="inserirFruta()">Inserir</button>

            <h3>Lista de frutas</h3>

            <ul>
                <li v-for="fruta in frutas">{{fruta}}</li>
            </ul>

            <h4>Total: <b>{{totalFrutas}}</b></h4>
        </div>
    </div>

    <script src="../js/vue.js"></script>
    <script src="js/script.js"></script>
</body>
</html>
```

aula-09/js/script.js

```
let root = new Vue({
    el: "#root",
    data: {
        frutas: [],
        novaFruta: '',
        totalFrutas: 0
    },
    methods: {
        inserirFruta() {
            this.frutas.push(this.novaFruta);
            this.novaFruta = '';
            this.totalFrutas = this.frutas.length;
        }
    }
});
```

```
methods: {
  inserirFruta: function(){
    this.frutas.push(this.novaFruta);
    this.novaFruta = '';
    this.totalFrutas = this.frutas.length;
  }
});
});
```

A screenshot of a web browser window. The title bar says "Diretivas e métodos" and the address bar shows "127.0.0.1:5500/aula-09/index.html". Below the address bar are navigation icons and tabs for "Apps", "Gerador de Formul...", and "Formulário com vali...". The main content area contains a text input field and a button labeled "Inserir". Below this is the heading "Lista de frutas" and the text "Total: 0".

A screenshot of a web browser window, identical to the one above but with different content. The main content area now displays a bulleted list of fruits: "banana", "laranja", and "maçã". Below this is the heading "Lista de frutas" and the text "Total: 3".

Aula 10 - Diretivas v-bind e v-on

aula-10/index.html

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <link rel="stylesheet" href="../css/bootstrap.min.css">
  <title>Diretivas v-bind e v-on</title>
</head>
<body>

  <div id="root">
    <div class="container">
      <br>
      <h3>O país selecionado é {{pais}}</h3>

      <img :src = "imagem" :alt = "nome_imagem">
      <!-- <img v-bind:src = "imagem"> -->

      <br /><br />

      <div>
        <button class="btn btn-danger" @click="mostrarPortugal()">Portugal</button>
        <button class="btn btn-success" @click="mostrarBrasil()">Brasil</button>
      <!--
        <button class="btn btn-danger" v-on:click="mostrarPortugal()">Portugal</button>
        <button class="btn btn-success" v-on:click="mostrarBrasil()">Brasil</button>
      -->
      </div>
    </div>
  </div>

  <script src="../js/vue.js"></script>
  <script src="js/script.js"></script>
</body>
</html>
```

aula-10/js/script.js

```
let root = new Vue({
  el: "#root",
  data: {
    pais: "",
    imagem: "",
    nome_imagem: ""
  },
  methods: {
    mostrarPortugal() {
      this.pais = "Portugal"
      this.nome_imagem = "portugal"
      this.imagem = require("../img/portugal.jpg")
    },
    mostrarBrasil() {
      this.pais = "Brasil"
      this.nome_imagem = "brasil"
      this.imagem = require("../img/brasil.jpg")
    }
  }
})
```

```

methods: {
    mostrarPortugal: function(){
        this.pais = "Portugal";
        this.nome_imagem = "Bandeira de Portugal";
        this.imagem = "images/bandeira_portugal.jpg";
    },
    mostrarBrasil: function(){
        this.pais = "Brasil";
        this.nome_imagem = "Bandeira do Brasil";
        this.imagem = "images/bandeira_brasil.jpg";
    }
},
};

root.mostrarPortugal();

```

Diretivas v-bind e v-on
 127.0.0.1:5500/aula-10/index.html
 Apps Gerador de Formul... Formulário com vali... Gerador de CNPJ ... Como criar formulá... Gerador c



[Portugal](#) [Brasil](#)

Diretivas v-bind e v-on
 127.0.0.1:5500/aula-10/index.html
 Apps Gerador de Formul... Formulário com vali... Gerador de CNPJ ... Como criar formulá... Gerador c

O país selecionado é Brasil



[Portugal](#) [Brasil](#)

Aula 11 - Diretivas v-html e v-text

aula-11/index.html

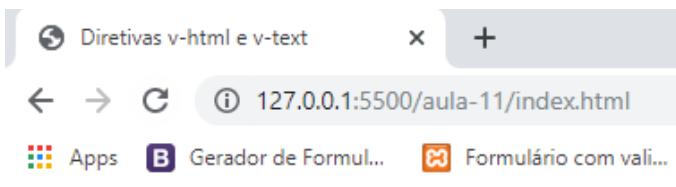
```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <link rel="stylesheet" href="../css/bootstrap.min.css">
  <title>Diretivas v-html e v-text</title>
</head>
<body>

  <div id="root">
    <div class="container">
      <br>
      <span v-html="valor"></span>
      <button class="btn btn-primary" @click="t1">Título 1</button>
      <button class="btn btn-danger" @click="t2">Título 2</button>
      <button class="btn btn-success" @click="t3">Título 3</button>
    </div>
  </div>

  <script src="../js/vue.js"></script>
  <script src="js/script.js"></script>
</body>
</html>
```

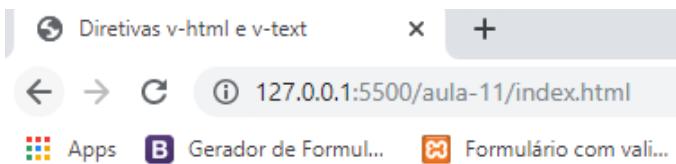
aula-11/js/script.js

```
let root = new Vue({
  el: "#root",
  data: {
    valor: '<p>Título Vue</p>'
  },
  methods: {
    t1: function(){
      this.valor = '<h1>Título Vue</h1>';
    },
    t2: function(){
      this.valor = '<h2>Título Vue</h2>';
    },
    t3: function(){
      this.valor = '<h3>Título Vue</h3>';
    }
});
```



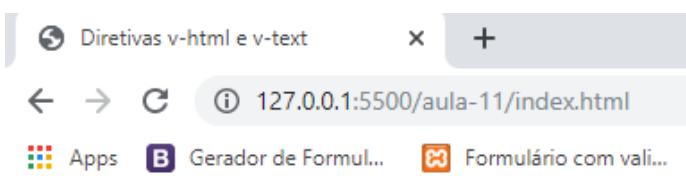
Título Vue

Título 1 Título 2 Título 3



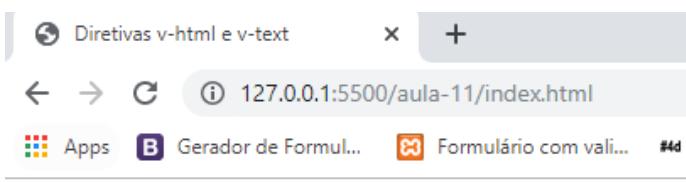
Título Vue

Título 1 Título 2 Título 3



Título Vue

Título 1 Título 2 Título 3



Título Vue

Título 1 Título 2 Título 3

Aula 12 - Diretivas v-once e v-pre

aula-12/index.html

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <link rel="stylesheet" href="../css/bootstrap.min.css">
  <title>Diretivas v-once e v-pre</title>
</head>
<body>

  <div id="root">
    <div class="container">
      <br>
      <p v-pre>{{Este é o texto dentro de double mustache}}</p>
      <p v-once>{{nome}}</p>
      <p v-text="nome"></p>
      <button class="btn btn-success" @click="executar">Click</button>
    </div>
  </div>

  <script src="../js/vue.js"></script>
  <script src="js/script.js"></script>
</body>
</html>
```

aula-12/js/script.js

```
let root = new Vue({
  el: "#root",
  data: {
    nome: "João"
  },
  methods: {
    executar: function(){
      this.nome = "Bruno";
    }
  }
});
```

S Diretivas v-once e v-pre X +

← → C ⓘ 127.0.0.1:5500/aula-12/index.html

_apps Gerador de Formul... Formulário com vali... #4d

{{Este é o texto dentro de double mustache}}

João

João

Click

S Diretivas v-once e v-pre X +

← → C ⓘ 127.0.0.1:5500/aula-12/index.html

_apps Gerador de Formul... Formulário com vali... #4d

{{Este é o texto dentro de double mustache}}

João

Bruno

Click

Aula 13 - Diretivas v-show e v-if

aula-13/index.html

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <link rel="stylesheet" href="../css/bootstrap.min.css">
  <title>Diretivas v-show e v-if</title>
</head>
<body>

<div id="root">
  <div class="container">
    <br />
    <p v-show="visivel" >Primeiro</p>
    <p v-if="visivel">Segundo</p>
    <br>
    <button class="btn btn-success" @click="executar">Click</button>
  </div>
</div>

<script src="../js/vue.js"></script>
<script src="js/script.js"></script>
</body>
</html>
```

aula-13/js/script.js

```
let root = new Vue({
  el: "#root",
  data: {
    visivel: true
  },
  methods: {
    executar: function() {
      this.visivel = !this.visivel;
    }
  }
});
```

The screenshot shows a browser window with the URL `127.0.0.1:5500/aula-13/index.html`. The page content includes:

```
<!doctype html>
<html lang="pt-br">
  <head>...</head>
  ...<body> == $0
    <div id="root">
      <div class="container">
        ::before
        <br>
        <p>Primeiro</p>
        <p>Segundo</p>
        <br>
        <button class="btn btn-success">Click</button>
        ::after
      </div>
    </div>...
```

The screenshot shows the same browser window after a click on the 'Click' button. The 'Primeiro' paragraph is now hidden, as indicated by the `display: none;` style applied to its `p` element. The browser's developer tools Elements tab shows the updated DOM structure.

```
<!doctype html>
<html lang="pt-br">
  <head>...</head>
  ...<body> == $0
    <div id="root">
      <div class="container">
        ::before
        <br>
        <p style="display: none;">Primeiro</p>
        <!-->
        <br>
        <button class="btn btn-success">Click</button>
        ::after
      </div>
    </div>...
```

Aula 14 - Introdução aos filters

aula-14/index.html

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <link rel="stylesheet" href="../css/bootstrap.min.css">
  <title>Introdução aos Filters</title>
</head>
<body>

  <div id="root">
    <div class="container">
      <br />
      <input v-model="novoNome">
      <br /><br />
      <div>
        <button class="btn btn-success" @click="adicionar">Adicionar</button>
      </div>
      <hr />
      <ul>
        <li v-for="nome in nomes">{{nome | capitalizar}}</li>
      </ul>
      <hr />
      <ul>
        <li v-for="nome in nomes">{{nome}}</li>
      </ul>
    </div>
  </div>

  <script src="../js/vue.js"></script>
  <script src="js/script.js"></script>
</body>
</html>
```

aula-14/js/script.js

