

2021/06/14 - AULA 10.1

Laboratório de Sistemas Computacionais Complexos

<https://uclab.xyz/sistemas-complexos-2021-aula10-1>



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:

Introdução ao VueJS

1. MPA vs. SPA
2. SSR vs. CSR
3. JavaScript no navegador
4. VueJS no browser
5. Ciclo de Vida de Componentes

Classificando Aplicações Web

Multi-Page Application (MPA)

- Aplicações Web “tradicionais”
- Servidor HTTP entrega **diferentes páginas** HTML conforme o caminho acessado
- Frameworks e bibliotecas geralmente utilizam **Server-Side Rendering (SSR)** para construir conteúdo dinâmico no **servidor** usando arquitetura **MVC**
- Exemplos: [Rails](#) (Ruby), [Django](#) (Python), [Laravel](#) (PHP), [Spring](#) (Java), etc.

Single-Page Application (SPA)

- Aplicações Web “modernas”
- Servidor HTTP entrega **uma única página** HTML para todos os caminhos acessados
- Frameworks e bibliotecas geralmente utilizam **Client-Side Rendering (CSR)** para construir conteúdo dinâmico no **navegador** usando arquitetura **MVVM**
- Exemplos: [Angular](#) (JS), [React](#) (JS), [Vue](#) (JS), [Svelte](#) (JS), [Preact](#) (JS), [Inferno](#) (JS), etc.

Classificando Aplicações Web

Server-Side Rendering (SSR)

- Maior processamento no **servidor**
- Mais simples fazer **Search Engine Optimization (SEO)**, pois os robôs indexadores dos sites de busca têm acesso direto ao HTML ao entrar na página
- Páginas podem precisar de **JavaScript no navegador** para melhorar a UX
- Páginas precisam ser entregues por um **servidor HTTP**

Client-Side Rendering (CSR)

- Maior processamento no **navegador**
- Mais complexo fazer **Search Engine Optimization (SEO)**, pois os robôs indexadores dos sites de busca precisam esperar o JavaScript renderizar o HTML
- Páginas podem precisar de **renderização prévia no servidor** para melhorar o SEO
- Páginas podem ser entregues por uma **Content Delivery Network (CDN)**

JavaScript no Navegador

```
<!DOCTYPE html>

<html>
  <head>
    <!-- Conteúdo do head -->
    <script>
      <!-- JS interno, carregado antes da criação do DOM -->
    </script>
    <!-- JS externo, carregado antes da criação do DOM -->
    <script src="script.js">
  </head>
  <body>
    <!-- Conteúdo do body -->
    <script>
      <!-- JS interno, pode manipular o DOM já criado -->
    </script>
    <!-- JS externo, pode manipular o DOM já criado -->
    <script src="script.js">
  </body>
</html>
```

JavaScript

- Linguagem de programação que permite a modificação dinâmica do HTML e CSS
- Pode ser inserido de forma **interna** ou **externa**, de modo similar ao CSS
- Pode ser inserido no **<head>** ou no **<body>** alterando quando o script é executado

VueJS: The Progressive JS Framework



VueJS

- Projeto de código aberto criado por Evan You
Existe um documentário! <https://youtu.be/OrxmtDw4pVI>
- Framework JavaScript criado para adicionar **dinamismo** às páginas Web
- Fortemente inspirado no **Angular** (versão 1) para a manipulação de HTML + CSS e no **React** para a implementação interna e bibliotecas auxiliares
- **Curva de aprendizado baixa** e documentação bem escrita disponível em <https://vuejs.org>

VueJS: The Progressive JS Framework

```
<html>
  <head>
    <!-- Versão desenvolvimento, com warnings no console -->
    <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js">
    </script>
  </head>
  <body>
    <div id="app">
      {{ message }} <!-- Notação "bigode" -->
    </div>
    <script>
      let app = new Vue({
        el: "#app",
        data: {
          message: "Hello, Vue!"
        }
      });
    </script>
  </body>
</html>
```

VueJS

- Conhecido como framework **progressivo** pois pode ser utilizado no HTML puro
- Código do framework precisa ser incluído no **<head>** para disponibilizar classe **Vue**
- Instância Vue utiliza algum elemento como **âncora** para manipular o seu DOM

VueJS: The Progressive JS Framework

```
<body>
  <div id="app">
    <ul>
      <li v-for="fruit in fruits" v-bind:key="fruit">
        {{ fruit }}
      </li>
    </ul>
  </div>
  <script>
    let app = new Vue({
      el: "#app",
      data: {
        fruits: ["abacate", "banana", "maçã", "mamão"]
      }
    });
  </script>
</body>
```

VueJS

- Permite gerar HTML com JS utilizando a propriedade especial **v-for**
- Propriedades podem ter seus valores alterados pelo Vue usando **v-bind:property**, abreviada com **:property**
- **data** contém dados cujo ciclo de vida é gerenciado pelo Vue

VueJS: The Progressive JS Framework

```
<body>
  <div id="app">
    <div v-if="active">Mostrando uma mensagem secreta</div>
    <button v-on:click="toggle">Mostrar segredo!</button>
  </div>
  <script>
    let app = new Vue({
      el: "#app",
      data: {
        active: true
      },
      methods: {
        toggle: function() {
          this.active = !this.active;
        }
      }
    });
  </script>
</body>
```

