

Vue.JS desde cero - Aulas 1 a 14

Bluuweb - Ignacio Gutierrez

<https://www.youtube.com/watch?v=GAQB7Y4X5fM&list=PLPI81lqbj-4J-gfAERGDCdOQtVgRhSvIT>

Resumo do curso feito por Roberto Pinheiro

Aula 1 - Introdução

aula-1/index.html

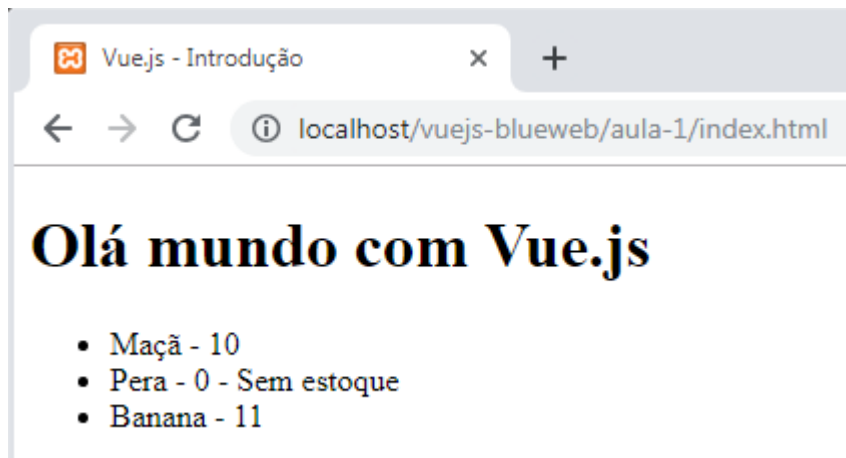
```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Vue.js - Introdução</title>
</head>
<body>
  <div id="app">
    <h1>{{titulo}}</h1>
    <ul>
      <!-- <li v-for="fruta of frutas">{{fruta}}</li> -->
      <li v-for="fruta of frutas">{{fruta.nome}} - {{fruta.quantidade}}
        <span v-if="fruta.quantidade === 0"> - Sem estoque</span>
      </li>
    </ul>
  </div>

  <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
  <script src="aula-1.js"></script>

</body>
</html>
```

aula-1/aula-1.js

```
const app = new Vue({
  el: '#app',
  data: {
    titulo: 'Olá mundo com Vue.js',
    // frutas: ['Maçã', 'Pera', 'Banana']
    frutas: [
      {
        nome: "Maçã",
        quantidade: 10
      },
      {
        nome: "Pera",
        quantidade: 0
      },
      {
        nome: "Banana",
        quantidade: 11
      }
    ]
  }
});
```



Aula 2 - Methods, v-model e Evento Click

localhost/vuejs-blueweb/aula-2/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>Vue.js - Methods v-model e evento click</title>
</head>
<body>
  <div id="app">
    <h1>{{titulo}}</h1>

    <input type="text" v-model="novaFruta">
    <button @click="addFruta">Adicionar</button>

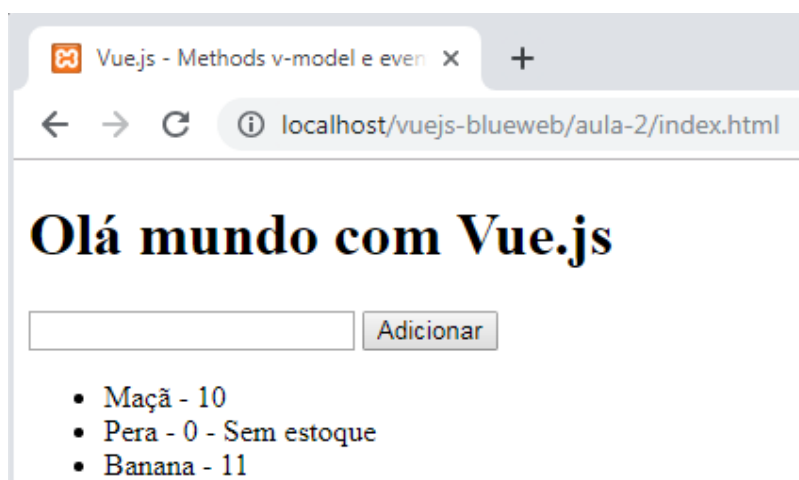
    <ul>
      <!-- <li v-for="fruta of frutas">{{fruta}}</li> -->
      <li v-for="fruta of frutas">{{fruta.nome}} - {{fruta.quantidade}}
        <span v-if="fruta.quantidade === 0"> - Sem estoque</span>
      </li>
    </ul>
  </div>

  <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
  <script src="aula-2.js"></script>

</body>
</html>
```

aula-2/aula-2.js

```
const app = new Vue({
  el: '#app',
  data: {
    titulo: 'Olá mundo com Vue.js',
    novaFruta: '',
    // frutas: ['Maçã', 'Pera', 'Banana']
    frutas: [
      {
        nome: "Maçã",
        quantidade: 10
      },
      {
        nome: "Pera",
        quantidade: 0
      },
      {
        nome: "Banana",
        quantidade: 11
      }
    ]
  },
  methods: {
    addFruta(){
      // console.log('O botão foi clicado!');
      this.frutas.push(
        {nome: this.novaFruta, quantidade: 0}
      );
    }
  }
});
```



Vue.js - Methods v-model e even x +

← → ↻ ⓘ

localhost/vuejs-blueweb/aula-2/index.html

Olá mundo com Vue.js

- Maçã - 10
- Pera - 0 - Sem estoque
- Banana - 11

Vue.js - Methods v-model e even x +

← → ↻ ⓘ

localhost/vuejs-blueweb/aula-2/index.html

Olá mundo com Vue.js

- Maçã - 10
- Pera - 0 - Sem estoque
- Banana - 11
- Laranja - 0 - Sem estoque

Aula 3 - keyup, v-model e Computed

localhost/vuejs-blueweb/aula-3/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>Vue.js - keyup, v-model e computed</title>
</head>
<body>
  <div id="app">
    <h1>{{titulo}}</h1>

    <input type="text" v-model="novaFruta" @keyup.enter="addFruta">
    <button @click="addFruta">Adicionar</button>

    <ul>
      <!-- <li v-for="fruta of frutas">{{fruta}}</li> -->
      <li v-for="fruta of frutas">
        <input type="text" v-model.number="fruta.quantidade">
        {{fruta.nome}}
        <button @click="fruta.quantidade++">+</button>
        <span v-if="fruta.quantidade === 0"> - Sem estoque</span>
      </li>
    </ul>
    <h4>Total: {{somarFruta}}</h4>
  </div>

  <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
  <script src="aula-3.js"></script>

</body>
</html>
```

aula-3/aula-3.js

```
const app = new Vue({
  el: '#app',
  data: {
    titulo: 'Olá mundo com Vue.js',
    novaFruta: "",
    total: 0,
    // frutas: ['Maçã', 'Pera', 'Banana']
    frutas: [
      {
        nome: "Maçã",
        quantidade: 10
      },
      {
        nome: "Pera",
        quantidade: 0
      },
      {
        nome: "Banana",
        quantidade: 11
      }
    ]
  },
  methods: {
    addFruta(){
      // console.log('O botão foi clicado!');
      this.frutas.push(
        {nome: this.novaFruta, quantidade: 0}
      );
      this.novaFruta = "";
    }
  },
  computed: {
    somarFruta(){
      this.total = 0;
      for(fruta of this.frutas){
        this.total += fruta.quantidade;
      }
      return this.total;
    }
  }
});
```

Vue.js - keyup, v-model e compu

+

← → ↻ ⓘ

localhost/vuejs-blueweb/aula-3/index.html

Olá mundo com Vue.js

Adicionar

• 10

Maçã

+

• 0

Pera

+

- Sem estoque

• 11

Banana

+

Total: 21

Vue.js - keyup, v-model e compu

+

← → ↻ ⓘ

localhost/vuejs-blueweb/aula-3/index.html

Olá mundo com Vue.js

Adicionar

• 12

Maçã

+

• 3

Pera

+

• 15

Banana

+

Total: 30

Aula 4 - Classes (v-bind: class)