```
let root = new Vue({
  el: "#root",
  data: {
    novoNome: '',
    nomes: []
  },
})
```

```
methods: {
  adicionar: function(){
    this.nomes.push(this.novoNome);
    this.novoNome = ' ';
  }
},
filters: {
  capitalizar: function(value){
    return value.toUpperCase();
  }
}
});
```

The screenshot shows a web browser window with the title "Introdução aos Filters". The address bar displays the URL "127.0.0.1:5500/aula-14/index.html". The page content includes a form field, a green "Adicionar" button, and two lists of names. The first list, labeled "CARLOS", contains the names CARLOS, JOÃO, ANA, ROBERTO, and MARIA. The second list, labeled "carlos", contains the names carlos, João, Ana, roberto, and maria. This demonstrates the use of the uppercase filter on the first list.

Introdução aos Filters

127.0.0.1:5500/aula-14/index.html

Apps Gerador de Formul... Formulário com vali...

Adicionar

- CARLOS
- JOÃO
- ANA
- ROBERTO
- MARIA

- carlos
- João
- Ana
- roberto
- maria

Aula 15 - Computed properties

aula-15/index.html

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <link rel="stylesheet" href="../css/bootstrap.min.css">
  <title>Introdução aos Filters</title>
</head>
<body>

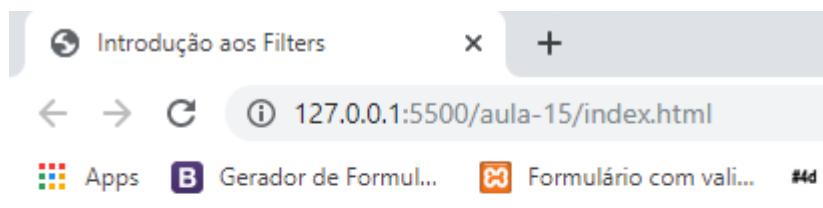
<div id="root">
  <div class="container">
    <br />
    <input v-model="novoNome">
    <br /><br />
    <div>
      <button class="btn btn-success" @click="adicionar">Adicionar</button>
    </div>
    <hr />
    <ul>
      <li v-for="nome in nomes">{{nome | capitalizar}}</li>
    </ul>
    <hr />
    <ul>
      <li v-for="nome in nomes">{{nome}}</li>
    </ul>

  </div>
</div>

<script src="../js/vue.js"></script>
<script src="js/script.js"></script>
</body>
</html>
```

aula-15/js/script.js

```
let root = new Vue({  
  el: "#root",  
  data: {  
    nome: 'João',  
    sobrenome: 'Ribeiro'  
  },  
  methods: {  
    nomeCompletoM: function(){  
      return this.nome + " " + this.sobrenome;  
    }  
  },  
  computed: {  
    nomeCompletoC: function(){  
      return this.nome + " " + this.sobrenome;  
    }  
  }  
});
```



O meu nome é João Luis Dias Ribeiro

O meu nome é João Luis Dias Ribeiro

João Luis Dias

Aula 16 - Introdução a custom components

aula-16/index.html

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <link rel="stylesheet" href="../css/bootstrap.min.css">
  <title>Introdução a custom components</title>
</head>
<body>

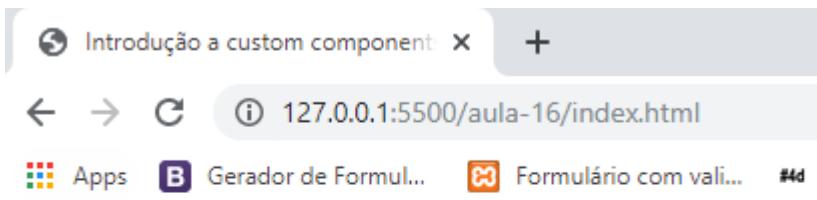
  <div id="root">
    <div class="container">
      <br />
      <novo></novo>
      <novo />
    </div>
  </div>

  <script src="../js/vue.js"></script>
  <script src="js/script.js"></script>
</body>
</html>
```

aula-16/js/script.js

```
Vue.component('novo', {
  template: `
    <div>
      <p>Olá Mundo!</p>
      <h1>Título 1</h1>
      <h2>Título 2</h2>
      <hr />
    </div>
  `,
})
```

```
let root = new Vue({
  el: "#root",
  component: [
    'novo'
  ]
});
```



Olá Mundo!

Título 1

Título 2

Olá Mundo!

Título 1

Título 2

Aula 17 - Components dentro de components

aula-17/index.html

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <link rel="stylesheet" href="../css/bootstrap.min.css">
  <title>Components dentro de components</title>
</head>
<body>

  <div id="root">
    <div class="container">
      <br />
      <pai></pai>
    </div>
  </div>

  <script src="../js/vue.js"></script>
  <script src="js/script.js"></script>
</body>
</html>
```

aula-17/js/script.js

```
Vue.component('pai', {
  template: `
    <div>
      <p>Este é o texto do componente pai</p>
      <filho></filho>
    </div>
  `

})

Vue.component('filho', {
  template: `
    <div>
      <p>Este é o texto do componente filho</p>
    </div>
  `

})
```

```
let root = new Vue({  
  el: "#root",  
  component: [  
    'pai',  
    'filho'  
  ]  
});
```

The screenshot shows a browser window with the title "Components dentro de componente". The address bar indicates the URL is 127.0.0.1:5500/aula-17/index.html. Below the address bar, there are navigation icons and a tab labeled "Gerador de Formul...". The main content area displays two lines of text: "Este é o texto do componente pai" and "Este é o texto do componente filho".

Este é o texto do componente pai
Este é o texto do componente filho

Aula 18 - Introdução a props

aula-18/index.html

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <link rel="stylesheet" href="../css/bootstrap.min.css">
  <title>Introdução a props</title>
</head>
<body>

  <div id="root">
    <div class="container">
      <br />
      <lista-nomes :nomescomponent = 'nomes'></lista-nomes>
    </div>
  </div>

  <script src="../js/vue.js"></script>
  <script src="js/script.js"></script>
</body>
</html>
```

aula-18/js/script.js

```
Vue.component('lista-nomes', {
  props: [
    'nomescomponent'
  ],
  template:
`


  <li v-for="nome in nomescomponent">{{nome}}</li>
</ul>
`,

})

let root = new Vue({
  el: "#root",
  data: {
    nomes: ['João', 'Ana', 'Carlos']
  }
});
```

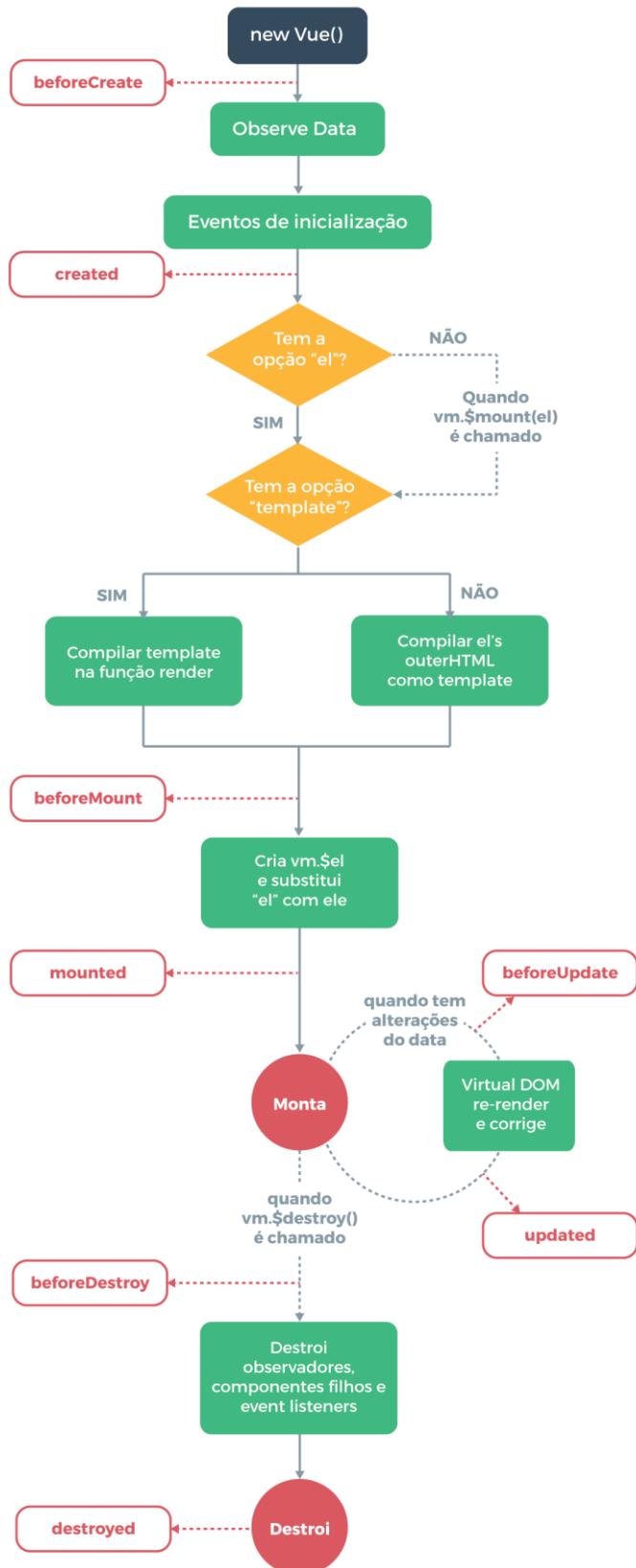
Introdução a props x +

← → ⌂ ⓘ 127.0.0.1:5500/aula-18/index.html

_apps Gerador de Formul... Formulário com vali...

- João
- Ana
- Carlos

Aula 19 - Ciclo de vida da instância Vue



aula-19/index.html

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <link rel="stylesheet" href="../css/bootstrap.min.css">
  <title>Ciclo de vida da instância Vue</title>
</head>
<body>

  <div id="root">
    <div class="container">
      <br />
      <p>{{nome}}</p>
      <button class="btn btn-primary" @click="click">Click</button>
      <button class="btn btn-danger" @click="destruir">Destruir</button>
    </div>
  </div>

  <script src="../js/vue.js"></script>
  <script src="js/script.js"></script>
</body>
</html>
```

