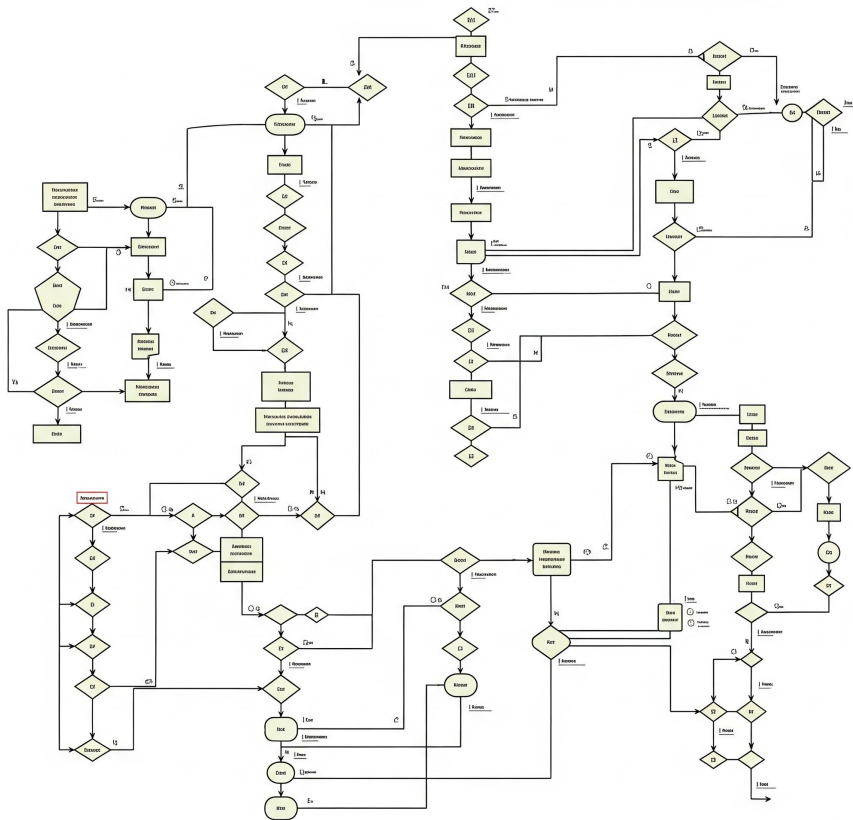


Next.js: Renderização no Servidor e Arquitetura do Roteamento

Felipe Colpo Bagesteiro (fcbagesteiro@inf.ufsm.br)

O problema da Escalabilidade

- A utilização de bibliotecas no frontend acaba por trazer alguns problemas de escalabilidade
- A falta de estruturação envolvendo múltiplos componentes pode atrasar tanto a aplicação, quanto o desenvolvimento



Biblioteca X Framework

- Bibliotecas: Coleções de código JavaScript, CSS e, às vezes, HTML, que são projetadas para serem reutilizadas em vários projetos.
 - React, JQuery
- Framework: fornece ferramentas e componentes prontos para uso, acelerando o desenvolvimento de interfaces de usuário para aplicações web
 - Vue, Angular, Svelte, Next

Aplicação de Frameworks no Frontend

- Um framework nos dá uma estrutura, uma forma padronizada de construir.
- “O que define a escolha de um framework não é sua performance, mas sim quem vai utilizar”

- Tipos

- Frameworks Opinativos (Convention over Configuration)
 - Ruby on rails
 - Django
- Frameworks Progressivos
 - Vue