localhost/vuejs-blueweb/aula-4/index.html

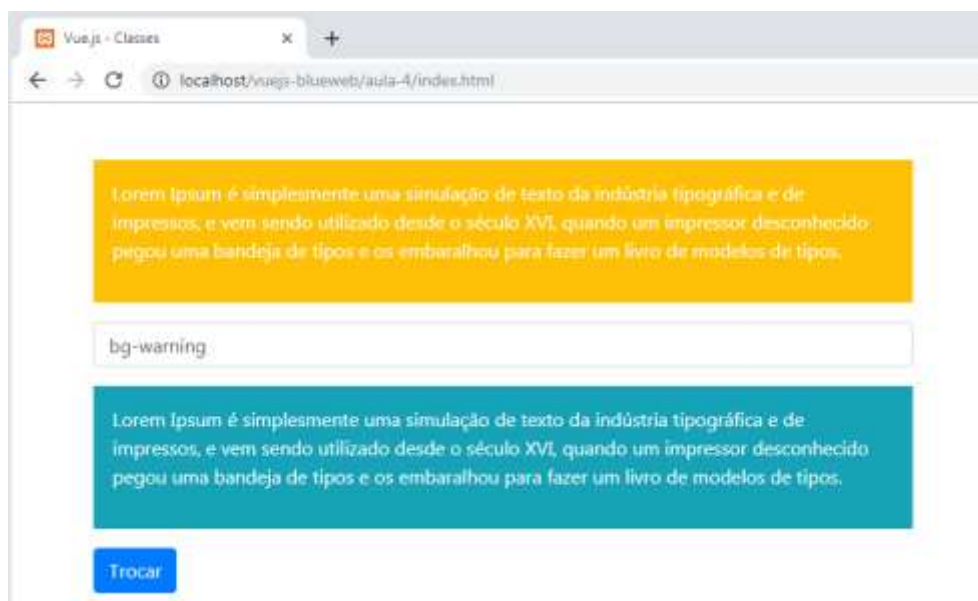
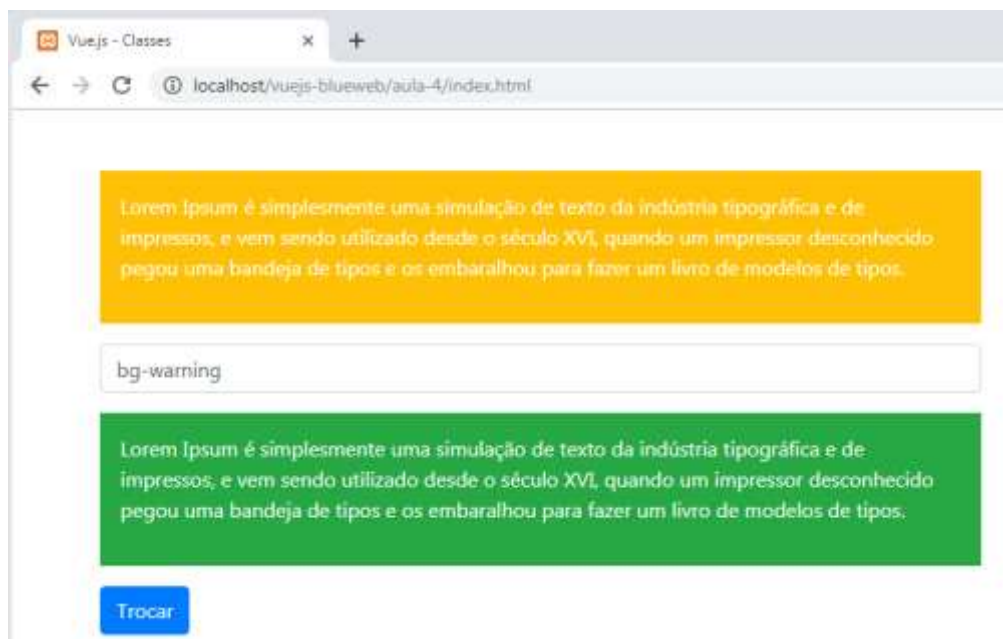
```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Vue.js - Classes</title>
  <link
                                                                    rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">
</head>
<body>
  <div id="app" class="container mt-5">
    <div :class="['p-3', 'text-white', fundo]">
      <p>Lorem Ipsum é simplesmente uma simulação de texto da indústria tipográfica e de
impressos, e vem sendo utilizado desde o século XVI, quando um impressor desconhecido
pegou uma bandeja de tipos e os embaralhou para fazer um livro de modelos de tipos.</p>
    </div>
    <input type="text" class="form-control my-3" v-model="fundo">
    <div class="p-3 text-white" :class="{ 'bg-info' : cor, 'bg-success' : !cor}">
      <p>Lorem Ipsum é simplesmente uma simulação de texto da indústria tipográfica e de
impressos, e vem sendo utilizado desde o século XVI, quando um impressor desconhecido
pegou uma bandeja de tipos e os embaralhou para fazer um livro de modelos de tipos.</p>
    </div>
    <button class="btn btn-primary mt-3" @click="cor = !cor">Trocar</button>
  </div>

  <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
  <script src="aula-4.js"></script>

</body>
</html>
```

aula-4/aula-4.js

```
const app = new Vue({  
  el: '#app',  
  data: {  
    fundo: 'bg-warning',  
    cor: false  
  },  
  methods: {  
  }  
});
```



Aula 5 - LocalStorage com Vue.js

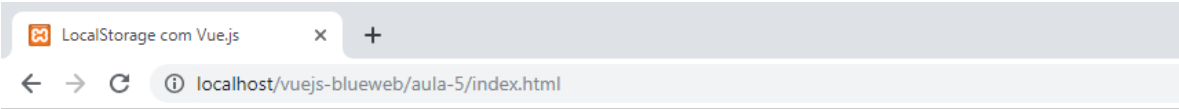
localhost/vuejs-blueweb/aula-5/index.html

```
<!doctype html>
<html lang="pt-br">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <link
    href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
    rel="stylesheet"
  >
  <title>LocalStorage com Vue.js</title>
</head>
<body>
  <div id="app" class="container mt-5">
    <h3>{{titulo}}</h3>
    <input type="text" id="atividade" class="form-control my-3" v-model="novaTarefa" v-
on:keyup.enter="agregarTarefa">
    <button
      type="submit"
      class="btn
      btn-primary"
      @click="agregarTarefa">Guardar</button>
    <div class="mt-3" v-for="(item, index) of tarefas">
      <div :class="['alert', item.status ? 'alert-success' : 'alert-danger']" role="alert">
        <div class="d-flex justify-content-between align-items-center">
          <div>
            {{index}} - {{item.nome}} - {{item.status}}
          </div>
          <div>
            <button class="btn btn-success" @click="editarTarefa(index)">Ok</button>
            <button class="btn btn-danger" @click="eliminarTarefa(index)">X</button>
          </div>
        </div>
      </div>
    </div>
  </div>
  <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
  <script src="app.js"></script>
</body>
</html>
```

aula-5/app.js

```
const app = new Vue({
  el: '#app',
  data: {
    titulo: 'GYM com Vue',
    tarefas: [],
    novaTarefa: ''
  },
  methods: {
    agregarTarefa: function(){
      // console.log('O botão foi clicado! ', this.novaTarefa);
      this.tarefas.push({
        nome: this.novaTarefa,
        status: false
      });
      console.log(this.tarefas);
      this.novaTarefa = '';
      localStorage.setItem('gym-vue', JSON.stringify(this.tarefas));
    },
    editarTarefa: function(index){
      this.tarefas[index].status = true;
      localStorage.setItem('gym-vue', JSON.stringify(this.tarefas));
    },
    eliminarTarefa: function(index){
      this.tarefas.splice(index, 1);
      localStorage.setItem('gym-vue', JSON.stringify(this.tarefas));
    }
  },
  created: function(){
    let dadosDB = JSON.parse(localStorage.getItem('gym-vue'));
    if(dadosDB === null){
      this.tarefas = [];
    } else {
      this.tarefas = dadosDB;
    }
  }
});
```

Estado inicial



GYM com Vue

Guardar

Application

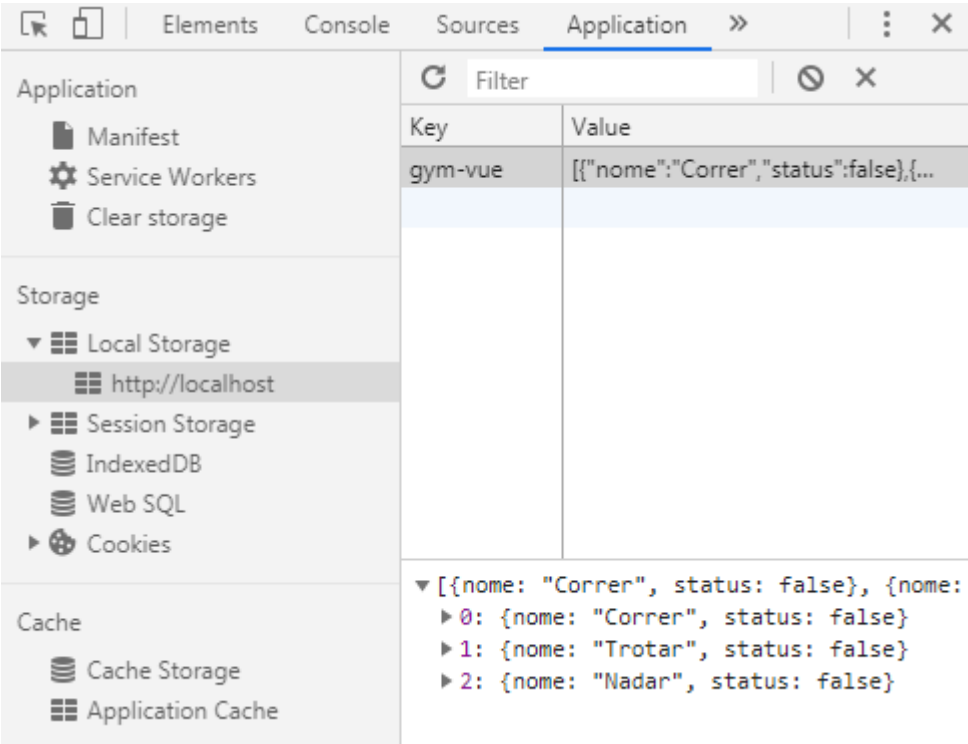
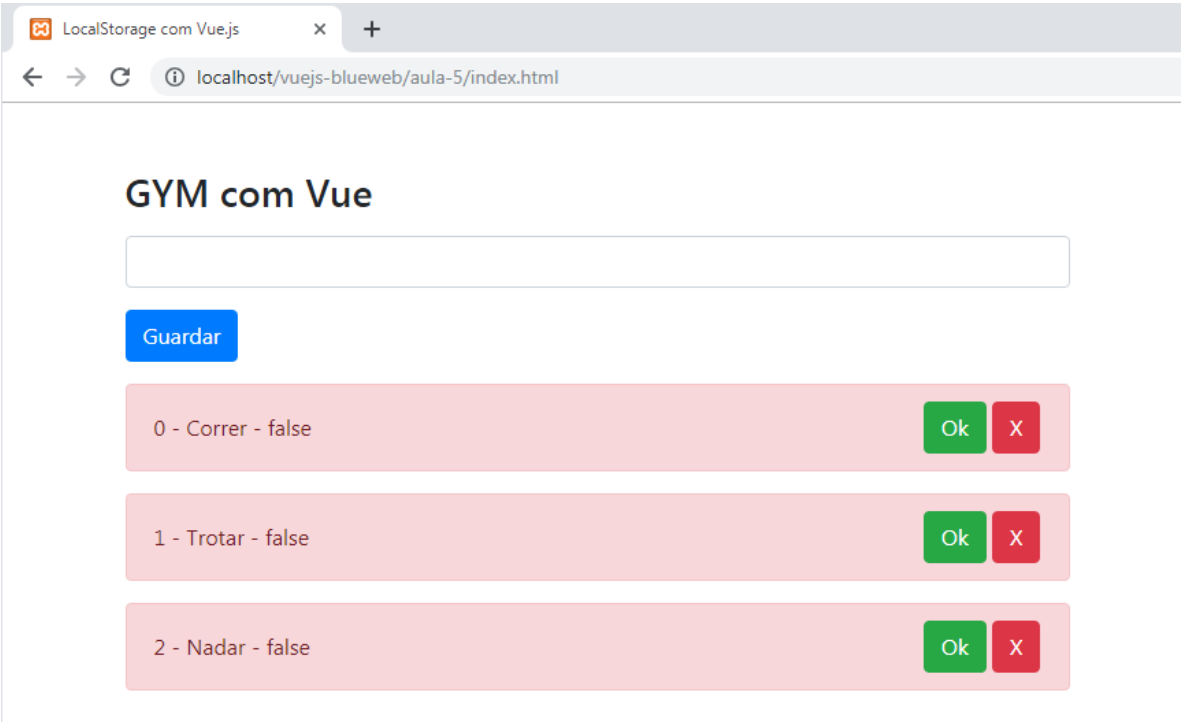
Manifest
Service Workers
Clear storage