aula-19/js/script.js

```
let root = new Vue({
  el: "#root",

  data: {
    nome: ""
  },

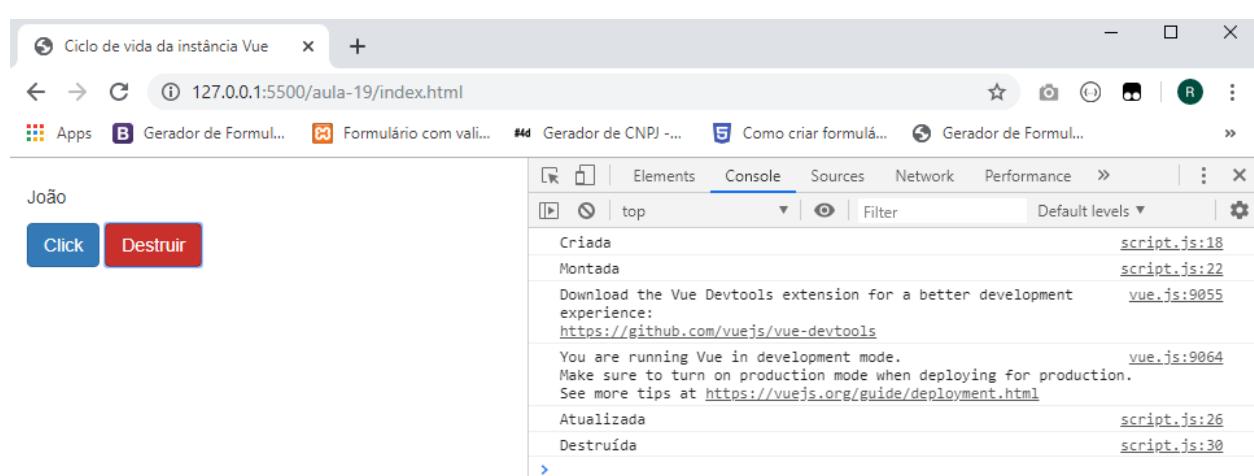
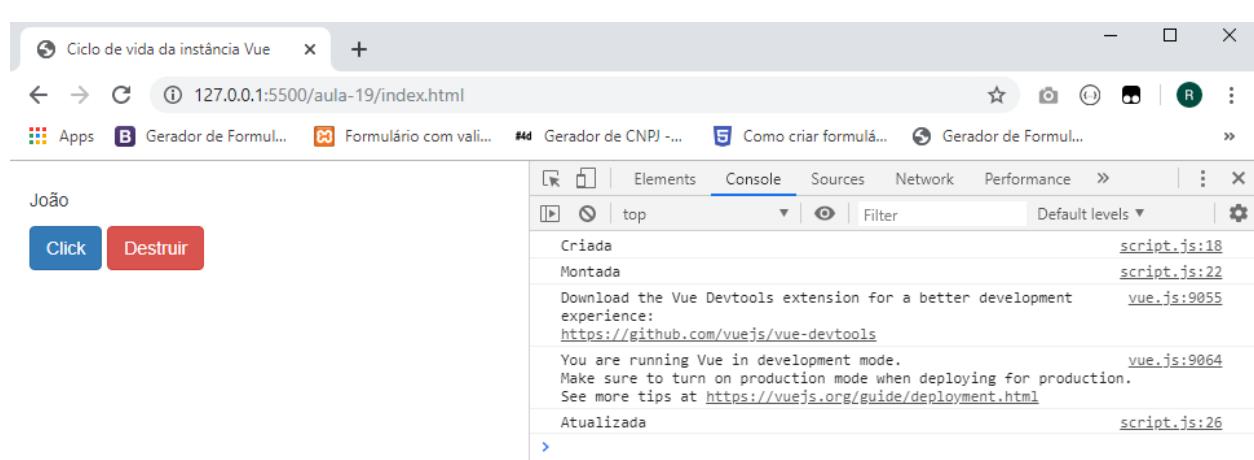
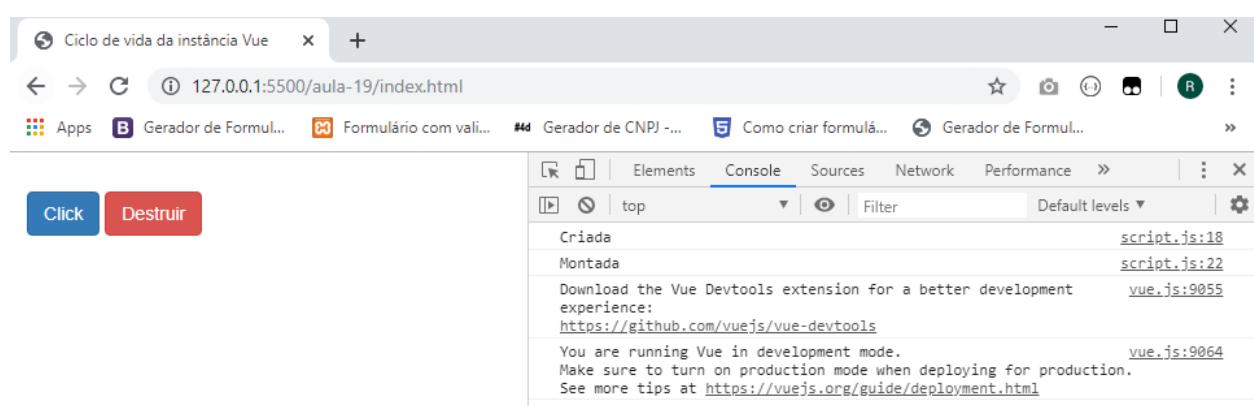
  methods: {
    click: function(){
      this.nome = 'João';
    },
    destruir: function(){
      root.$destroy();
    }
  },

  created: function(){
    console.log('Criada');
  }
},
```

```

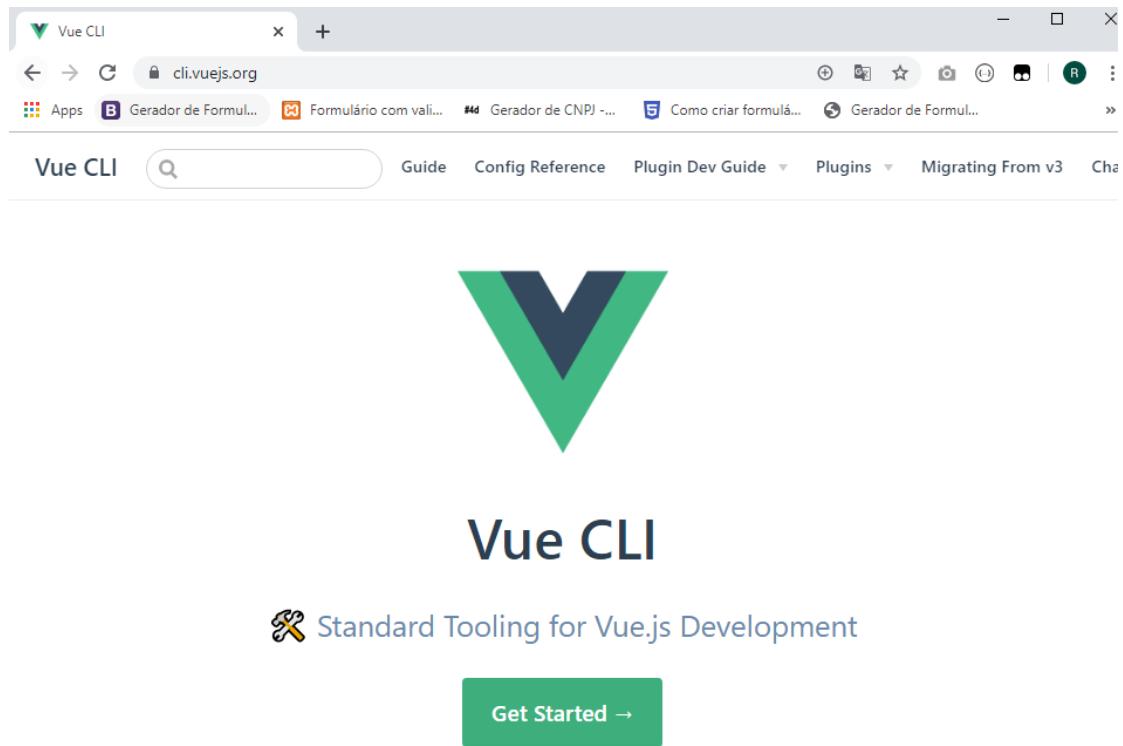
mounted: function(){
    console.log('Montada');
},
updated: function(){
    console.log('Atualizada');
},
destroyed: function(){
    console.log('Destruída');
}
);

```



Aula 20 - Instalação da Vue CLI

<https://cli.vuejs.org/>



- Inicialmente instale o Node JS.

```
PS C:\sites\vuejs-bluuweb> npm -v  
6.12.0
```

- Em seguida instale o Vue CLI com o seguinte comando:

```
npm install -g @vue/cli
```

```
npm install -g @vue/cli  
# OR  
yarn global add @vue/cli
```

```
Thank you for installing EJS: built with the Jake JavaScript build tool (https://jakejs.com/)  
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.11 (node_modules\@vue\cli\node_modules\fsevents):  
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.11: wanted {"os":"darwin","arch":"any"} (current: {"os":  
": "win32", "arch": "x64"})  
+ @vue/cli@4.1.1  
added 1082 packages from 623 contributors in 355.631s
```

```
vue --version
```

```
C:\sites\vuejs-sys4soft
λ vue --version
@vue/cli 4.1.1
```

- Instale o cmder.