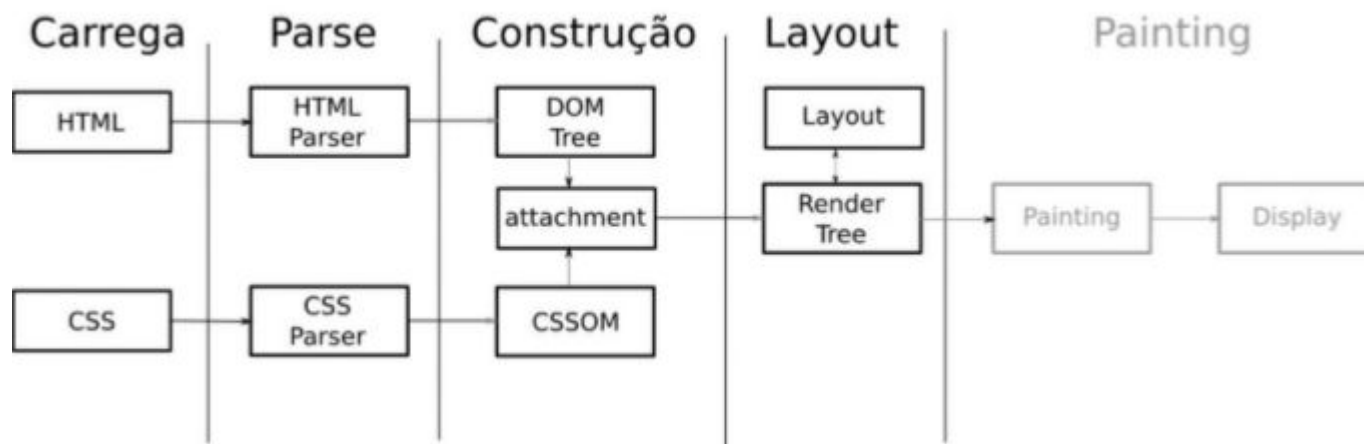
Next.js



- Framework React opinativo
- Orientado a React
- Estruturado pela Vercel
- Sistema de roteamento inteligente
- Otimizações automáticas
- Compilador escrito em Rust
- Permite diferentes formatos de renderização das suas páginas

Renderização

- Processo de transferência dos dados da página web do servidor para o cliente
- Conceito é dividido em 3 técnicas de renderização:
 - Static Site Generation (SSG)
 - Server-Side Rendering (SSR)
 - Incremental Static Regeneration (ISR)



Static Site Generation (SSG)

- Geração Estática
- O HTML de cada página da sua aplicação durante o processo de compilação(Build)
- Quem renderiza?
 - O servidor, mas apenas uma vez.
- Quando?
 - No momento do build, antes que qualquer usuário acesse o site.
- O que o usuário recebe?
 - Um arquivo HTML completo e pré-renderizado, que carrega de forma extremamente rápida no navegador.

Server-Side Rendering(SSR)

- Renderização no Lado do Servidor
- HTML de uma página é gerado no servidor a cada requisição
- Quem renderiza?
 - O servidor.
- Quando?
 - Em tempo real, toda vez que um usuário solicita a página..
- O que o usuário recebe?
 - Um HTML completo e atualizado, gerado especificamente para aquela requisição.

Incremental Static Regeneration(ISR)

- Regeneração Estática Incremental
- Permite que você atualize páginas estáticas (geradas com SSG) após o build ter sido concluído.
- Gera a página estaticamente uma vez e define um intervalo de tempo para revalidação
- Quem renderiza?
 - O servidor.
- Quando?
 - No build (primeira vez) e depois em segundo plano, em intervalos de tempo, quando chegam novas requisições
- O que o usuário recebe?
 - Um HTML estático e instantâneo. O primeiro usuário após o intervalo de tempo ainda recebe a versão antiga (stale), mas sua visita dispara a regeneração. O próximo usuário já recebe a página atualizada."

Criação de um projeto com Next.js

- Necessária a instalação de um gerenciador de pacotes node, como npm ou yarn.
- `npx create-next-app@latest`

```
What is your project named?  my-app
Would you like to use TypeScript?  No / Yes
Which linter would you like to use?  ESLint / Biome / None
Would you like to use Tailwind CSS?  No / Yes
Would you like your code inside a `src/` directory?  No / Yes
Would you like to use App Router? (recommended)  No / Yes
Would you like to use Turbopack? (recommended)  No / Yes
Would you like to customize the import alias (`@/*` by default)?  No / Yes
```

Fundamentos do Roteamento

- "Pastas são rotas, arquivos page.js são a UI"
- Divido comumente em:
 - Rotas Estáticas
 - Rotas Aninhadas
 - Rotas Dinâmicas
- Definição de Arquivos Especiais
 - page
 - layout
 - loading