Storage

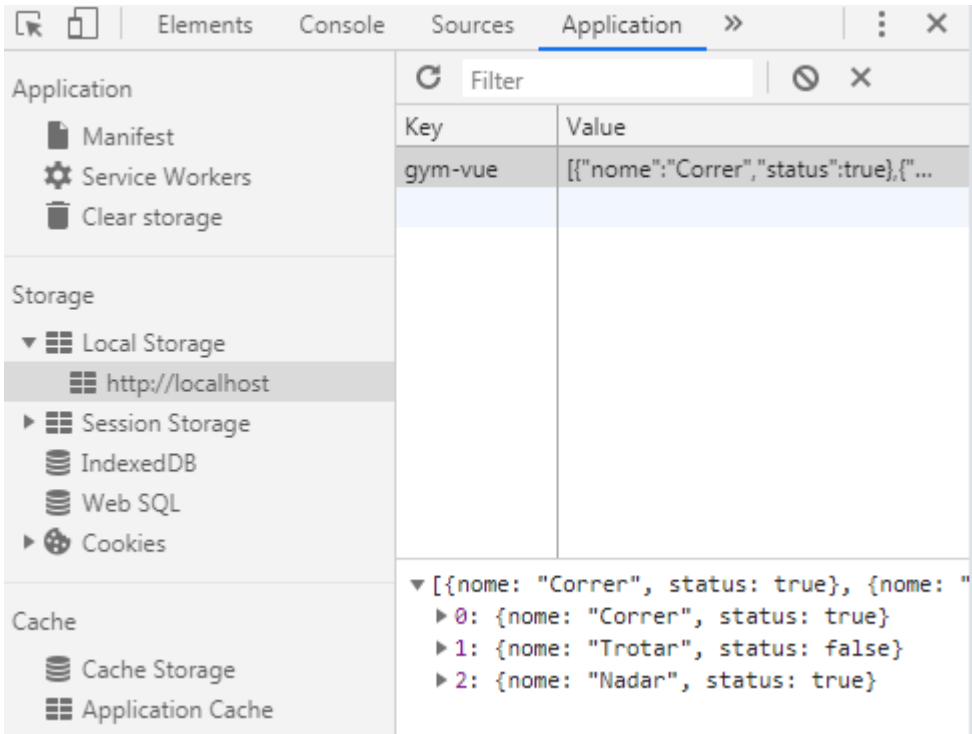
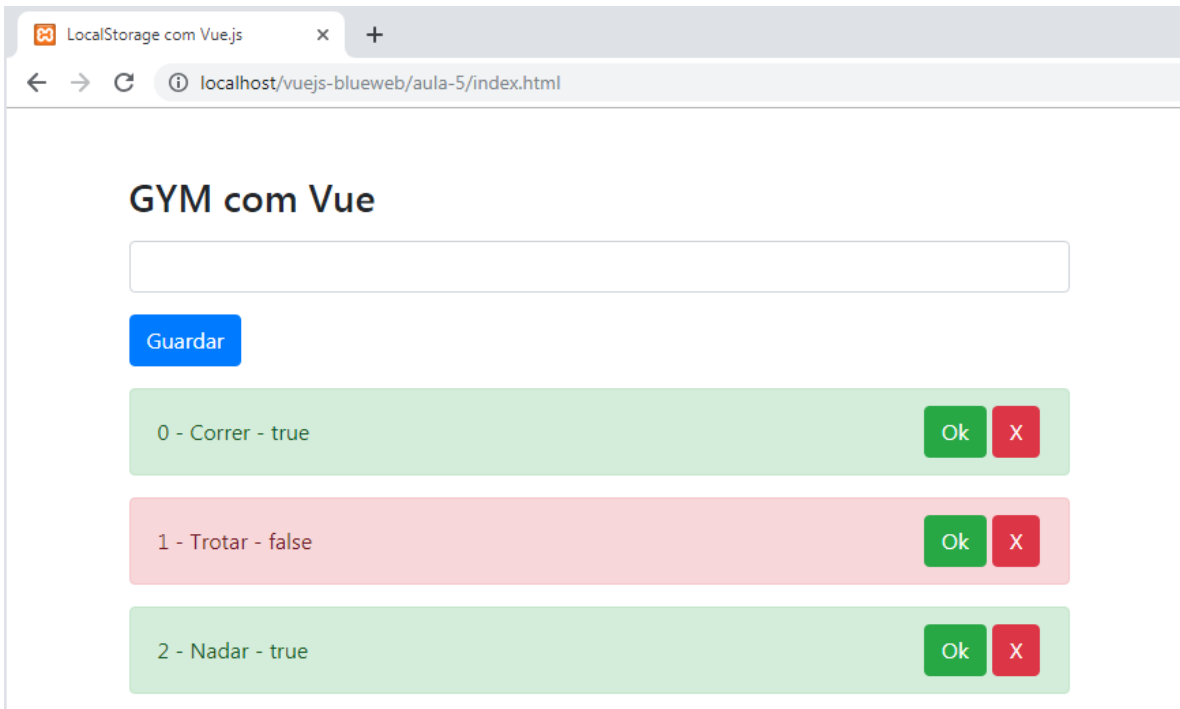
Local Storage
http://localhost

Key	Value
gym-vue	[]

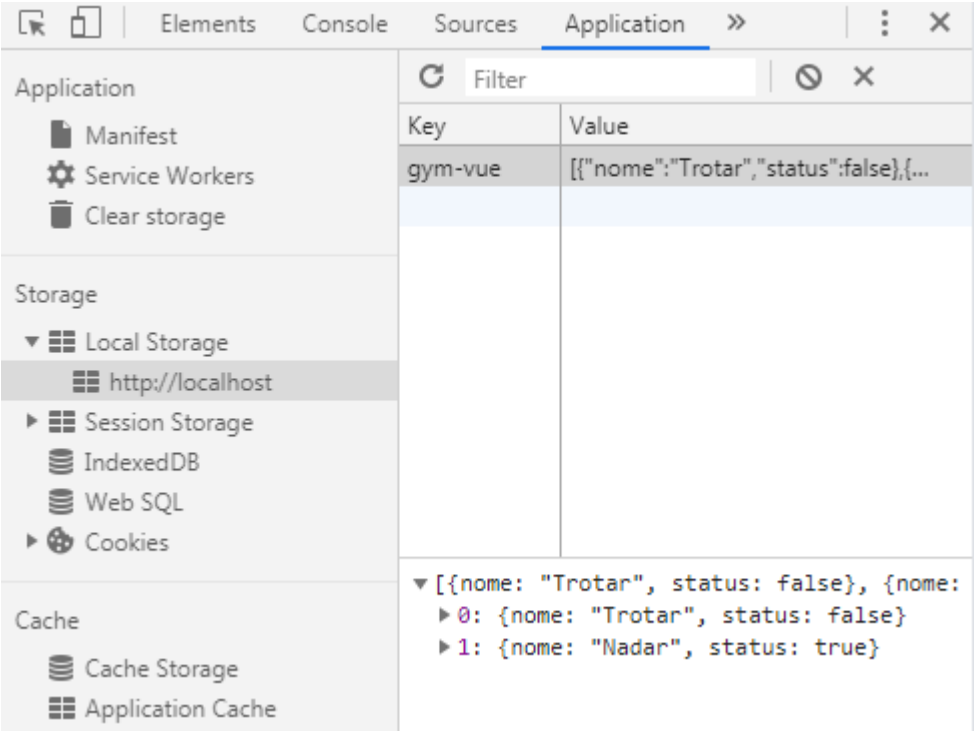
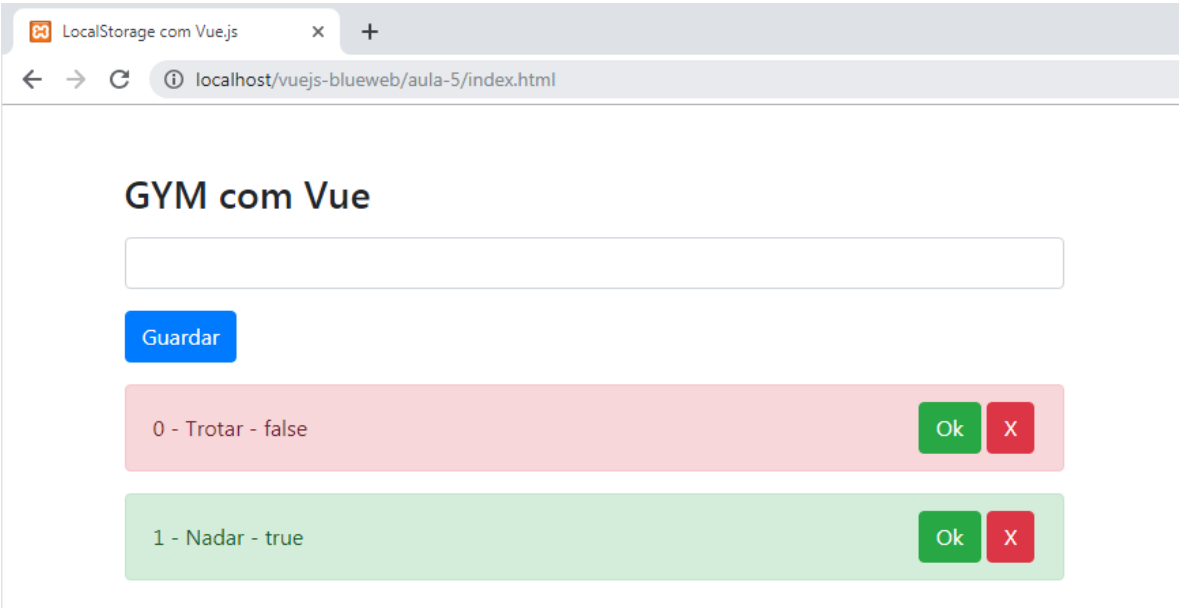
Agregando tareas