Aula 22 - Criação de um projeto com Vue CLI

vue create vuecli001

```
Vue CLI v4.1.1
  Creating project in C:\sites\vuejs-sys4soft\vuecli001.
  Initializing git repository...
  Installing CLI plugins. This might take a while...

> yorkie@2.0.0 install C:\sites\vuejs-sys4soft\vuecli001\node_modules\yorkie
> node bin/install.js
setting up Git hooks
done

> core-js@3.6.0 postinstall C:\sites\vuejs-sys4soft\vuecli001\node_modules\core-js
> node -e "try{require('./postinstall')}catch(e){}"

> ejs@2.7.4 postinstall C:\sites\vuejs-sys4soft\vuecli001\node_modules\ejs
> node ./postinstall.js

added 1130 packages from 826 contributors and audited 24264 packages in 247.12s
found 0 vulnerabilities

  Invoking generators...
  Installing additional dependencies...

added 59 packages from 53 contributors and audited 24551 packages in 44.447s
found 0 vulnerabilities

  Running completion hooks...

  Generating README.md...

  Successfully created project vuecli001.
  Get started with the following commands:

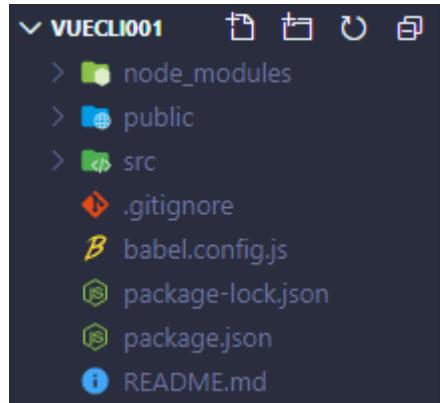
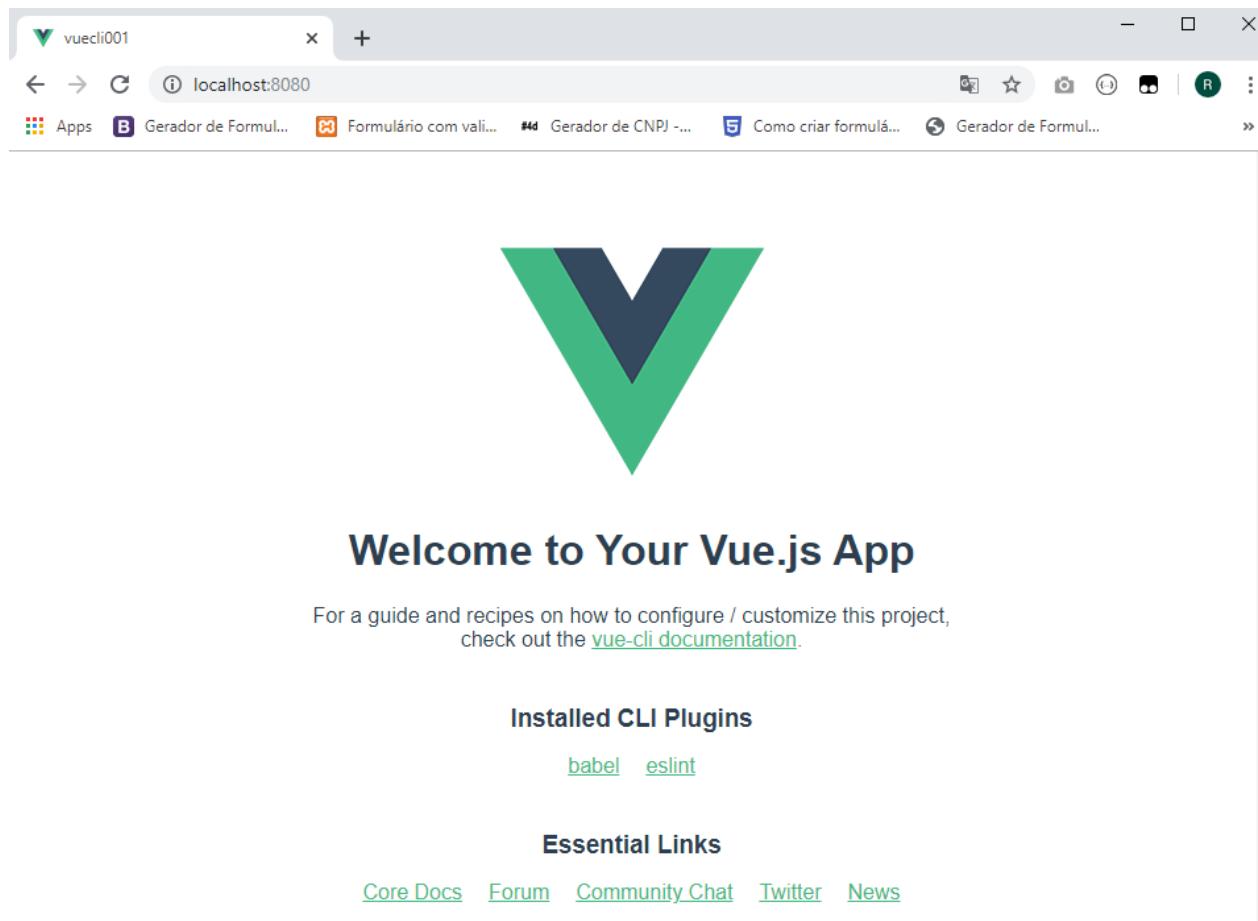
$ cd vuecli001
$ npm run serve
```

C:\sites\vuejs-sys4soft
λ

cd vuecli001
npm run serve

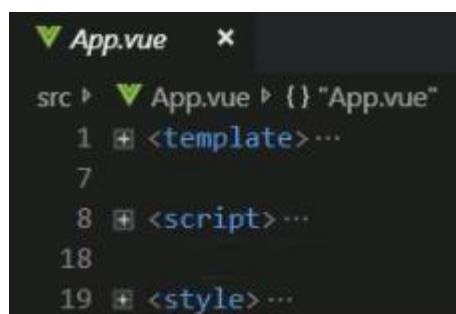
```
DONE Compiled successfully in 9540ms
App running at:
- Local:  http://localhost:8080/
- Network: http://192.168.0.10:8080/

Note that the development build is not optimized.
To create a production build, run npm run build.
```



- Instale o plugin Vetur.

Áreas de uma app vue:



Aula 23 - Análise e alteração de um projeto Vue CLI

src\components\Novo.vue

```
<template>
<div>
  <h1>Olá Vue!</h1>
</div>
</template>
```

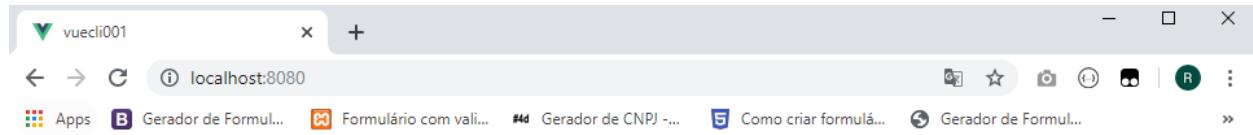
src\App.vue

```
<template>
<div id="app">
  <p>Este é o componente App</p>
  <novo />
</div>
</template>

<script>
import Novo from './components/Novo.vue'

export default {
  name: 'app',
  components: {
    Novo
  }
}
</script>

<style>
#app {
  font-family: 'Avenir', Helvetica, Arial, sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
  text-align: center;
  color: #2c3e50;
  margin-top: 60px;
}
</style>
```



Este é o componente App

Olá Vue!

Aula 24 - Mais sobre componentes Vue e NPM

src\components\Novo.vue

```
<template>
<div>
  <h1>Olá mundo! Meu nome é {{nome}} {{sobrenome}}</h1>
</div>
</template>

<script>
export default {
  methods: {
    click: function(){
      }
    },
  data: function(){
    return {
      nome: "João",
      sobrenome: "Ribeiro"
    }
  }
}
</script>
```

src\App.vue

```
<template>
<div id="app">
  <novo />
</div>
</template>

<script>
import Novo from './components/Novo.vue'

export default {
  name: 'app',
  components: {
    Novo
  }
}
</script>

<style>
#app {
  font-family: 'Avenir', Helvetica, Arial, sans-serif;
  -webkit-font-smoothing: antialiased;
```

```
-moz-osx-font-smoothing: grayscale;  
text-align: center;  
color: #2c3e50;  
margin-top: 60px;  
}  
</style>
```



Olá mundo! Meu nome é João Ribeiro

Aula 25 - Custom Components na Vue CLI

src\components\componente1.vue

```
<template>
  <div>
    <input type="text" v-model="nome">
    <p>O valor inserido é {{nome}}</p>
    <div>
      <button @click="limpar">Limpar</button>
    </div>
    <br />
  </div>
</template>

<script>
export default {
  data: function(){
    return {
      nome: ""
    }
  },
  methods: {
    limpar: function(){
      this.nome = "";
    }
  }
}
</script>

<style>
</style>
```

src\app.vue

```
<template>
  <div id="app">
    <componente1></componente1>
    <componente1 />
  </div>
</template>

<script>
// import Novo from './components/Novo.vue'
import componente1 from "./components/componente1.vue";

export default {
```

```

name: 'app',
components: {
  componente1
}
}
</script>

<style>
#app {
  font-family: 'Avenir', Helvetica, Arial, sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
  text-align: center;
  color: #2c3e50;
  margin-top: 60px;
}
</style>

```



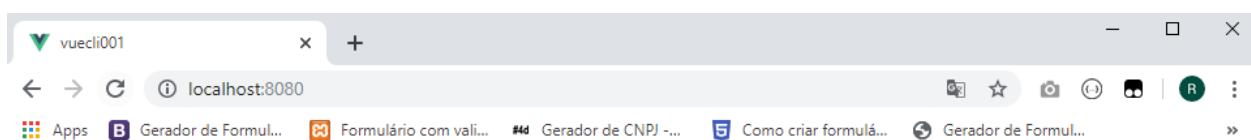
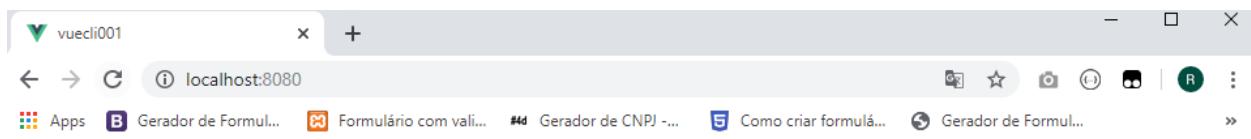
O valor inserido é

O valor inserido é



O valor inserido é João

O valor inserido é Roberto



O valor inserido é

O valor inserido é

Aula 26 - É possível ter duas instâncias Vue

src\main.js

```
import Vue from 'vue'  
import App from './App.vue'  
import App1 from './App1.vue'  
  
Vue.config.productionTip = false  
  
new Vue({  
  render: h => h(App),  
}).$mount('#app')  
  
new Vue({  
  render: h => h(App1),  
}).$mount('#app1')
```

public\index.html

```
<!DOCTYPE html>  
<html lang="en">  
  <head>  
    <meta charset="utf-8">  
    <meta http-equiv="X-UA-Compatible" content="IE=edge">  
    <meta name="viewport" content="width=device-width,initial-scale=1.0">  
    <link rel="icon" href="<%= BASE_URL %>favicon.ico">  
    <title>vuecli001</title>  
  </head>  
  <body>  
    <noscript>  
      <strong>We're sorry but vuecli001 doesn't work properly without JavaScript enabled.  
Please enable it to continue.</strong>  
    </noscript>  
    <div id="app"></div>  
    <div id="app1"></div>  
    <!-- built files will be auto injected -->  
  </body>  
</html>
```

src\App.vue

```
<template>
  <p>Componente APP</p>
</template>

<script>
</script>

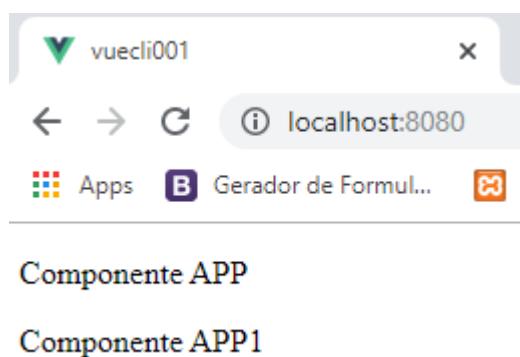
<style>
</style>
```

src\App1.vue

```
<template>
  <p>Componente APP1</p>
</template>

<script>
</script>

<style>
</style>
```



Aula 27 - Registrar Custom Components globalmente

Os componentes podem ser registrados localmente ou globalmente.

