

2021/06/17 - AULA 11.1

# Laboratório de Sistemas Computacionais Complexos

---

<https://uclab.xyz/sistemas-complexos-2021-aula11-1>

---



**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 NuxtJS

1. MPA + SSR
2. SPA + CSR
3. SPA + SSR
4. Benefícios do NuxtJS
5. Estrutura do NuxtJS
6. 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.

## 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

# Classificando Aplicações Web

## 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.

## 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)**

# Classificando Aplicações Web

## 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.

## 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**

# NuxtJS: The Intuitive Vue Framework



## NuxtJS

- Projeto de código aberto criado pelos irmãos Alexandre Chopin e Sébastien Chopin
- Framework criado usando VueJS v2  
NuxtJS baseado em v3 em progresso com release para 2021
- Permite dois modos de operação:  
Server-Side Rendering e Geração Estática
- Configuração padrão da Vuex e Vue-Router e arquitetura de plugins para facilitar instalação de bibliotecas e módulos em <https://nuxtjs.org>

# NuxtJS: The Intuitive Vue Framework

```
.
├── nuxt.config.js
├── package.json
├── package-lock.json
├── jsconfig.json
├── README.md
├── assets
│   └── style.scss
├── components
│   └── Logo.vue
├── layouts
│   └── default.vue
├── middleware
│   └── middleware.js
├── pages
│   └── index.vue
├── plugins
│   └── plugin.js
├── static
│   └── favicon.ico
├── store
│   └── state.js
```

## NuxtJS

- `nuxt.config.js` configura o projeto Nuxt incluindo módulos e plugins para executar **localmente** utilizando NodeJS
- `package.json` e `package-lock.json` contêm dependências NodeJS
- `jsconfig.json` contém **alias** para facilitar a importação de arquivos dentro dos single-file components



# NuxtJS: The Intuitive Vue Framework

```
.
├── nuxt.config.js
├── package.json
├── package-lock.json
├── jsconfig.json
├── README.md
├── assets
│   └── style.scss
├── components
│   └── Logo.vue
├── layouts
│   └── default.vue
├── middleware
│   └── middleware.js
├── pages
│   └── index.vue
├── plugins
│   └── plugin.js
├── static
│   └── favicon.ico
├── store
│   └── state.js
```

## NuxtJS

- A pasta **assets** contém arquivos estáticos que serão servidos **com pré-processamento** tais como arquivos transpilados ou imagens a serem transformadas
- A pasta **static** contém arquivos estáticos que serão servidos **sem pré-processamento** tais como CSS/JS de terceiros ou ícones

# NuxtJS: The Intuitive Vue Framework

```
.
├── nuxt.config.js
├── package.json
├── package-lock.json
├── jsconfig.json
├── README.md
├── assets
│   └── style.scss
├── components
│   └── Logo.vue
├── layouts
│   └── default.vue
├── middleware
│   └── middleware.js
├── pages
│   └── index.vue
├── plugins
│   └── plugin.js
├── static
│   └── favicon.ico
├── store
│   └── state.js
```

## NuxtJS

- A pasta **components** contém single file components que representam **blocos de construção da aplicação**
- A pasta **pages** contém single file components que representam **páginas da aplicação**
- A pasta **layouts** contém single file components que representam **estruturas gerais da aplicação**

# NuxtJS: The Intuitive Vue Framework

```
.
├── nuxt.config.js
├── package.json
├── package-lock.json
├── jsconfig.json
├── README.md
├── assets
│   └── style.scss
├── components
│   └── Logo.vue
├── layouts
│   └── default.vue
├── middleware
│   └── middleware.js
├── pages
│   └── index.vue
├── plugins
│   └── plugin.js
├── static
│   └── favicon.ico
├── store
│   └── state.js
```

## NuxtJS

- A pasta `middleware` contém funções que podem ser executadas **antes de renderizar páginas** tais como autenticação / autorização
- A pasta `plugins` contém código JavaScript que deve ser executado **antes de inicializar o Vue** tais como setup de uma biblioteca para acesso ao servidor

# NuxtJS: The Intuitive Vue Framework

```
.
├── nuxt.config.js
├── package.json
├── package-lock.json
├── jsconfig.json
├── README.md
├── assets
│   └── style.scss
├── components
│   └── Logo.vue
├── layouts
│   └── default.vue
├── middleware
│   └── middleware.js
├── pages
│   └── index.vue
├── plugins
│   └── plugin.js
├── static
│   └── favicon.ico
└── store
    └── state.js
```