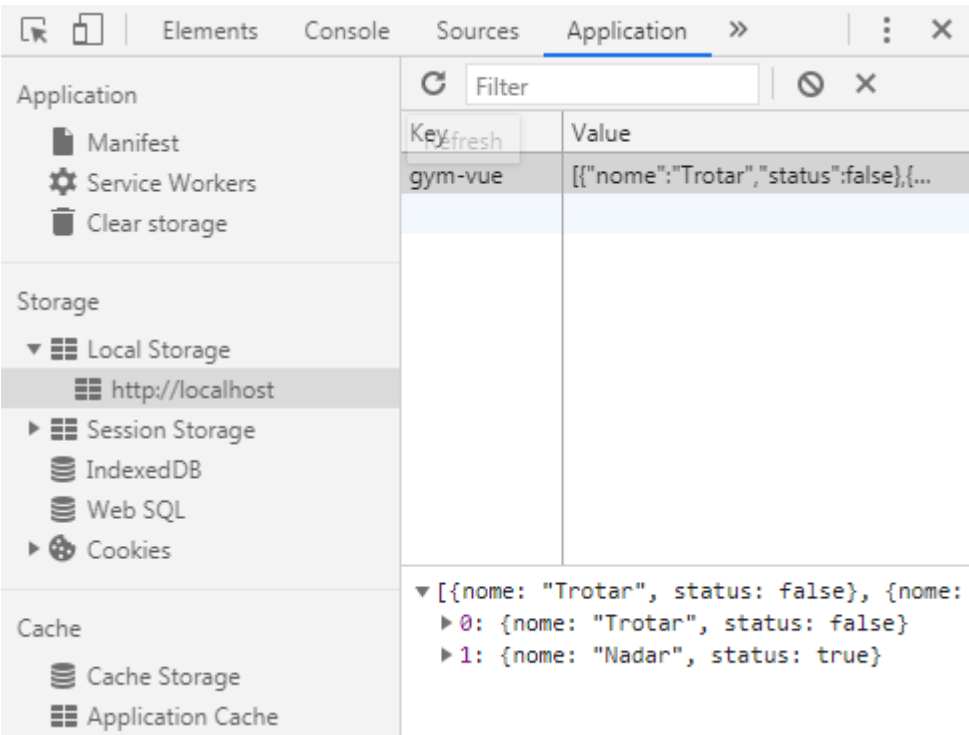
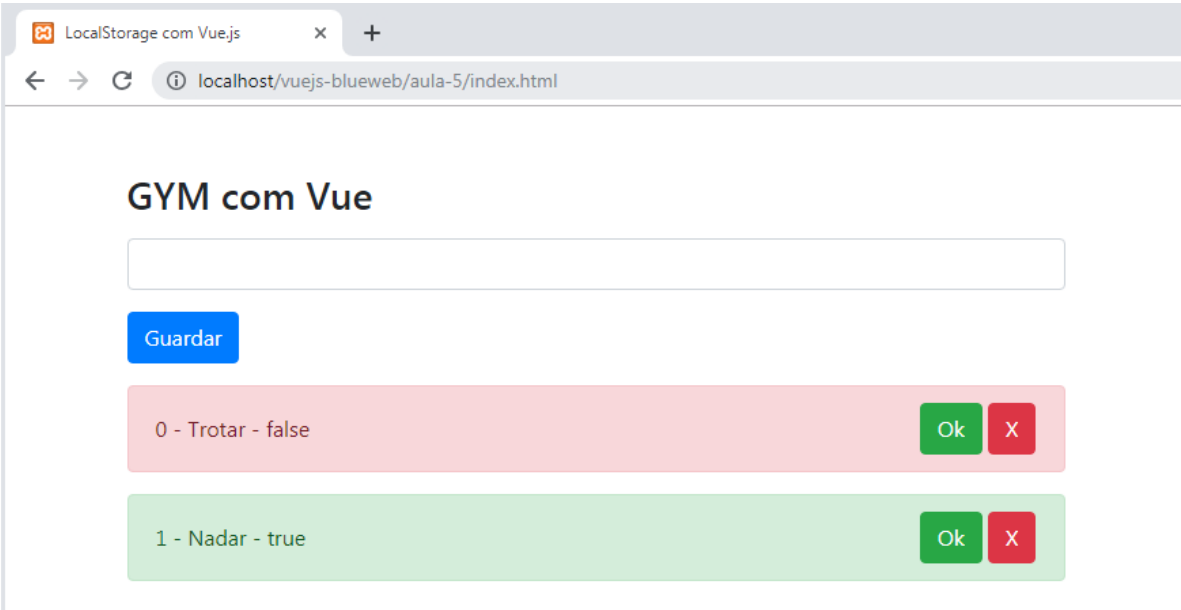
Tarefas concluídas