Registrando um componente globalmente

src\main.js

```
import Vue from 'vue'  
import App from './App.vue'  
import EstadoServidor from './components/EstadoServidor.vue'  
  
Vue.config.productionTip = false  
  
Vue.component('estado-servidor', EstadoServidor);  
  
new Vue({  
  render: h => h(App),  
}).$mount('#app')
```

src\components\EstadoServidor.vue

```
<template>  
  <div>  
    <p>Estado do servidor: {{estado}}</p>  
    <div>  
      <button @click="ligar">Ligar</button>  
      <button @click="desligar">Desligar</button>  
    </div>  
  </div>  
</template>  
  
<script>  
export default {  
  data: function(){  
    return {  
      estado: 'Ligado'  
    }  
  },  
  methods: {  
    ligar(){  
      this.estado = "Ligado"  
    },  
    desligar(){  
      this.estado = "Desligado"  
    }  
  }  
</script>
```

```
}
```

```
</script>
```

src\App.vue

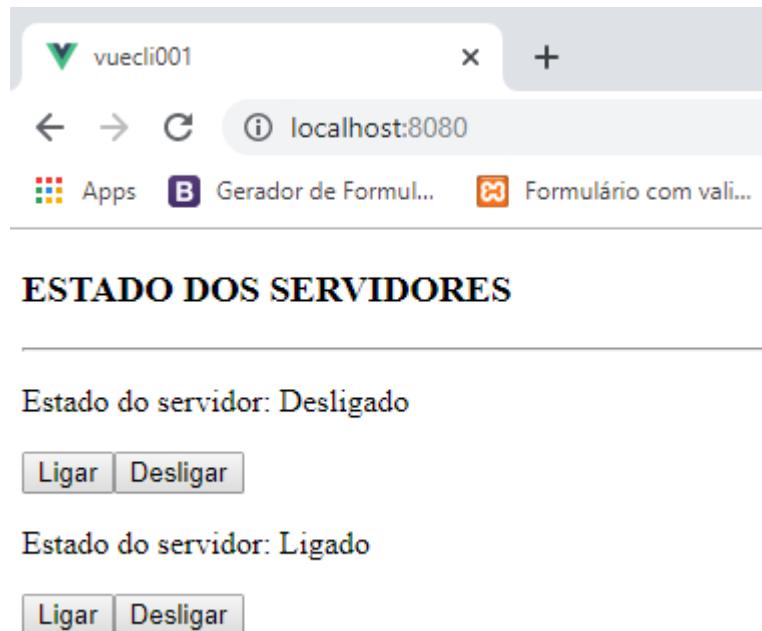
```
<template>
<div>
  <h3>ESTADO DOS SERVIDORES</h3>
  <hr>
  <estado-servidor></estado-servidor>
  <estado-servidor></estado-servidor>
</div>
</template>

<script>
```

```
</script>
```

```
<style>
```

```
</style>
```



Aula 28 - Bootstrap e acesso a base de dados - Esclarecimentos

Bootstrap CDN

public\index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,initial-scale=1.0">
    <link rel="icon" href="<%= BASE_URL %>/favicon.ico">
    <link rel="stylesheet"
      href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css" integrity="sha384-Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh" crossorigin="anonymous">
    <title>vuecli001</title>
  </head>
  <body>
    <noscript>
      <strong>We're sorry but vuecli001 doesn't work properly without JavaScript enabled. Please enable it to continue.</strong>
    </noscript>
    <div class="container">
      <div id="app"></div>
    </div>
    <!-- built files will be auto injected -->
  </body>
</html>
```

src\components\EstadoServidor.vue

```
<template>
  <div>
    <p>Estado do servidor: {{estado}}</p>
    <div>
      <button class="btn btn-success mr-2" @click="ligar">Ligar</button>
      <button class="btn btn-danger" @click="desligar">Desligar</button>
    </div>
  </div>
</template>

<script>
export default {
  data: function(){
    return {
      estado: 'Ligado'
    }
  }
}
```

```

},
methods: {
  ligar(){
    this.estado = "Ligado"
  },
  desligar(){
    this.estado = "Desligado"
  }
}
</script>

```

Esta é a App Vue

Dados do Servidor

Estado do servidor: Desligado

Ligar **Desligar**

Estado do servidor: Ligado

Ligar **Desligar**

Bootstrap para Vue

<https://bootstrap-vue.js.org/>

With **BootstrapVue** you can build responsive, mobile-first, and ARIA accessible projects on the web using Vue.js and the world's most popular front-end CSS library — Bootstrap v4.

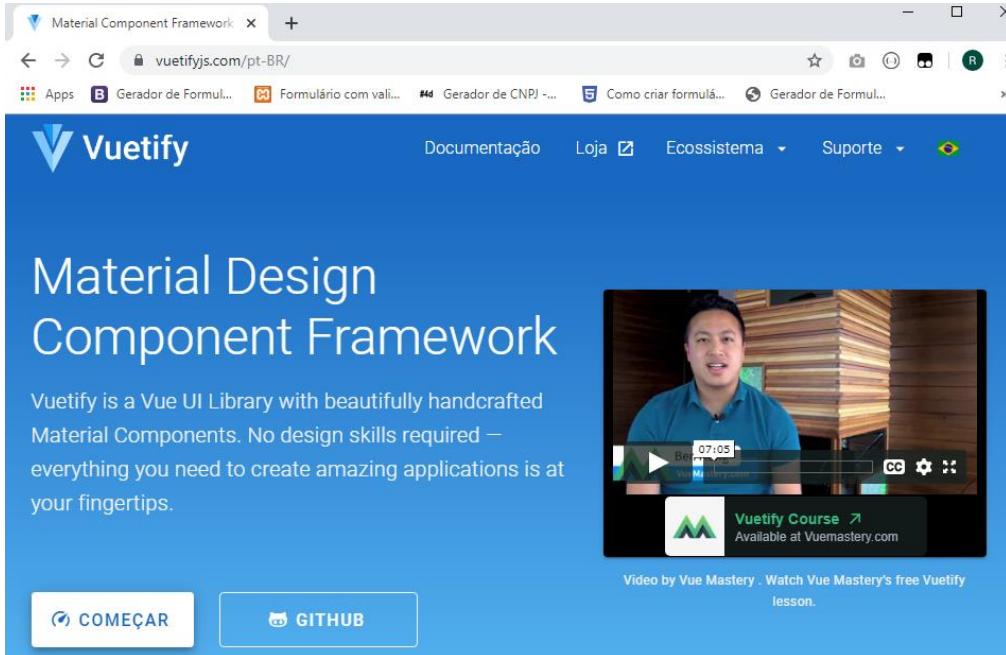
B **Bootstrap v4** is the world's most popular framework for building responsive, mobile-first sites.

V **Vue.js** (pronounced /ju:/, like view) is a progressive framework for building user interfaces.

Current Version
v2.1.0

Vuetify

<https://vuetifyjs.com/pt-BR/>



Aula 29 - Registrar componentes localmente

src\components\Servidor.vue

```
<template>
  <div>
    <p>Dados do Servidor</p>
  </div>
</template>
```

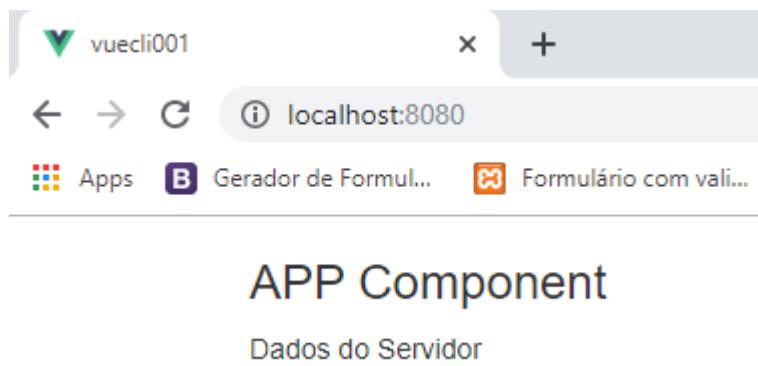
src\App.vue

```
<template>
  <div>
    <h3>APP Component</h3>
    <app-servidor></app-servidor>
  </div>
</template>

<script>
import Servidor from "./components/Servidor";

export default {
  components: {
    'app-servidor': Servidor
  }
}
</script>

<style>
</style>
```



Aula 30 - Exercício prático com componentes locais

src\App.vue

```
<template>
  <div>
    <p>Esta é a App Vue</p>
    <app-servidor></app-servidor>
  </div>
</template>

<script>
import Servidor from "./components/Servidor";

export default {
  components: {
    'app-servidor': Servidor
  }
}
</script>
```

src\components\EstadoServidor.vue

```
<template>
  <div>
    <p>Estado do servidor: {{estado}}</p>
    <div>
      <button class="btn btn-success" @click="ligar">Ligar</button>
      <button class="btn btn-danger" @click="desligar">Desligar</button>
    </div>
  </div>
</template>

<script>
export default {
  data: function(){
    return {
      estado: 'Ligado'
    }
  },
  methods: {
    ligar(){
      this.estado = "Ligado"
    },
    desligar(){
      this.estado = "Desligado"
    }
  }
}
</script>
```

src\components\TemperaturaServidor.vue

```
<template>
  <div>
    <p>Temperatura do servidor: 55 graus</p>
  </div>
</template>
```

src\components\Servidor.vue

```
<template>
  <div>
    <h4>Dados do Servidor</h4>
    <hr>
    <app-estado-servidor></app-estado-servidor>
    <br />
    <app-temperatura-servidor></app-temperatura-servidor>
  </div>
</template>

<script>

import EstadoServidor from "./EstadoServidor";
import TemperaturaServidor from "./TemperaturaServidor";

export default {
  components:{
    'app-estado-servidor': EstadoServidor,
    'app-temperatura-servidor': TemperaturaServidor
  }
}
</script>
```

Esta é a App Vue

Dados do Servidor

Estado do servidor: Ligado

Ligar Desligar

Temperatura do servidor: 55 graus

Aula 32 - Styles e scoped

src\components\comp1.vue