VueJS

- Permite **manipular HTML** com JS utilizando a propriedade especial **v-if**
- Eventos JS podem ser capturados pelo Vue para reações usando **v-on:event**, abreviado como **@event**
- **methods** contém funções que podem alterar dados em **data**

VueJS: The Progressive JS Framework

```
<body>
  <div id="app">
    <input type="text" v-model="message" />
    <p>Você está digitando: {{ lowercaseMessage }}</p>
  </div>
  <script>
    let app = new Vue({
      el: "#app",
      data: {
        message: ""
      },
      computed: {
        lowercaseMessage: function() {
          return this.message.toLowerCase();
        }
      }
    });
  </script>
</body>
```

VueJS

- Tem suporte a *two-way data binding*, isto é, alteração e leitura de um mesmo dado gerenciado pelo Vue usando a propriedade **v-model**
- **computed** contém variáveis cujos valores são **derivados** de outros dados de modo **lazy** (eles só são mudados quando o dado de origem é alterado)

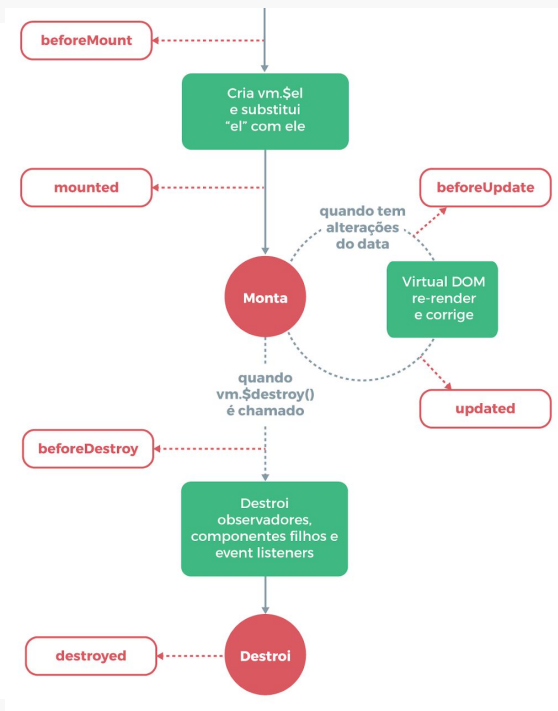
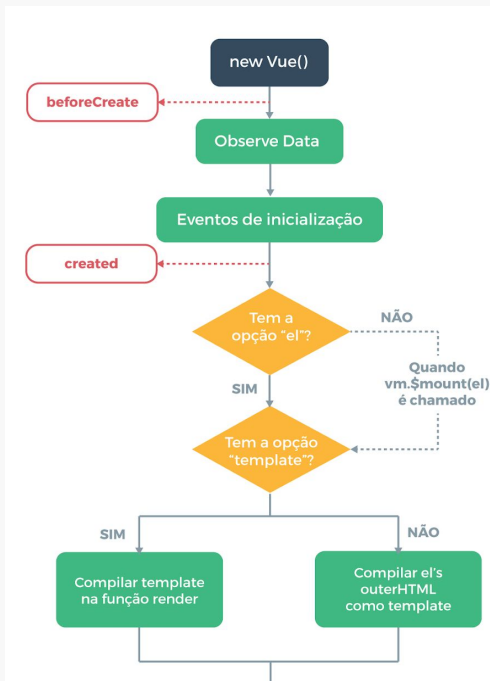
VueJS: The Progressive JS Framework

```
<body>
  <div id="app">
    <secret msg="segredo1" />
    <secret msg="segredo2" />
  </div>
  <script>
    new Vue({ el: "#app" });
    Vue.component("secret", {
      props: ["msg"],
      template: `
        <div v-if="active">{{ msg }}</div>
        <button v-on:click="toggle">Mostrar segredo!</button>
      `,
      data: function() { return { active: true } },
      methods: {
        toggle: function() { this.active = !this.active; }
      }
    });
  </script>
</body>
```

VueJS

- Principal abstração do VueJS são **componentes**, que servem para reutilizar trechos de HTML, CSS e JS associados
- Páginas podem ser vistas como **hierarquias de componentes**
- Componentes aceitam **props** que podem ser passados do componente pai para o filho

VueJS: The Progressive JS Framework



VueJS

- Componentes do Vue possuem um **ciclo de vida** administrado pelo framework
- Cada etapa desse ciclo permite associar **lifecycle hooks** ao componente: métodos que são executados no ponto específico a que eles estão associados
- Destaques: **mounted**, **updated**

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 10.1

Laboratório de Sistemas Computacionais Complexos

<https://uclab.xyz/sistemas-complexos-2021-aula10-1>



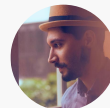
Renato Cordeiro Ferreira
renatocf@ime.usp.br



João Francisco Daniel
joaofran@ime.usp.br



Alfredo Goldman
gold@ime.usp.br



Thatiane de Oliveira Rosa
thatiane@ime.usp.br