

2021/06/14 - AULA 10.2

Laboratório de Sistemas Computacionais Complexos

<https://uclab.xyz/sistemas-complexos-2021-aula10-2>



Renato Cordeiro Ferreira
renatocf@ime.usp.br



João Francisco Daniel
joaofran@ime.usp.br



Alfredo Goldman
gold@ime.usp.br



Thatiane de Oliveira Rosa
thatiane@ime.usp.br

Em caso de dúvidas

Acessem [slido.com](https://www.slido.com) com #complexos

ou



Agenda

Tema da aula:

Projetos VueJS

1. Single File Components
2. Vue CLI
3. Vue Router
4. Vuex

VueJS: The Progressive JS Framework

```
<template>
  <div>
    <div v-if="active">{{ msg }}</div>
    <button v-on:click="toggle">Mostrar segredo!</button>
  </div>
</template>

<script>
export default {
  props: ["msg"],
  data: function() { return { active: true } },
  methods: {
    toggle: function() { this.active = !this.active; }
  }
}
</script>

<style>
/* CSS relacionado ao componente vai aqui */
</style>
```

VueJS

- Para aumentar a clareza da criação de componentes, é possível criar a abstração de **Single File Components (SFC)** com arquivos .vue que integram o HTML + CSS + JS associados a um componente
- Precisam ser **pré-processados** em HTML, CSS e JS puros com ajuda da [Vue CLI](#)

VueJS: The Progressive JS Framework

```
.
├── babel.config.js
├── package.json
├── package-lock.json
├── public
│   ├── favicon.ico
│   └── index.html
├── README.md
├── src
│   ├── main.js
│   ├── App.vue
│   ├── assets
│   │   └── logo.png
│   ├── components
│   │   └── HelloWorld.vue
│   ├── router
│   │   └── index.js
│   ├── store
│   │   └── index.js
│   └── views
│       ├── About.vue
│       └── Home.vue
```

Vue CLI

- Cria um projeto para **aplicação** **VueJS** (SPA + CSR) configurada para executar **localmente** utilizando NodeJS
- A pasta **public** contém arquivo de entrada do SPA **index.html** e imagem de ícone **favicon.ico**
- **package.json** e **package-lock.json** contêm dependências NodeJS

VueJS: The Progressive JS Framework

```
.
├── babel.config.js
├── package.json
├── package-lock.json
├── public
│   ├── favicon.ico
│   └── index.html
├── README.md
├── src
│   ├── main.js
│   ├── App.vue
│   ├── assets
│   │   └── logo.png
│   ├── components
│   │   └── HelloWorld.vue
│   ├── router
│   │   └── index.js
│   ├── store
│   │   └── index.js
│   └── views
│       ├── About.vue
│       └── Home.vue
```

Vue CLI

- `babel.config.js` configura o [Babel](#) para permitir o uso do JS com versão mais avançada
- `main.js` contém a configuração do objeto Vue, sendo importado pelo `index.html`
- `App.vue` é o [single file component](#) raiz, montado pelo `main.js` para a aplicação Vue

VueJS: The Progressive JS Framework

```
.
├── babel.config.js
├── package.json
├── package-lock.json
├── public
│   ├── favicon.ico
│   └── index.html
├── README.md
├── src
│   ├── main.js
│   ├── App.vue
│   ├── assets
│   │   └── logo.png
│   ├── components
│   │   └── HelloWorld.vue
│   ├── router
│   │   └── index.js
│   ├── store
│   │   └── index.js
│   └── views
│       ├── About.vue
│       └── Home.vue
```

Vue CLI

- A pasta **assets** contém arquivos que serão servidos sem processamento
- A pasta **components** contém single file components úteis para a aplicação
- A pasta **views** contém componentes que representam **páginas da aplicação**

VueJS: The Progressive JS Framework

```
import Vue from "vue";
import VueRouter from "vue-router";
import Home from "../views/Home.vue";
```

```
Vue.use(VueRouter);
```

```
const routes = [
  {
    path: "/",
    name: "Home",
    component: Home
  },
  {
    path: "/about",
    name: "About",
  }
];
```

```
const router = new VueRouter({
  routes
});
```

```
export default router;
```

Vue Router

- A pasta `router` contém a configuração do [Vue Router](#), uma biblioteca que permite fazer o `roteamento` de caminhos na URL para outros componentes que representam páginas da aplicação
- O Vue Router é inserido como um `plugin` do Vue que pode ser acessada com `this.$router`

VueJS: The Progressive JS Framework

```
import Vue from "vue";
import Vuex from "vuex";

Vue.use(Vuex);

export default new Vuex.Store({
  // Armazena dados de forma global
  state: {},

  // Armazena métodos para alterar dados (setters)
  // que não podem ser assíncronos
  mutations: {},

  // Armazena métodos para alterar dados (setters)
  // que podem ser assíncronos
  actions: {},

  // Define "namespaces" para fazer mais separação
  // entre conjuntos de state, mutation e actions relacionados
  modules: {}
});
```

Vuex

- A pasta **store** contém a configuração do [Vuex Store](#), uma biblioteca que permite fazer o **armazenamento** de dados global para troca de dados **transversal** entre componentes irmãos/primos
- A Vuex Store é inserida como um **plugin** do Vue, que pode ser acessada com **this.\$store**