Eliminando tarefas



Atualizando a página



Aula 6 - Propriedades computadas

localhost/vuejs-blueweb/aula-6/index.html

```
<!doctype html>
<html lang="pt-br">

<head>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

  <!-- Bootstrap CSS -->
  <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" rel="stylesheet">
  <title>Propriedades computadas</title>
</head>

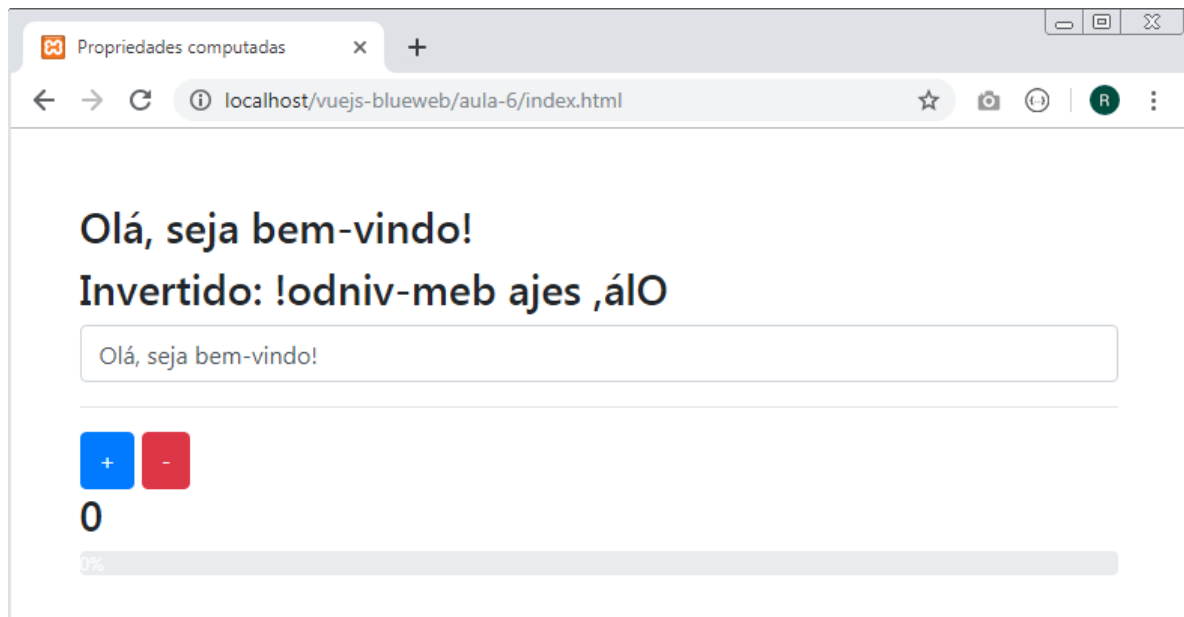
<body>
  <div id="app" class="container mt-5">
    <h3>{{mensagem}}</h3>
    <h3>Invertido: {{invertido}}</h3>
    <input type="text" class="form-control" v-model="mensagem">
    <hr>
    <button class="btn btn-primary" @click="contador++">+</button>
    <button class="btn btn-danger" @click="contador--">-</button>
    <h3>{{contador}}</h3>
    <div class="progress">
      <div class="progress-bar" role="progressbar" :style="{width: contador + '%'}" aria-
        valuenow="25" aria-valuemin="0" aria-valuemax="100" :class="cor">{{contador}}%</div>
    </div>
  </div>

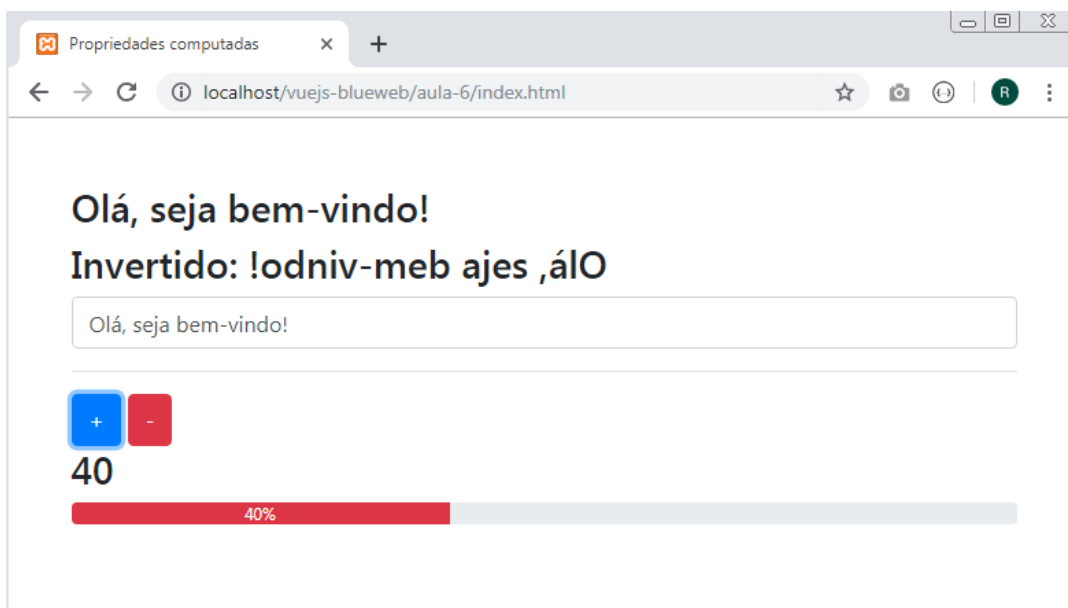
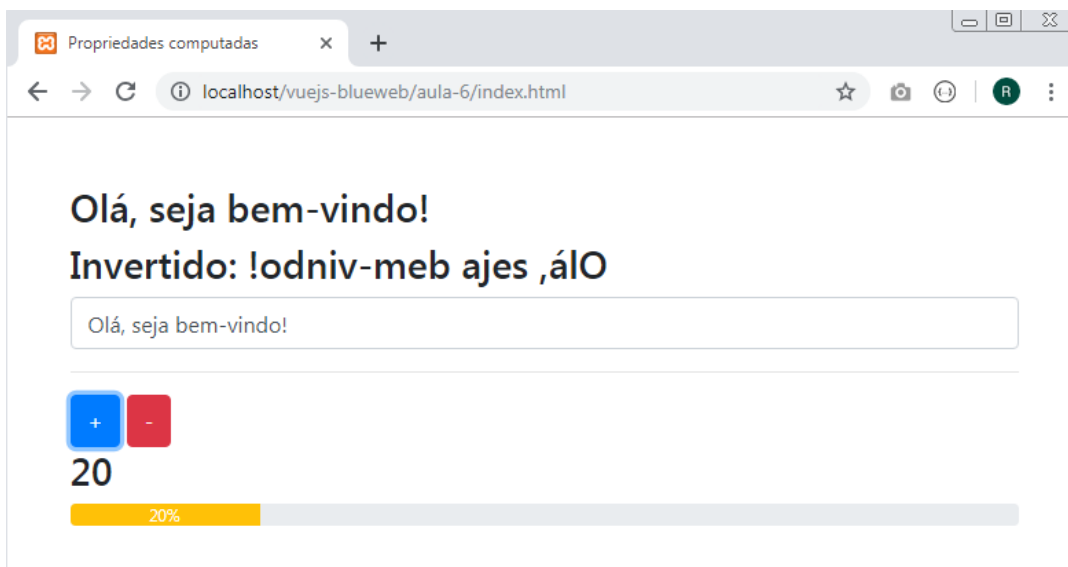
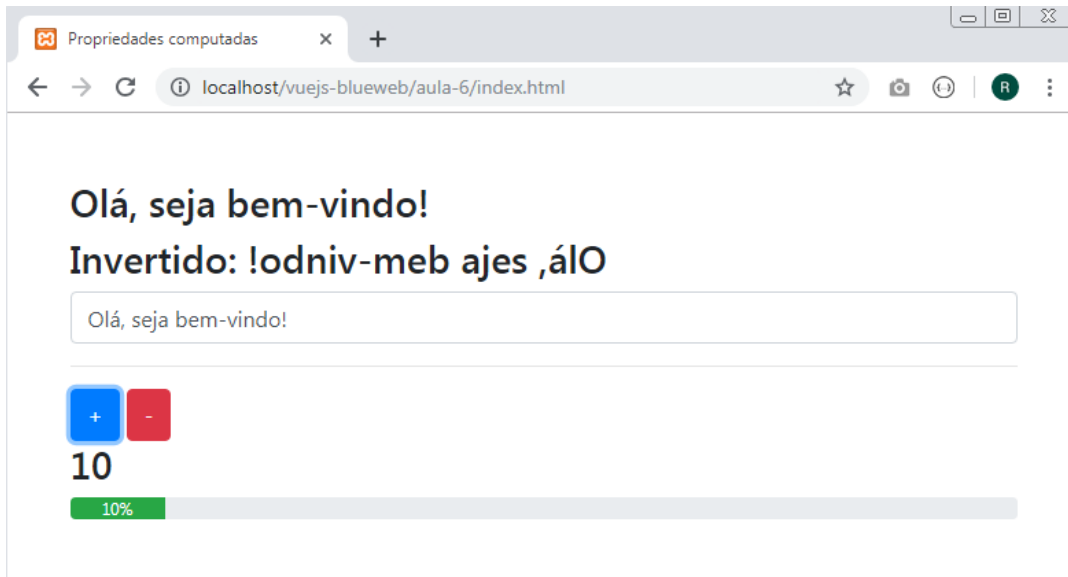
  <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
  <script src="app.js"></script>
</body>

</html>
```

aula-6/app.js

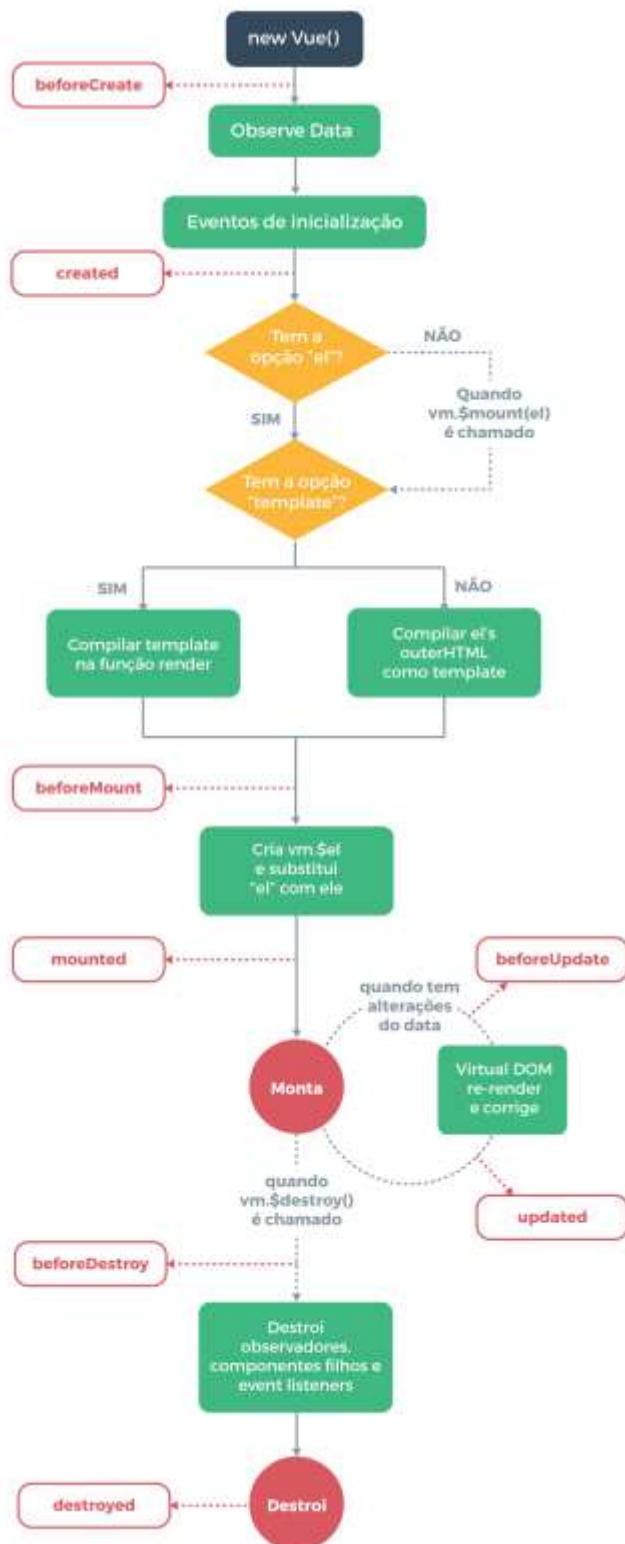
```
const app = new Vue({
  el: '#app',
  data: {
    mensagem : 'Olá, seja bem-vindo!',
    contador: 0
  },
  computed: {
    invertido(){
      return this.mensagem.split('').reverse().join('');
    },
    cor(){
      return {
        'bg-success' : this.contador <= 10,
        'bg-warning' : this.contador > 10 && this.contador <= 20,
        'bg-danger' : this.contador > 20
      }
    }
  }
});
```