```
<template>
  <div>
    <p>Este é o texto do comp1.</p>
  </div>
</template>

<style scoped>
  p {
    color: red;
  }
</style>
```

src\components\comp2.vue

```
<template>
  <div>
    <p>Este é o texto do comp2.</p>
  </div>
</template>

<style scoped>
  p {
    background-color: yellow;
  }
</style>
```

src\App.vue

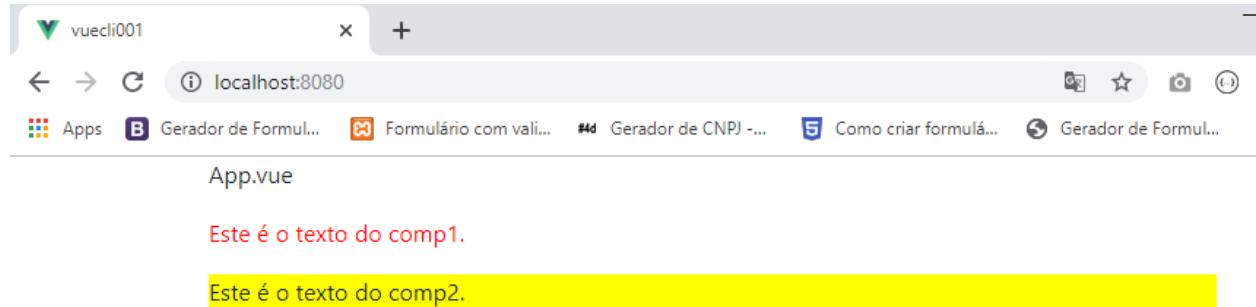
```
<template>
  <div>
    <p>App.vue</p>
    <comp1></comp1>
    <comp2></comp2>
  </div>
</template>

<script>
import comp1 from "./components/comp1";
import comp2 from "./components/comp2";

export default {
  components: {
    comp1, comp2
  }
}
</script>
```

```
<style>
```

```
</style>
```



Aula 33 - Comunicação de parent para child components

src\App.vue

```
<template>
<div class="container mt-3 box">
  <div class="row">
    <div class="col-12 p-4">
      <P>Mecanismo para alterar o valor do nome</P>
      <button @click="alterarNome">Alterar nome</button>
      <app-detalhe v-bind:meunome="nomeoriginal"></app-detalhe>
    </div>
  </div>
</div>
</template>

<script>
import Detalhe from './components/Detalhe';
export default {
  components: {
    'app-detalhe': Detalhe
  },
  data: function(){
    return {
      'nomeoriginal': 'João'
    }
  },
  methods:{
    alterarNome(){
      this.nomeoriginal = "Carlos";
    }
  }
}
</script>

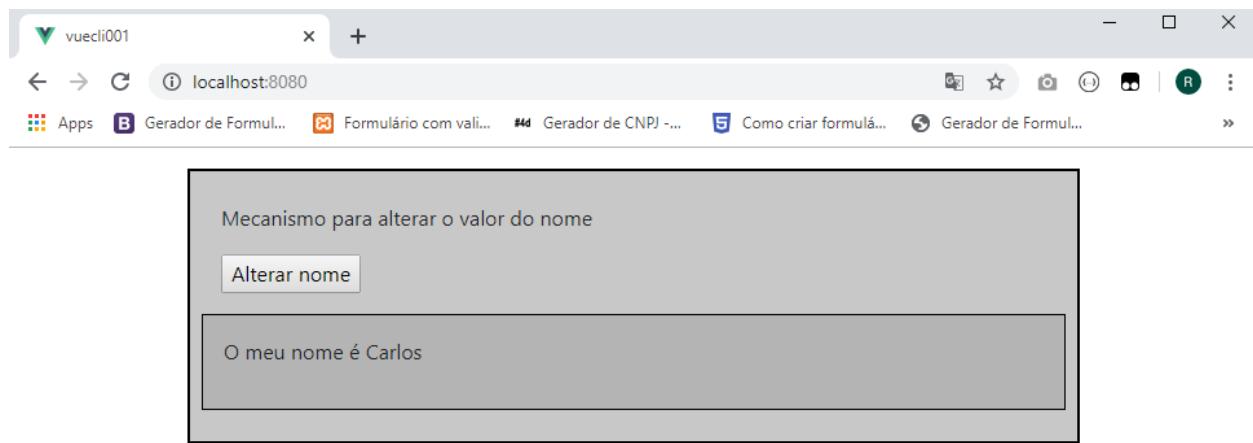
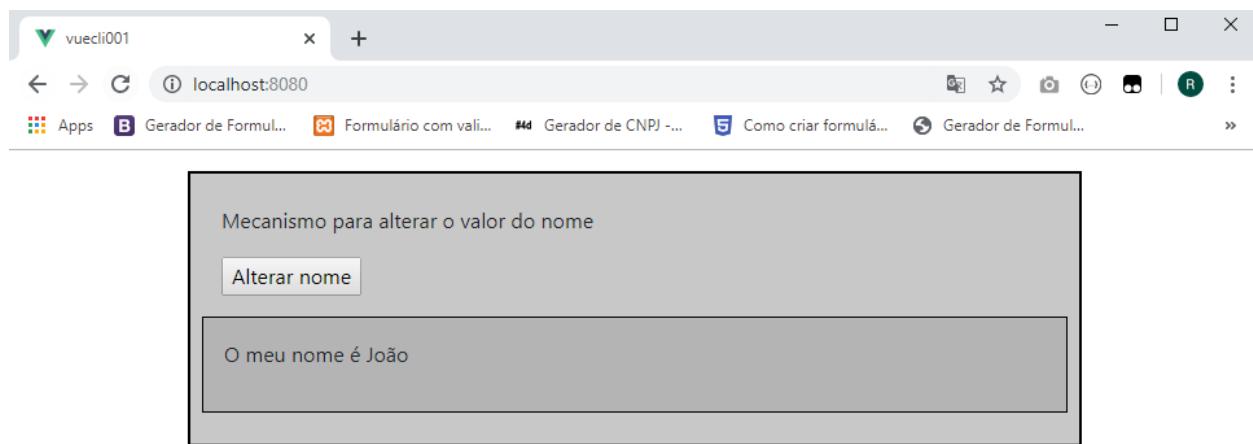
<style scoped>
.box{
  border: 2px solid black;
  background-color: rgb(200, 200, 200);
}
</style>
```

src\components\Detalhe.vue

```
<template>
  <div class="row box1 mt-3 p-3">
    <p>O meu nome é {{meunome}}</p>
  </div>
</template>

<script>
export default {
  props: ['meunome']
}
</script>

<style scoped>
  .box1{
    border: 1px solid black;
    background-color: rgb(180, 180, 180);
  }
</style>
```



Aula 34 - Comunicação de parent para child components - Exercício prático 1

src\components\infopessoa.vue

```
<template>
  <p>{{infonome}} {{infosobrenome}}</p>
</template>

<script>
export default {
  props: ['infonome', 'infosobrenome']
}
</script>
```

src\app.vue

```
<template>
<div class="container mt-2">
  <button class="btn btn-primary mr-2" @click="mostrarNome">Nome</button>
  <button class="btn btn-success" @click="mostrarSobrenome">Sobrenome</button>
  <hr>
  <app-infopessoa :infonome="nome" :infosobrenome="sobrenome"></app-infopessoa>
</div>
</template>

<script>
import infopessoa from './components/infopessoa';
export default {
  components: {
    'app-infopessoa': infopessoa
  },
  data: function(){
    return{
      nome: '',
      sobrenome: ''
    }
  },
  methods:{
    mostrarNome(){
      this.nome = "João";
    },
    mostrarSobrenome(){
      this.sobrenome = "Ribeiro"
    }
  }
}
</script>
```

vuecli001 x +

localhost:8080

Apps Gerador de Formul... Formulário com vali... Gerador de CNPJ -...

Nome Sobrenome

vuecli001 x +

localhost:8080

Apps Gerador de Formul... Formulário com vali...

Nome Sobrenome

João

vuecli001 x +

localhost:8080

Apps Gerador de Formul... Formulário com vali...

Nome Sobrenome

João Ribeiro

Aula 35 - Comunicação de parent para child components - Exercício prático 2

src\components\componente.vue

```
<template>
  <div class="alert alert-info">
    <p>{{analisar()}}</p>
  </div>
</template>

<script>
export default {
  props: ['meuvalor'],
  methods: {
    analisar(){
      if(this.meuvalor < 100){
        return 'Menor que 100';
      } else if (this.meuvalor == 100){
        return 'Igual a 100!';
      } else {
        return 'Maior que 100'
      }
    }
  }
}
</script>
```

src\App.vue

```
<template>
<div class="container mt-2">
  <button class="btn btn-primary m-1" @click="executar(0)">Click 0</button>
  <button class="btn btn-primary m-1" @click="executar(100)">Click 100</button>
  <button class="btn btn-primary m-1" @click="executar(200)">Click 200</button>
  <hr>
  <!-- componente child -->
  <app-componente :meuvalor="valor"></app-componente>
</div>
</template>

<script>
import componente from './components/componente';
export default {
  components: {
    'app-componente': componente
  },
}
```

```
data: function(){
  return{
    valor: 0,
  }
},
methods:{
  executar(value){
    this.valor = value;
  }
}
</script>
```

A screenshot of a web browser window titled "vuecli001". The address bar shows "localhost:8080". The page content includes three blue buttons labeled "Click 0", "Click 100", and "Click 200". Below these buttons is a light blue rectangular box containing the text "Menor que 100".

A screenshot of a web browser window titled "vuecli001". The address bar shows "localhost:8080". The page content includes three blue buttons labeled "Click 0", "Click 100", and "Click 200". Below these buttons is a light blue rectangular box containing the text "Igual a 100!".

A screenshot of a web browser window titled "vuecli001". The address bar shows "localhost:8080". The page content includes three blue buttons labeled "Click 0", "Click 100", and "Click 200". Below these buttons is a light blue rectangular box containing the text "Maior que 100".

Aula 36 - Props Validation

src\components\componente.vue

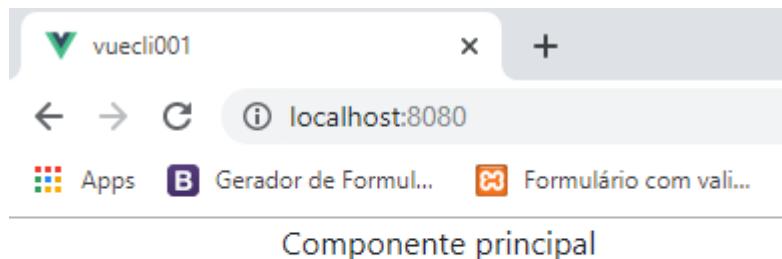
```
<template>
  <div>
    <p>Componente child</p>
    <p>{{valor1}}</p>
  </div>
</template>