VueJS: The Progressive JS Framework

```
import Vue from "vue";
import Vuex from "vuex";

Vue.use(Vuex);

export default new Vuex.Store({
  // Armazena dados de forma global
  state: {},

  // Armazena métodos para alterar dados (setters)
  // que não podem ser assíncronos
  mutations: {},

  // Armazena métodos para alterar dados (setters)
  // que podem ser assíncronos
  actions: {},

  // Define "namespaces" para fazer mais separação
  // entre conjuntos de state, mutation e actions relacionados
  modules: {}
});
```

Vuex

- A pasta **store** contém a configuração do [Vuex Store](#), uma biblioteca que permite fazer o **armazenamento** de dados global para troca de dados **transversal** entre componentes irmãos/primos
- A Vuex Store é inserida como um **plugin** do Vue, que pode ser acessada com **this.\$store**

VueJS: The Progressive JS Framework

```
import Vue from "vue";
import Vuex from "vuex";

Vue.use(Vuex);

export default new Vuex.Store({
  // Armazena dados de forma global
  state: {},

  // Armazena métodos para alterar dados (setters)
  // que não podem ser assíncronos
  mutations: {},

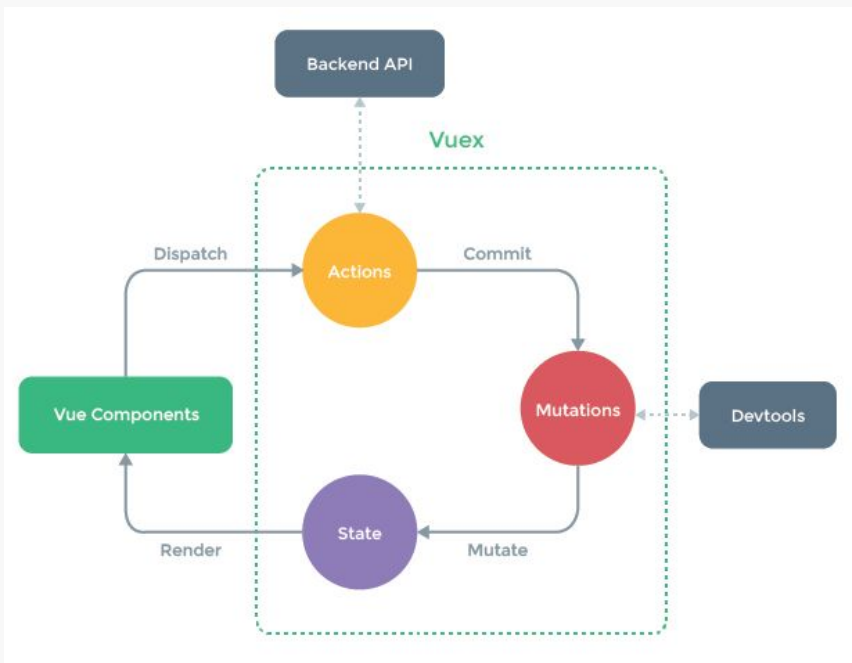
  // Armazena métodos para alterar dados (setters)
  // que podem ser assíncronos
  actions: {},

  // Define "namespaces" para fazer mais separação
  // entre conjuntos de state, mutation e actions relacionados
  modules: {}
});
```

Vuex

- O campo **state** contém os **dados** a serem armazenados pela Vuex Store
- O campo **mutations** contém **métodos** que podem alterar dados (setters), mas cujo código **não pode ser assíncrono**
- O campo **actions** contém **métodos** que podem alterar dados (setters), mas cujo código **pode ser assíncrono**
- O campo **modules** define **namespaces** para conjuntos de state + mutations + actions

VueJS: The Progressive JS Framework



VueJS

- O campo **state** contém os **dados** a serem armazenados pela Vuex Store
- O campo **mutations** contém **métodos** que podem alterar dados (setters), mas cujo código **não pode ser assíncrono**
- O campo **actions** contém **métodos** que podem alterar dados (setters), mas cujo código **pode ser assíncrono**
- O campo **modules** define **namespaces** para conjuntos de state + mutations + actions

Licença

Estes slides são concedidos sob uma Licença Creative Commons. Sob as seguintes condições: **Atribuição, Uso Não-Comercial e Compartilhamento pela mesma Licença**

Mais detalhes sobre essa licença em: creativecommons.org/licenses/by-nc-sa/3.0/

Créditos

Imagens usadas nesta apresentação são provenientes de: [freepik.com](https://www.freepik.com)

Frequência



Senha do Estudante: **gnpgqo**

2021/06/14 - AULA 10.2

Laboratório de Sistemas Computacionais Complexos

<https://uclab.xyz/sistemas-complexos-2021-aula10-2>



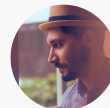
Renato Cordeiro Ferreira
renatocf@ime.usp.br



João Francisco Daniel
joaofran@ime.usp.br



Alfredo Goldman
gold@ime.usp.br



Thatiane de Oliveira Rosa
thatiane@ime.usp.br