Aula 7 - Lifecycle

Ciclo de vida de Vue



localhost/vuejs-blueweb/aula-7/index.html

```
<!doctype html>
<html lang="pt-br">

<head>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

  <!-- Bootstrap CSS -->
  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">
  <title>Lyfecicle</title>
</head>

<body>
  <div id="app" class="container mt-5">
    <h3>{{titulo}}</h3>
    <button class="btn btn-primary mt-2" @click="titulo = 'título modificado'">Alterar</button>
    <button class="btn btn-danger mt-2" @click="destruir()">Destruir</button>
  </div>

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

</html>
```

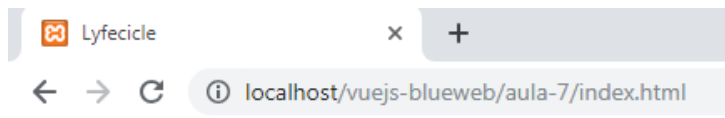
aula-7/app.js

```
const app = new Vue({
  el: '#app',
  data: {
    titulo: 'Vue.js - Ciclo de vida'
  },
  beforeCreate() {
    console.log('beforeCreate');
  },
  created(){
    console.log('created');
  },
  beforeMount(){
    console.log('beforeMount');
  },
  mounted(){
    console.log('mounted');
  },
}
```

```

beforeUpdate(){
  console.log('beforeUpdate');
},
updated(){
  console.log('updated');
},
beforeDestroy(){
  console.log('beforeDestroy');
},
destroyed(){
  console.log('destroyed');
},
methods: {
  destruir() {
    this.$destroy();
  }
}
});

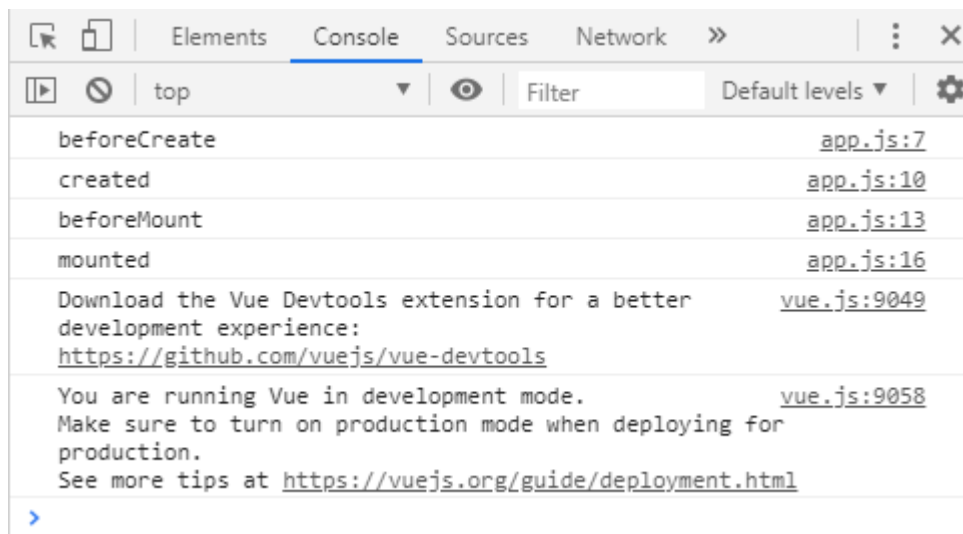
```

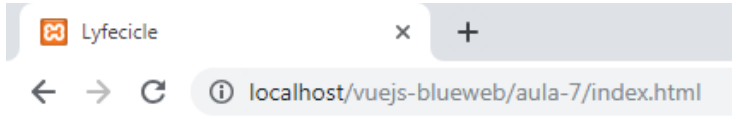


Vue.js - Ciclo de vida

Alterar

Destruir

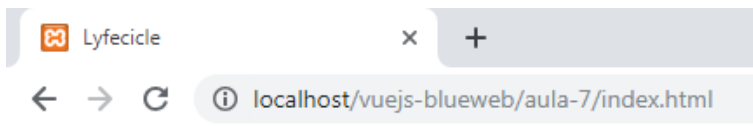
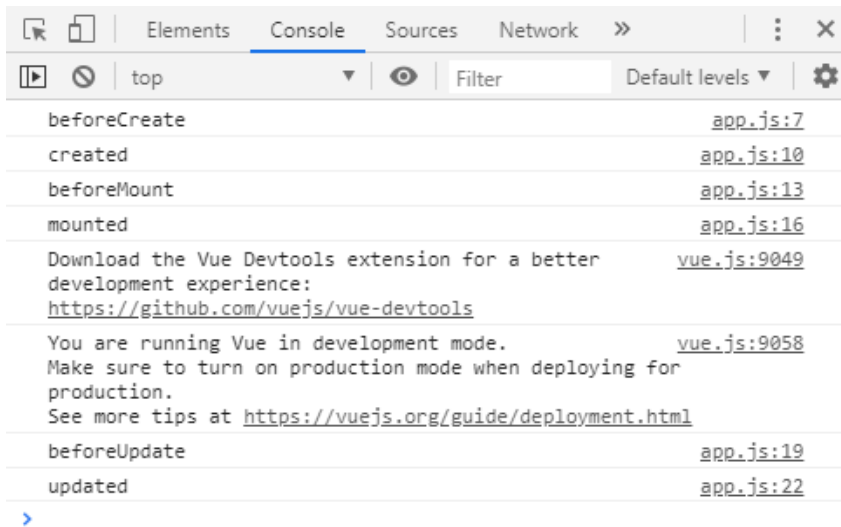




título modificado

Alterar

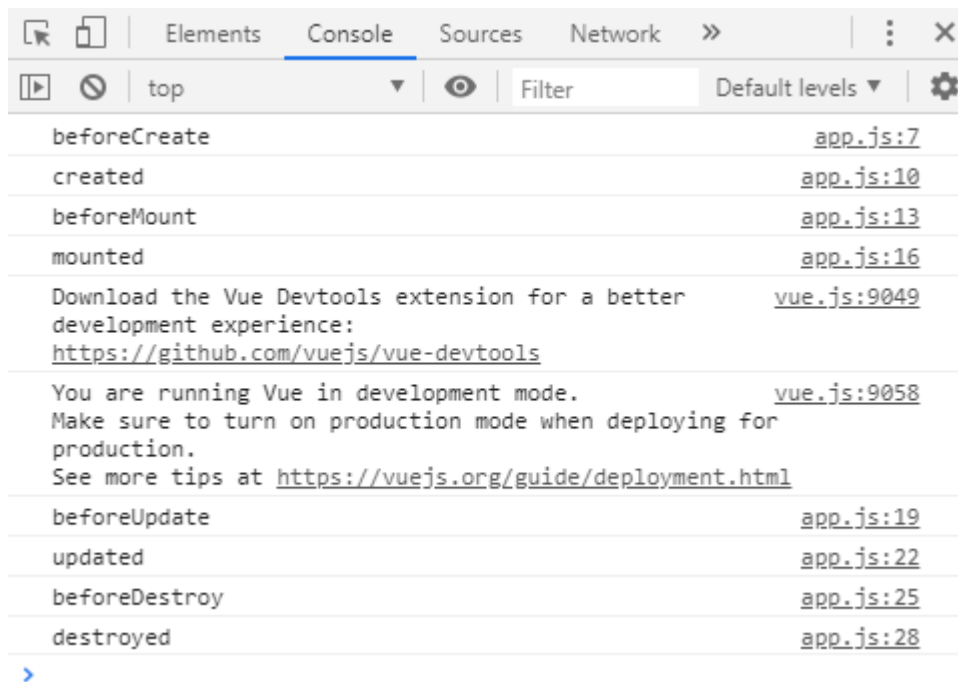
Destruir



título modificado

Alterar

Destruir



Aula 8 - Componentes básicos

Instalar no Visual Studio Code, a extensão:

es6-string-html

localhost/vuejs-blueweb/aula-8/index.html

```
<!doctype html>
<html lang="pt-br">

<head>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

  <!-- Bootstrap CSS -->
  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">
  <title>Componentes básicos</title>
</head>

<body>
  <div id="app" class="container mt-5">
    <saudacoes></saudacoes>
    <contador></contador>
    <contador></contador>
  </div>

  <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
  <script src="componentes/saudacoes.js"></script>
  <script src="componentes/contador.js"></script>
  <script src="app.js"></script>

</body>

</html>
```

aula-8/app.js

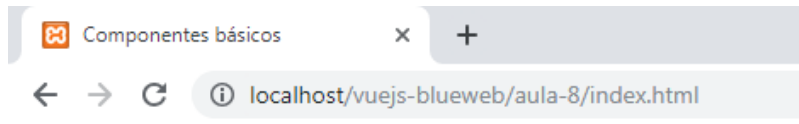
```
const app = new Vue({
  el: '#app'
});
```

aula-8/componentes/saudacoes.js

```
Vue.component('saudacoes',{
  template: `//html
  `
  <div>
    <h1>{{saudacoes}}</h1>
    <h3>Componentes básicos</h3>
  </div>
  `
  ,
  data(){
    return {
      saudacoes: 'Bem vindo a Vue.js!'
    }
  }
});
```

aula-8/componentes/contador.js

```
Vue.component('contador',{
  template: `//html
  `
  <div>
    <h3>{{numero}}</h3>
    <button class="btn btn-primary" @click="numero++">+</button>
  </div>
  `
  ,
  data(){
    return {
      numero: 0
    }
  }
});
```



Bem vindo a Vue.js!