## NuxtJS

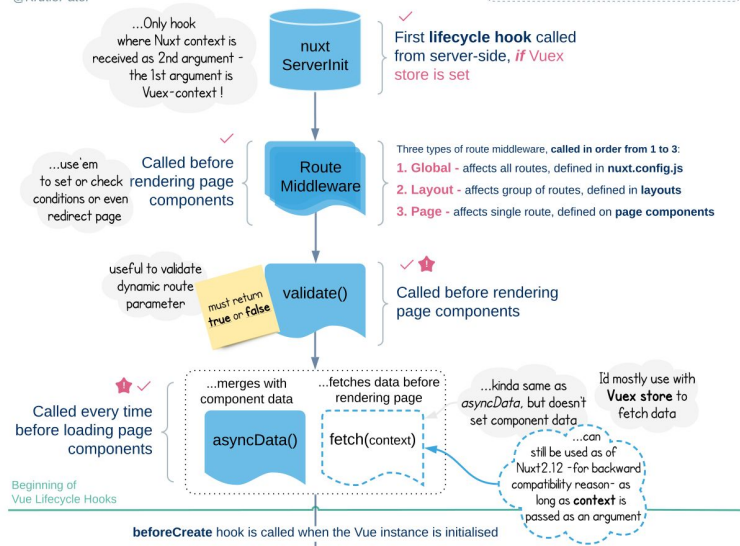
- A pasta `store` pode conter a `Vuex` global `state.js` ou vários módulos `name.js` para comunicação transversal entre componentes
- No NuxtJS, as rotas do `Vue Router` são `inferidas` a partir dos diretórios da pasta `pages` sem necessidade de configuração manual

# NuxtJS: The Intuitive Vue Framework

## NUXT.JS LIFECYCLE HOOKS

Nuxt >= 2.12 - April 2020

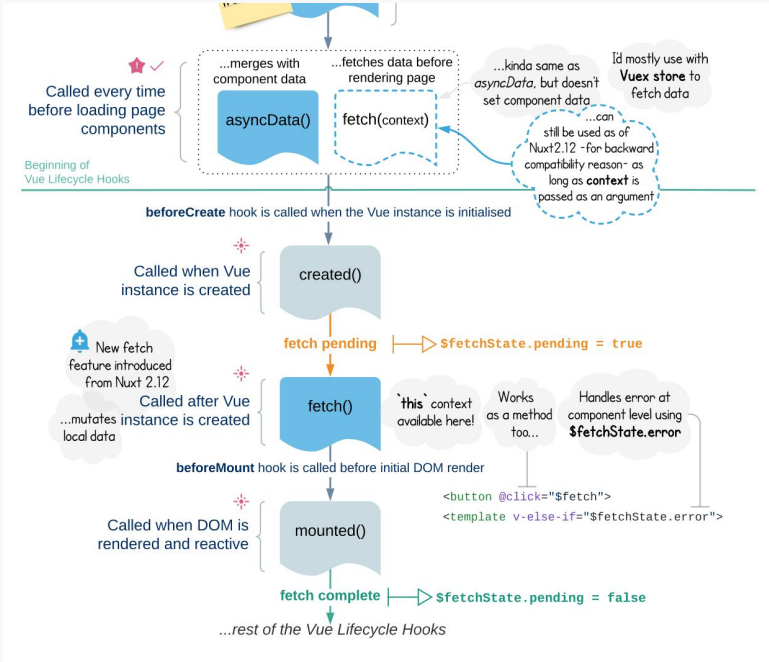
@KrutiePatel



## NuxtJS

- No Nuxt, componentes do Vue possuem etapas adicionais no seu **ciclo de vida** administrado pelo framework
- Algumas etapas como **nuxtServerInit** e **asyncData** são úteis quando utilizamos o Nuxt para Server-Side Rendering
- Outras etapas como **fetch** são úteis para ambos os modos de operação para fazer chamadas do cliente para o servidor

# NuxtJS: The Intuitive Vue Framework



# NuxtJS

- No Nuxt, componentes do Vue possuem etapas adicionais no seu **ciclo de vida** administrado pelo framework
- Algumas etapas como **nuxtServerInit** e **asyncData** são úteis quando utilizamos o Nuxt para Server-Side Rendering
- Outras etapas como **fetch** são úteis para ambos os modos de operação para fazer chamadas do cliente para o servidor

# 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/](https://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: **zsyt5**



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