<script>
export default {
  props: {
    valor1: {
      type: [String, Array],
      default: 'João Ribeiro'
    }
  }
}
</script>
```

src\App.vue

```
<template>
  <div>
    <p>Componente principal</p>
    <app-componente :valor1="nome"></app-componente>
  </div>
</template>

<script>
import componente from './components/componente';
export default {
  data: function(){
    return {
      nome: 'João'
    },
    components: {
      'app-componente': componente
    }
}
</script>
```



Componente principal

Componente child

João

Aula 37 - Comunicação de child component para parent component

src\App.vue

```
<template>
  <div>
    <p>(App.vue) Nome = {{nome}}</p>
    <button class="btn btn-primary" @click="alterar">Alterar</button>
    <hr>
    <app-componente :meunome = nome @nomeAlterado="nome = $event"></app-componente>
  </div>
</template>

<script>
import componente from './components/componente'
export default {
  components: {
    'app-componente': componente
  },
  data: function(){
    return {
      nome: 'João'
    }
  },
  methods: {
    alterar(){
      this.nome='Carlos';
    }
  }
}
</script>
```

src\components\componente.vue

```
<template>
  <div>
    <p>(Componente.vue) Nome = {{meunome}} </p>
    <button class="btn btn-success" @click="alterar">Alterar</button>
  </div>
</template>

<script>
export default {
  props: [
    'meunome'
  ],
  data: function(){
    return {
      nome: 'João'
    }
  },
  methods: {
    alterar(){
      this.nome='Carlos';
    }
  }
}
</script>
```

```
methods:{  
    alterar(){  
        this.meunome = 'Joaquim';  
        this.$emit('nomeAlterado', this.meunome);  
    }  
}  
}</script>
```

vuecli001

localhost:8080

Apps Gerador de Formul... Formulário com vali...

(App.vue) Nome = João

Alterar

(Componente.vue) Nome = João

Alterar

vuecli001

localhost:8080

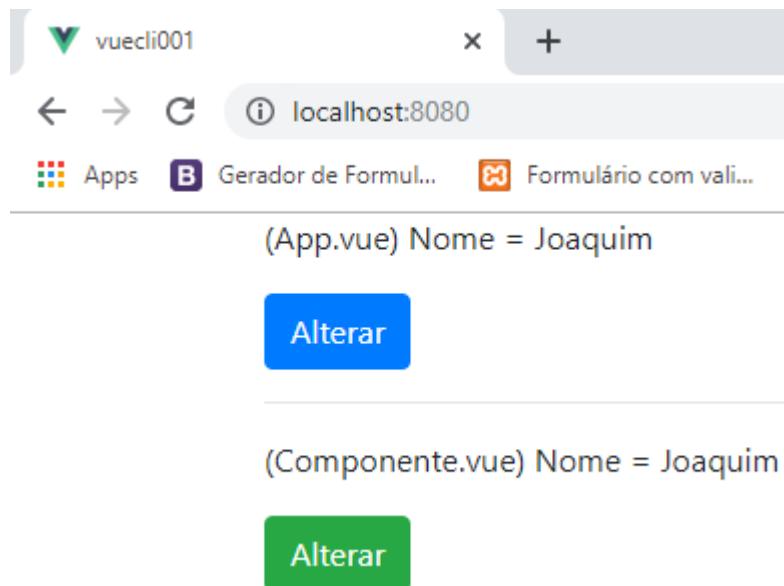
Apps Gerador de Formul... Formulário com vali...

(App.vue) Nome = Carlos

Alterar

(Componente.vue) Nome = João

Alterar



Aula 39 - Múltiplos componentes dentro do index

aula-39/index.html

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <link rel="stylesheet" href="../css/bootstrap.min.css">
  <title>Múltiplos componentes dentro do index</title>
</head>
<body>

  <div id="app">
    <div class="container m-3">
      <p>{{nome}} {{sobrenome}}</p>
      <comp1></comp1>
      <comp2></comp2>

    </div>
  </div>

  <script src="../js/vue.js"></script>
  <script src="js/script.js"></script>
</body>
</html>
```

aula-39/js/script.js

```
// ----

Vue.component('comp1', {
  template:
`

<p>Componente 1 - {{valor1}}</p>
  <p>Componente 1 - {{valor2}}</p>
  <button class="btn btn-primary m-3" @click="executar">Clicar</button>
</div>

`,
  data: function(){
    return {
      valor1: 100,
      valor2: 200
    }
  }
})


```

```

},
methods:{
  executar: function(){
    this.valor1 = 2000,
    this.valor2 = 500
  }
})
// ----

Vue.component('comp2', {
  template:
`<p>Componente 2</p>
`)

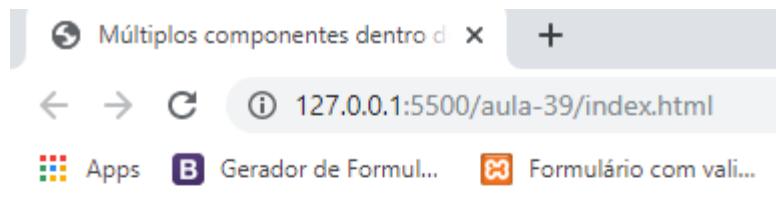
})
// ----

let app = new Vue({
  el: "#app",

  data: {
    nome: 'João',
    sobrenome: 'Ribeiro'
  }

});

```



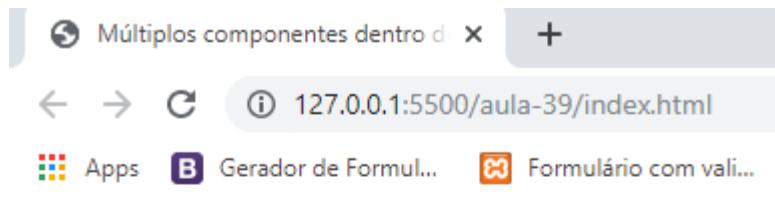
João Ribeiro

Componente 1 - 100

Componente 1 - 200

Clicar

Componente 2



João Ribeiro

Componente 1 - 2000

Componente 1 - 500

Clicar

Componente 2

Aula 40 - Comunicação entre componentes dentro do index

aula-40/index.html

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <link rel="stylesheet" href="../css/bootstrap.min.css">
  <title>Comunicação entre componentes dentro do index</title>
</head>
<body>

  <div id="app">
    <div class="container m-3">
      <p>Esse texto é da instância Vue</p>
      <button class="btn btn-primary m-2" @click="alterar">Alterar</button>
      <hr>
      <comp1 :valor = "valor"></comp1>
    </div>
  </div>

  <script src="../js/vue.js"></script>
  <script src="js/script.js"></script>
</body>
</html>
```

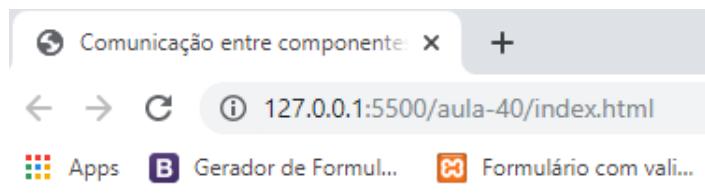
aula-40/js/script.js

```
// -----
Vue.component('comp1', {
  template:
  `

    <div>
      <p>Valor dentro do componente 1: {{valor}}</p>
    </div>

  `,
  data: function(){
    return {
      valor: 1000
    }
  },
});
```

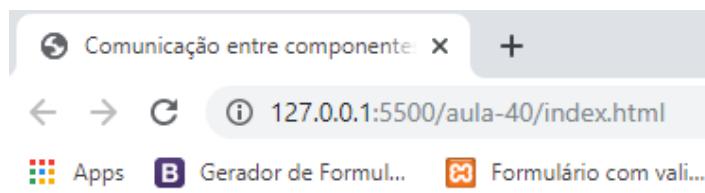
```
props: [
  'valor'
]
})
// -----
new Vue({
  el: "#app",
  data: {
    valor: 20000
  },
  methods: {
    alterar: function(){
      this.valor += 500;
      console.log(this.valor);
    }
  }
});
```



Esse texto é da instância Vue

Alterar

Valor dentro do componente 1: 20000



Esse texto é da instância Vue

Alterar

Valor dentro do componente 1: 20500

Comunicação entre componentes x +

← → C 127.0.0.1:5500/aula-40/index.html

grid Apps document Gerador de Formul... form Formulário com vali...

Esse texto é da instância Vue

outline Alterar

Valor dentro do componente 1: 21000

Aula 41 - Introdução ao VUEX

<https://vuex.vuejs.org/>

The screenshot shows a browser window displaying the official Vuex documentation at <https://vuex.vuejs.org/ptbr/>. The title bar says "O que é Vuex? | Vuex". The main content area features a large image titled "Vuex Explained Visually" which includes a Mailchimp logo and a brain icon, suggesting Vuex is a state management library for Vue.js. Below the image, a text block explains what Vuex is: "O Vuex é um padrão de gerenciamento de estado + biblioteca para aplicativos Vue.js. Ele serve como um store centralizado para todos os componentes em uma aplicação, com regras garantindo que o estado só possa ser mutado de forma previsível. Ele também se integra com a extensão oficial [Vue devtools](#) para fornecer recursos avançados sem". On the left sidebar, there's a "Marketing" section with a Mailchimp logo and text about tracking campaigns via Carbon ads, followed by navigation links like "Instalação", "O que é Vuex?", and "Começando". Another sidebar section titled "Conceitos Básicos" lists concepts such as "Estado", "Getters", "Mutações", "Ações", "Módulos", and "Estrutura da Aplicação".

aula-41/index.html

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <link rel="stylesheet" href="../css/bootstrap.min.css">
  <title>Introdução ao VUEX</title>
</head>
<body>

  <div id="app">
    <div class="container m-3">
      <p>O meu nome é {{meunome}} {{meusobrenome}}</p>
    </div>
  </div>

  <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/vuex/3.1.2/vuex.min.js"></script>
  <script src="js/script.js"></script>
</body>
</html>
```

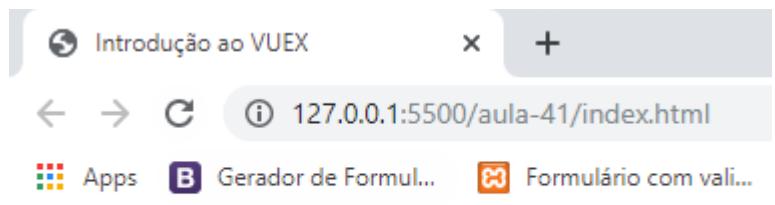
aula-41/js/script.js

```
// -----
// VUEX

const store = new Vuex.Store({
  state: {
    nome: 'João',
    sobrenome: 'Ribeiro'
  }
});

// -----


new Vue({
  el: "#app",
  computed: {
    meunome: function(){
      return store.state.nome;
    },
    meusobrenome: function(){
      return store.state.sobrenome;
    }
});
});
```