Componentes básicos

0



0



Bem vindo a Vue.js!

Componentes básicos

7



3



Aula 9 - Comunicação entre componentes

Parte 1 - De Pai para Filho

localhost/vuejs-blueweb/aula-9/index.html

```
<!doctype html>
<html lang="pt-br">

<head>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

  <!-- Bootstrap CSS -->
  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">
  <title>Comunicação entre Componentes</title>
</head>

<body>
  <div id="app" class="container mt-5">
    <pai></pai>
  </div>

  <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
  <script src="componentes/pai.js"></script>
  <script src="componentes/filho.js"></script>
  <script src="app.js"></script>

</body>

</html>
```

aula-9/app.js

```
const app = new Vue({
  el: '#app'
});
```

aula-9/componentes/pai.js

```
Vue.component('pai',{
  template: `//html
  `
  ,
  <div class="p-5 bg-dark text-white">
    <h2>Componente Pai: {{numeroPai}}</h2>
    <button @click="numeroPai++" class="btn btn-danger mb-3">+</button>
    <filho :numero="numeroPai"></filho>
  </div>
  ,
  data(){
    return {
      numeroPai: 0
    }
  }
})
```

aula-9/componentes/filho.js

```
Vue.component('filho',{
  template: `//html
  `
  ,
  props: ['numero']
})
```



Componente Pai: 7

+

Componente Filho: 7

Aula 10 - Comunicação entre componentes

Parte 2 - De Filho para Pai

localhost/vuejs-blueweb/aula-10/index.html

```
<!doctype html>
<html lang="pt-br">

<head>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

  <!-- Bootstrap CSS -->
  <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" rel="stylesheet">
  <title>Comunicação entre Componentes</title>
</head>

<body>
  <div id="app" class="container mt-5">
    <pai></pai>
  </div>

  <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
  <script src="componentes/pai.js"></script>
  <script src="componentes/filho.js"></script>
  <script src="app.js"></script>

</body>

</html>
```

aula-10/app.js

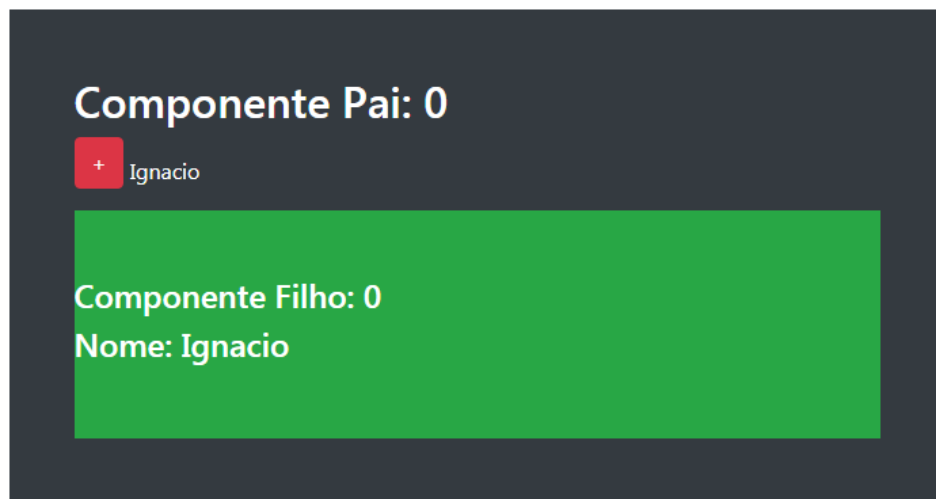
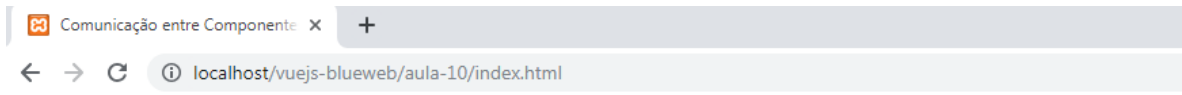
```
const app = new Vue({
  el: '#app'
});
```


aula-10/componentes/pai.js

```
Vue.component('pai',{
  template: `//html
  `
  <div class="p-5 bg-dark text-white">
    <h2>Componente Pai: {{numeroPai}}</h2>
    <button @click="numeroPai++" class="btn btn-danger mb-3">+</button>
    {{nomePai}}
    <filho :numero="numeroPai" @nomeFilho="nomePai = $event"></filho>
  </div>
  `
  ,
  data(){
    return {
      numeroPai: 0,
      nomePai: ""
    }
  }
})
```

aula-10/componentes/filho.js

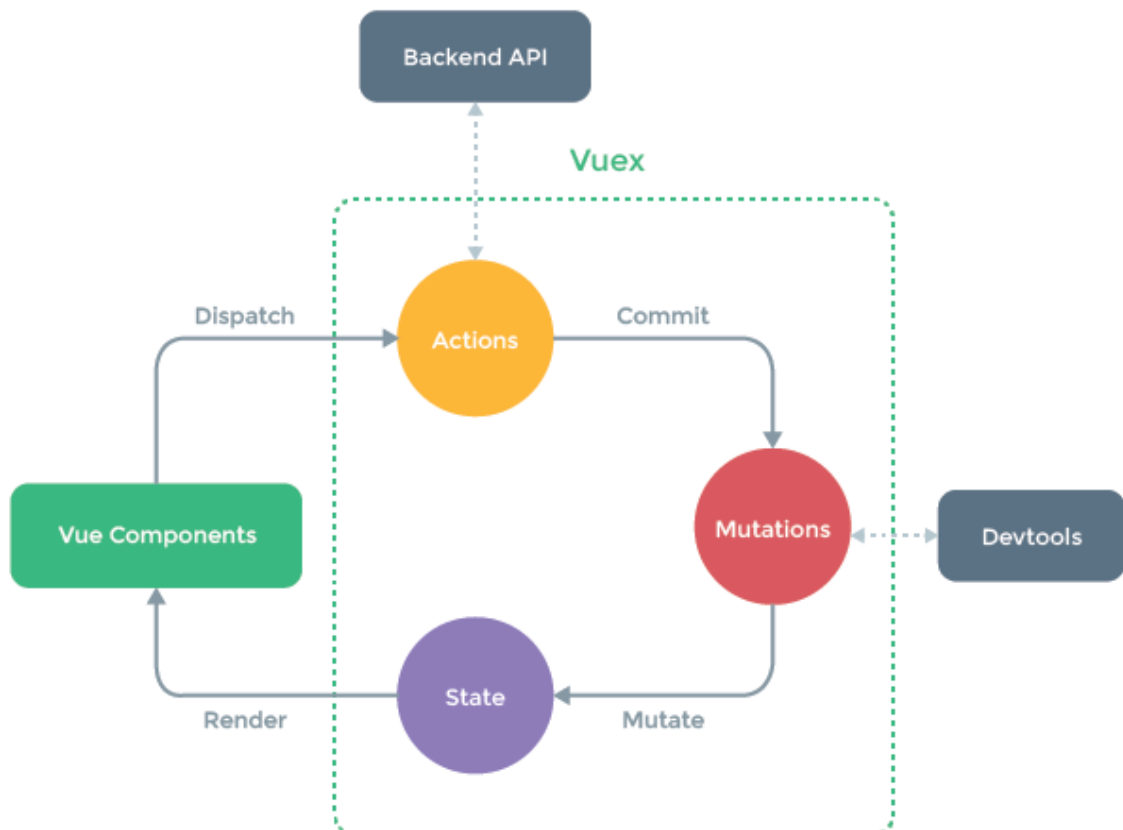
```
Vue.component('filho',{
  template: `//html
  `
  ,
  props: ['numero'],
  data(){
    return {
      nome: 'Ignacio'
    }
  },
  mounted(){
    this.$emit('nomeFilho', this.nome);
  }
})
```



Aula 11 - Introdução a Vuex

Inserir na página index.html, abaixo do script do Vue:

```
<script src="https://unpkg.com/vuex@3.1.0/dist/vuex.js"></script>
```



localhost/vuejs-blueweb/aula-11/index.html

```
<!doctype html>
<html lang="pt-br">

<head>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

  <!-- Bootstrap CSS -->
  <link                                     rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">
  <title>Vuex</title>
</head>

<body>
  <div id="app" class="container mt-5">
    <titulo></titulo>
  </div>

  <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
  <script src="https://unpkg.com/vuex@3.1.0/dist/vuex.js"></script>

  <script src="app.js"></script>

</body>

</html>
```

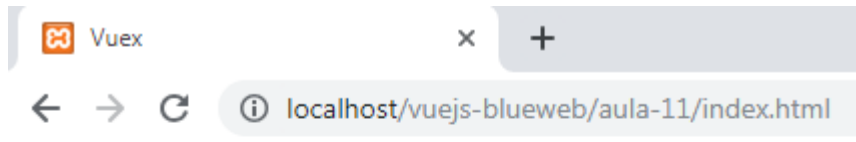
aula-11/app.js

```
Vue.component('titulo', {
  template: `//html
  `
  <div>
    <h1>numero {{$store.state.numero}}</h1>
    <filho></filho>
  </div>
  `
});
```

```
Vue.component('filho', {
  template: `//html
  `
  <div>
    <button class="btn btn-primary" @click="$store.commit('aumentar')">+</button>
    <h1>numero {{$store.state.numero}}</h1>
  </div>
  `
});
```

```
const store = new Vuex.Store({
  state: {
    numero: 10
  },
  mutations: {
    aumentar (state) {
      state.numero++
    }
  }
});
```

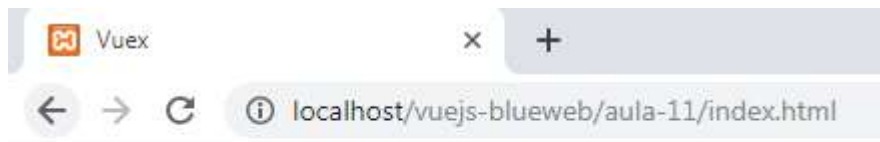
```
const app = new Vue({
  el: '#app',
  store: store
});
```



numero 10



numero 10



numero 13



numero 13

Aula 12 - mapState com Vuex

localhost/vuejs-blueweb/aula-12/index.html

```
<!doctype html>
<html lang="pt-br">

<head>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

  <!-- Bootstrap CSS -->
  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">
  <title>mapState com Vuex</title>
</head>

<body>
  <div id="app" class="container mt-5">
    <titulo></titulo>
  </div>

  <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
  <script src="https://unpkg.com/vuex@3.1.0/dist/vuex.js"></script>

  <script src="app.js"></script>

</body>

</html>
```

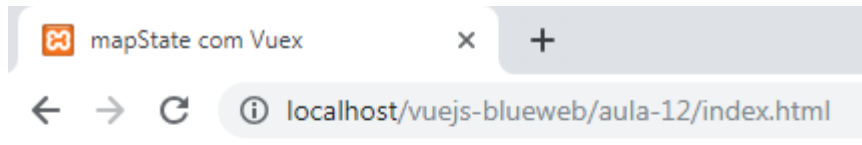
aula-12/app.js

```
Vue.component('titulo', {
  template: `//html
  `
  <div>
    <h1>numero {{numero}}</h1>
    <filho></filho>
  </div>
  `
  ,
  computed: {
    ...Vuex.mapState(['numero'])
  }
});
```

```
Vue.component('filho', {
  template: `//html
  `
  <div>
    <button class="btn btn-primary" @click="$store.commit('aumentar')">+</button>
    <h1>numero {{ $store.state.numero }}</h1>
  </div>
  `
  ,
});
```

```
const store = new Vuex.Store({
  state: {
    numero: 10
  },
  mutations: {
    aumentar (state) {
      state.numero++
    }
  }
});
```

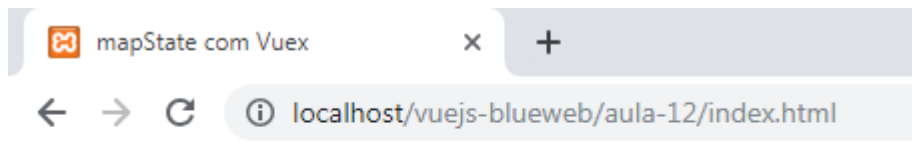
```
const app = new Vue({
  el: '#app',
  store: store
});
```

numero 10



numero 10



numero 13



numero 13

Aula 13 - mapMutations com Vuex

localhost/vuejs-blueweb/aula-13/index.html

```
<!doctype html>
<html lang="pt-br">

<head>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

  <!-- Bootstrap CSS -->
  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">
  <title>mapMutation com Vuex</title>
</head>

<body>
  <div id="app" class="container mt-5">
    <titulo></titulo>
  </div>

  <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
  <script src="https://unpkg.com/vuex@3.1.0/dist/vuex.js"></script>

  <script src="app.js"></script>

</body>

</html>
```

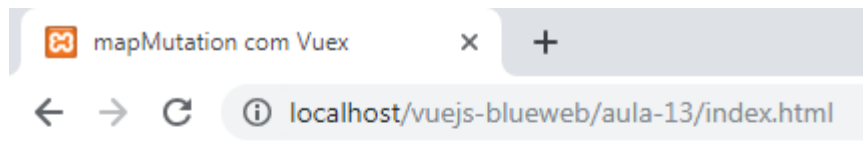
aula-13/app.js

```
Vue.component('titulo', {
  template: `//html
  <div>
    <h1>numero {{numero}}</h1>
    <filho></filho>
  </div>`,
  computed: {
    ...Vuex.mapState(['numero'])
  }
});
```

```
Vue.component('filho', {
  template: `//html
  <div>
    <button class="btn btn-primary" @click="aumentar">+</button>
    <button class="btn btn-primary" @click="diminuir(2)">-</button>
    <h1>numero {{numero}}</h1>
  </div>`,
  computed: {
    ...Vuex.mapState(['numero'])
  },
  methods: {
    ...Vuex.mapMutations(['aumentar', 'diminuir']),
  }
});
```

```
const store = new Vuex.Store({
  state: {
    numero: 10
  },
  mutations: {
    aumentar (state) {
      state.numero++
    },
    diminuir (state,n) {
      state.numero -= n
    }
  }
});
```

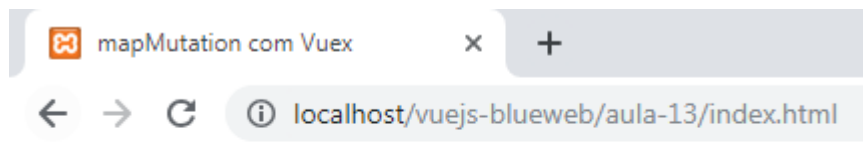
```
const app = new Vue({
  el: '#app',
  store: store
});
```



numero 10



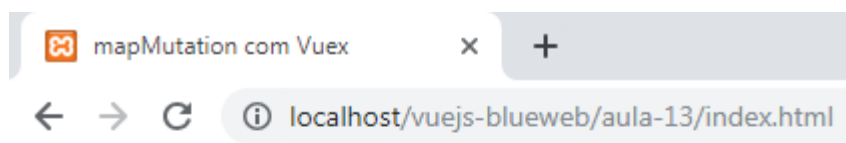
numero 10



numero 8



numero 8



numero 9



numero 9

Aula 14 - mapAction com Vuex

localhost/vuejs-blueweb/aula-14/index.html

```
<!doctype html>
<html lang="pt-br">

<head>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

  <!-- Bootstrap CSS -->
  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">
  <title>mapAction com Vuex</title>
</head>

<body>
  <div id="app" class="container mt-5">
    <titulo></titulo>
  </div>

  <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
  <script src="https://unpkg.com/vuex@3.1.0/dist/vuex.js"></script>

  <script src="app.js"></script>

</body>

</html>
```

aula-13/cursos.json

```
[
  {"nome": "HTML", "id": 1},
  {"nome": "CSS", "id": 2},
  {"nome": "JAVASCRIPT", "id": 3}
]
```

aula-14/app.js

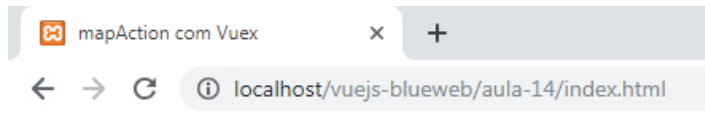
```
Vue.component('titulo', {
  template: `//html
  `
  <div>
    <h1>numero {{numero}}</h1>
    <filho></filho>
  </div>
  `
  ,
  computed: {
    ...Vuex.mapState(['numero'])
  }
});
```

```
Vue.component('filho', {
  template: `//html
  `
  <div>
    <button class="btn btn-primary" @click="aumentar">+</button>
    <button class="btn btn-primary" @click="diminuir(2)">-</button>
    <button class="btn btn-primary" @click="obterCursos">Obter cursos</button>
    <h1>numero {{numero}}</h1>
    <ul v-for="item of cursos">
      <li>{{item.nome}}</li>
    </ul>
  </div>
  `
  ,
  computed: {
    ...Vuex.mapState(['numero', 'cursos'])
  },
  methods: {
    ...Vuex.mapMutations(['aumentar', 'diminuir']),
    ...Vuex.mapActions(['obterCursos'])
  }
});
```

```
const store = new Vuex.Store({
  state: {
    numero: 10,
    cursos: []
  },
  mutations: {
    aumentar (state) {
      state.numero++
    },
  },
});
```

```
diminuir (state,n) {  
  state.numero -= n  
},  
preencherCursos(state, cursosAcao){  
  state.cursos = cursosAcao  
}  
},  
actions: {  
  obterCursos: async function({commit}){  
    const data = await fetch('cursos.json');  
    const cursos = await data.json();  
    commit('preencherCursos', cursos);  
  }  
}  
});
```

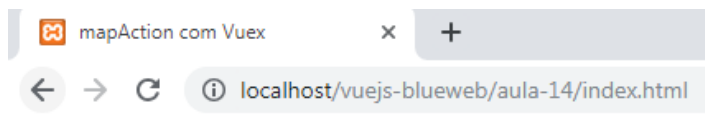
```
const app = new Vue({  
  el: '#app',  
  store: store  
});
```



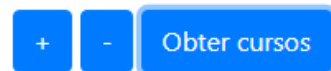
numero 10



numero 10



numero 10



numero 10

- HTML
- CSS
- JAVASCRIPT