O meu nome é João Ribeiro

Aula 42 - Mutations em VUEX

aula-42/index.html

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <link rel="stylesheet" href="../css/bootstrap.min.css">
  <title>Mutations em VUEX</title>
</head>
<body>

  <div id="app">
    <div class="container m-3">
      <button class="btn btn-success mr-2 mb-2" @click="aumentar()">Aumentar</button>
      <button class="btn btn-danger mb-2" @click="diminuir()">Diminuir</button>
      <p>Valor: {{valor}}</p>
    </div>
  </div>

  <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
  <script src="https://cdn.jsdelivr.net/npm/vuex@3.1.2/dist/vuex.min.js"></script>
  <script src="js/script.js"></script>
</body>
</html>
```

aula-42/js/script.js

```
// -----
// VUEX

const store = new Vuex.Store({
  state: {
    valor: 0
  },
  mutations: {
    incrementar(state){
      state.valor++;
    },
    decrementar(state){
      state.valor--;
    }
  }
});

// -----


new Vue({
  el: "#app",
```

```
computed:{  
    valor: function(){  
        return store.state.valor;  
    }  
},  
methods: {  
    aumentar: function(){  
        store.commit('incrementar');  
    },  
    diminuir: function(){  
        store.commit('decrementar');  
    }  
}  
});
```

Mutations em VUEX

127.0.0.1:5500/aula-42/index.html

Apps Gerador de Formul... Formulário com vali...

Aumentar Diminuir

Valor: 0

Mutations em VUEX

127.0.0.1:5500/aula-42/index.html

Apps Gerador de Formul... Formulário com vali...

Aumentar Diminuir

Valor: 5

Mutations em VUEX

127.0.0.1:5500/aula-42/index.html

Apps Gerador de Formul... Formulário com vali...

Aumentar Diminuir

Valor: 4

Aula 43 - Mutations de VUEX em múltiplos components

aula-43/index.html

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <link rel="stylesheet" href="../css/bootstrap.min.css">
  <title>Mutations de VUEX em múltiplos components</title>
</head>
<body>

  <div id="app">
    <div class="container m-3">
      <p>Valor na instância: {{valor}}</p>
      <button class="btn btn-success mr-2 mb-2" @click="aumentar()">Aumentar</button>
      <button class="btn btn-danger mb-2" @click="diminuir()">Diminuir</button>
      <hr>
      <componente1></componente1>
    </div>
  </div>

  <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/vuex/3.1.2/vuex.min.js"></script>
  <script src="js/script.js"></script>
</body>
</html>
```

aula-43/js/script.js

```
// -----
// VUEX

const store = new Vuex.Store({
  state: {
    valor: 0
  },
  mutations: {
    incrementar(state){
      state.valor++;
    },
    decrementar(state){
      state.valor--;
    }
  }
});

// -----
```

```

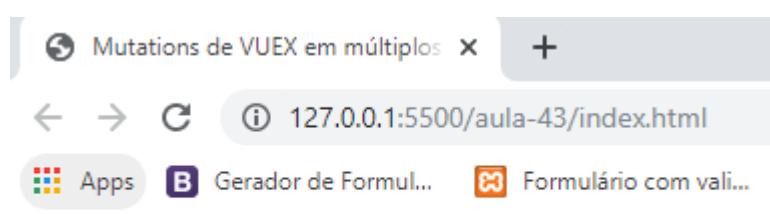
Vue.component('componente1', {
  template:
`<div>
  <button @click="aumentar()" class="btn btn-success mr-2">Aumentar</button>
  <button @click="diminuir()" class="btn btn-danger">Diminuir</button>
</div>
`,

  methods: {
    aumentar: function(){
      store.commit('incrementar');
    },
    diminuir: function(){
      store.commit('decrementar');
    }
  }
})

// ----

new Vue({
  el: "#app",
  computed:{
    valor: function(){
      return store.state.valor;
    }
  },
  methods: {
    aumentar: function(){
      store.commit('incrementar');
    },
    diminuir: function(){
      store.commit('decrementar');
    }
  }
});

```



Valor na instância: 0

Aumentar **Diminuir**

Aumentar **Diminuir**

S Mutations de VUEX em múltiplos X +

← → C ⓘ 127.0.0.1:5500/aula-43/index.html

_apps Gerador de Formul... Formulário com vali...

Valor na instância: 5

Aumentar Diminuir

Aumentar Diminuir

S Mutations de VUEX em múltiplos X +

← → C ⓘ 127.0.0.1:5500/aula-43/index.html

_apps Gerador de Formul... Formulário com vali...

Valor na instância: 3

Aumentar Diminuir

Aumentar Diminuir

Aula 44 - VUEX getters

aula-44/index.html

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <link rel="stylesheet" href="../css/bootstrap.min.css">
  <title>VUEX getters</title>
</head>
<body>

  <div id="app">
    <div class="container m-3">
      <p>{{nomeCompleto}}</p>
      <componente1></componente1>
    </div>
  </div>

  <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/vuex/3.1.2/vuex.min.js"></script>
  <script src="js/script.js"></script>
</body>
</html>
```

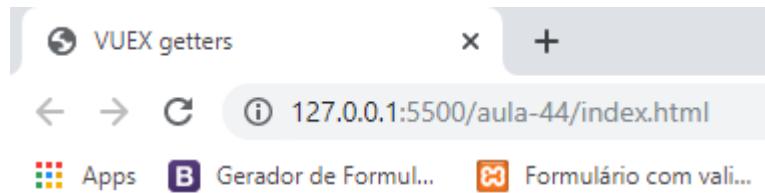
aula-44/js/script.js

```
// -----
// VUEX

const store = new Vuex.Store({
  state: {
    nome: 'João',
    sobrenome: 'Ribeiro'
  },
  getters: {
    nomeCompleto: function(state){
      return 'O meu nome é ' + state.nome + ' ' + state.sobrenome;
    }
  }
});
```

// -----

```
new Vue({  
  el: "#app",  
  computed:{  
    nomeCompleto: function(){  
      return store.getters.nomeCompleto;  
    }  
  }  
});
```



O meu nome é João Ribeiro

Aula 45 - VUEX mutations com parâmetros

aula-45/index.html

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <link rel="stylesheet" href="../css/bootstrap.min.css">
  <title>VUEX mutations com parâmetros</title>
</head>
<body>

  <div id="app">
    <div class="container m-3">
      <p>Valor: {{valor}}</p>
      <button class="btn btn-success mr-2 mb-2" @click="aumentar()">Aumentar</button>
      <button class="btn btn-danger mb-2" @click="diminuir()">Diminuir</button>
    </div>
  </div>

  <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
  <script src="https://cdn.jsdelivr.net/npm/vuex@3.1.2/dist/vuex.min.js"></script>
  <script src="js/script.js"></script>
</body>
</html>
```

aula-45/js/script.js

```
// -----
// VUEX

const store = new Vuex.Store({
  state: {
    valor: 0
  },
  mutations:{
    aumentar: function(state, value){
      state.valor+=value
    },
    diminuir: function(state, value){
      state.valor-=value
    }
  }
});

// -----


new Vue({
  el: "#app",
```

```
computed:{  
    valor: function(){  
        return store.state.valor;  
    }  
},  
methods: {  
    aumentar: function(){  
        store.commit('aumentar', 10);  
    },  
    diminuir: function(){  
        store.commit('diminuir', 5);  
    },  
}  
});
```

VUEX mutations com parâmetros × +

← → C ⓘ 127.0.0.1:5500/aula-45/index.html

_apps Gerador de Formul... Formulário com vali...

Valor: 0

Aumentar Diminuir

VUEX mutations com parâmetros × +

← → C ⓘ 127.0.0.1:5500/aula-45/index.html

_apps Gerador de Formul... Formulário com vali...

Valor: 10

Aumentar Diminuir

VUEX mutations com parâmetros × +

← → C ⓘ 127.0.0.1:5500/aula-45/index.html

_apps Gerador de Formul... Formulário com vali...

Valor: 5

Aumentar Diminuir

Aula 46 - VUEX actions

aula-46/index.html

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <link rel="stylesheet" href="../css/bootstrap.min.css">
  <title>VUEX actions</title>
</head>
<body>

  <div id="app">
    <div class="container m-3">
      <p>Valor: {{valor}}</p>
      <button class="btn btn-success mr-2 mb-2" @click="aumentar()">Aumentar</button>
      <button class="btn btn-danger mb-2" @click="diminuir()">Diminuir</button>
    </div>
  </div>

  <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
  <script src="https://cdn.jsdelivr.net/npm/vuex@3.1.2/dist/vuex.min.js"></script>
  <script src="js/script.js"></script>
</body>
</html>
```

aula-46/js/script.js

```
// -----
// VUEX

const store = new Vuex.Store({
  state: {
    valor: 0
  },
  mutations: {
    aumentar: function(state){
      state.valor++
    },
    diminuir: function(state){
      state.valor--
    }
  },
  actions: {
    aumentar: function(context){
      setTimeout(function(){
        context.commit('aumentar');
      },2000);
    }
  }
})
```

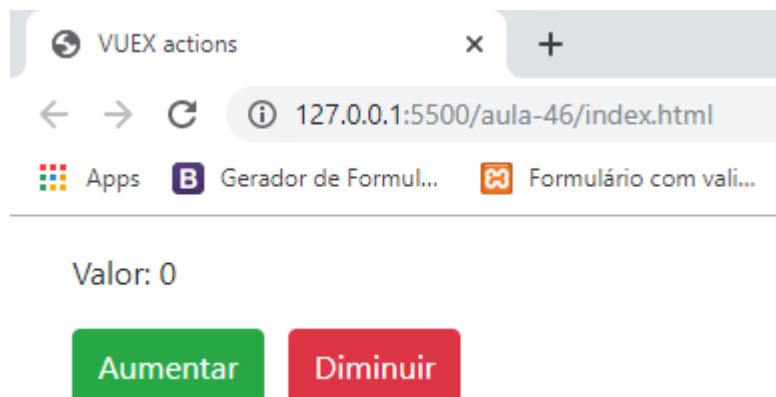
```

        }
    });

// ----

new Vue({
  el: "#app",
  computed: {
    valor: function(){
      return store.state.valor;
    }
  },
  methods: {
    aumentar: function(){
      store.dispatch('aumentar');
    },
    diminuir: function(){
      store.commit('diminuir');
    },
  }
});

```



- Clicando no botão "Aumentar", após 2 segundos:

