Introdução React

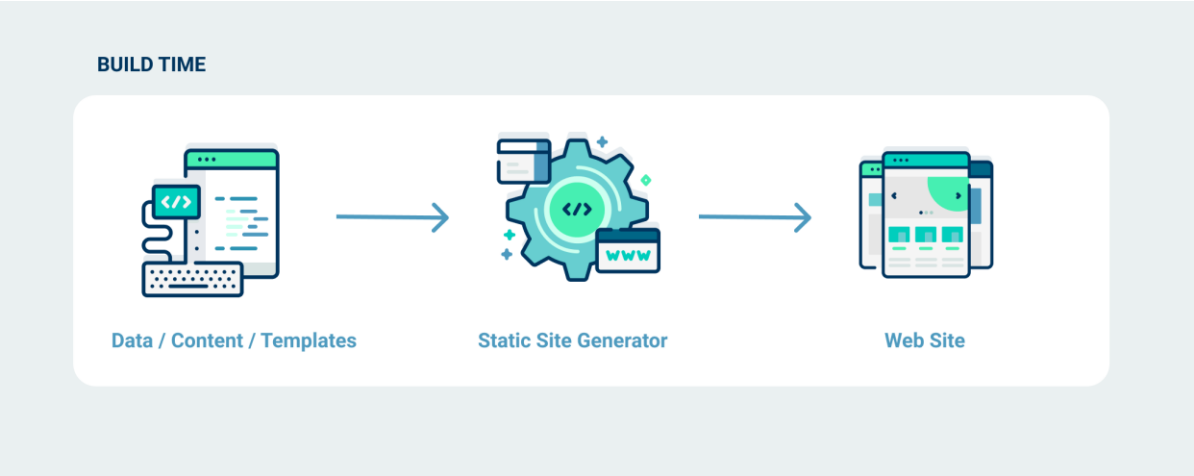
Fonte: Web e Documentação oficial React e Next JS

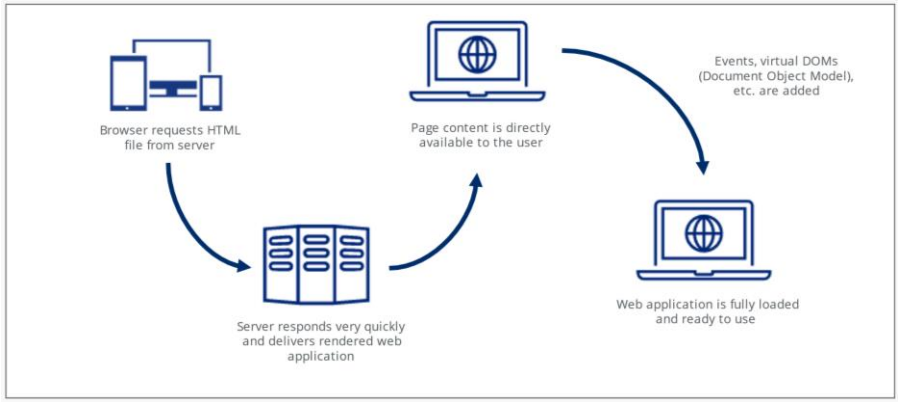
Abordagens para a criação de aplicativos web

- SSG (Static Site Generation)
- SSR (Server-side rendering)
- CSR (Client-side rendering)
- PWA (Progressive Web App)
- API (Application Programming Interface)
- Headless CMS
- Jamstack
- Low-code/no-code
- SPA (Single-Page Application)

SSG (Static Site Generation)

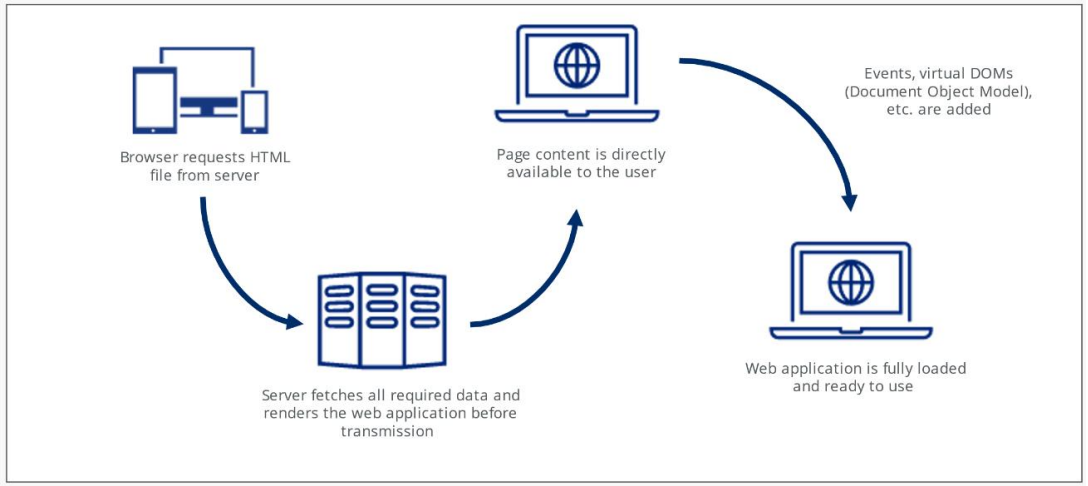
A geração de sites estáticos (SSG) é uma técnica em que o conteúdo HTML da página é gerado em tempo de compilação, em vez de ser gerado dinamicamente no servidor ou no cliente. O SSG é útil para sites com conteúdo estático ou que mudam pouco, pois permite que as páginas sejam carregadas mais rapidamente, sem precisar de um servidor para gerar o HTML.





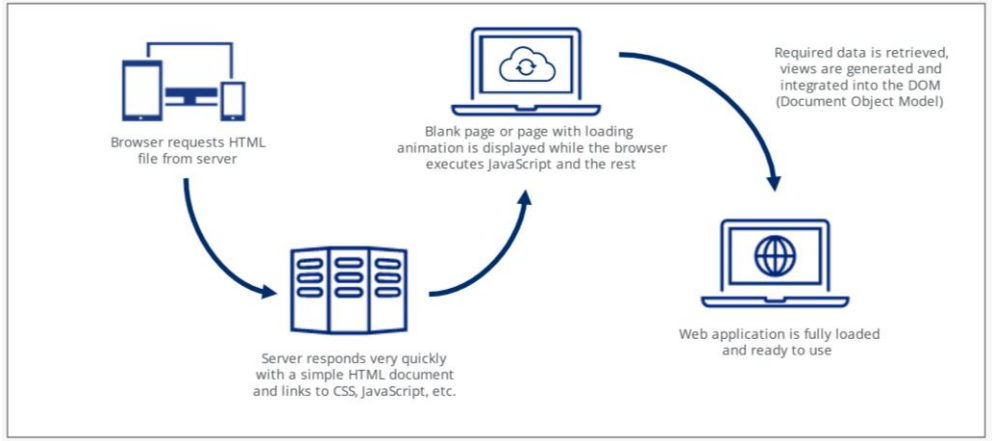
SSR (Server-side rendering)

A renderização do lado do servidor (SSR) é uma técnica em que o servidor gera HTML para a página da web com base nas solicitações do usuário. Com o SSR, o servidor pode enviar o HTML para o navegador mais rapidamente, permitindo uma experiência de usuário mais rápida. Além disso, o SSR pode melhorar a indexação do mecanismo de pesquisa, pois o conteúdo é renderizado no servidor e enviado ao navegador como HTML.



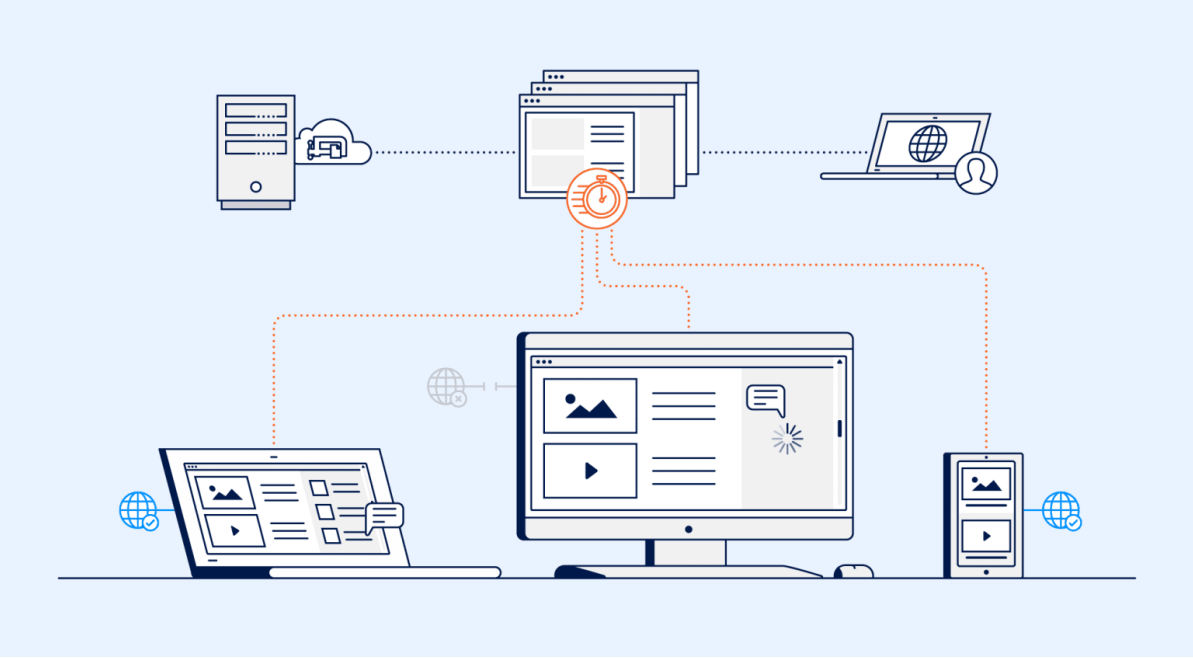
CSR (Client-side rendering)

Nessa abordagem, o cliente (navegador do usuário) é responsável por renderizar as páginas. O servidor envia apenas dados brutos para o cliente, e o JavaScript é usado para manipular e exibir esses dados. Isso torna a experiência do usuário mais rápida e interativa, pois as páginas são atualizadas em tempo real sem a necessidade de recarregar a página inteira. Essa abordagem é utilizada principalmente em aplicativos web complexos, como redes sociais, ferramentas de produtividade e jogos.



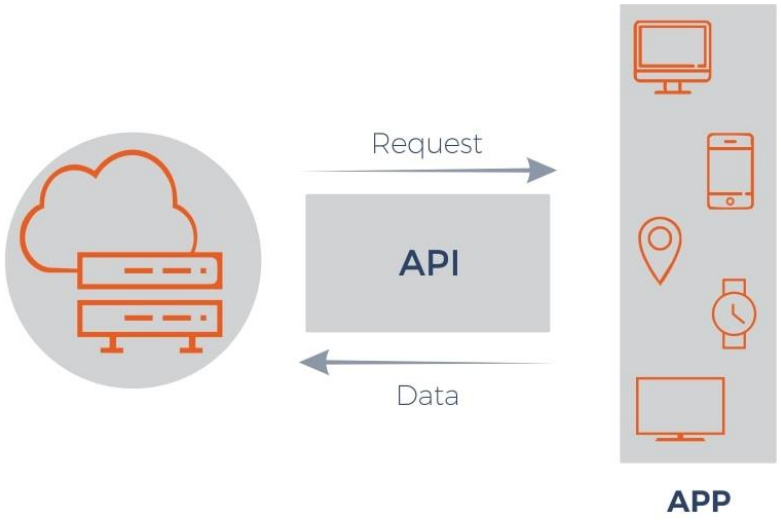
PWA (Progressive Web App)

Representa um aplicativo da web que pode ser instalado no dispositivo do usuário e usado como um aplicativo nativo. As vantagens do PWA incluem uma experiência de usuário mais fluida, a capacidade de funcionar offline e a capacidade de receber notificações push.



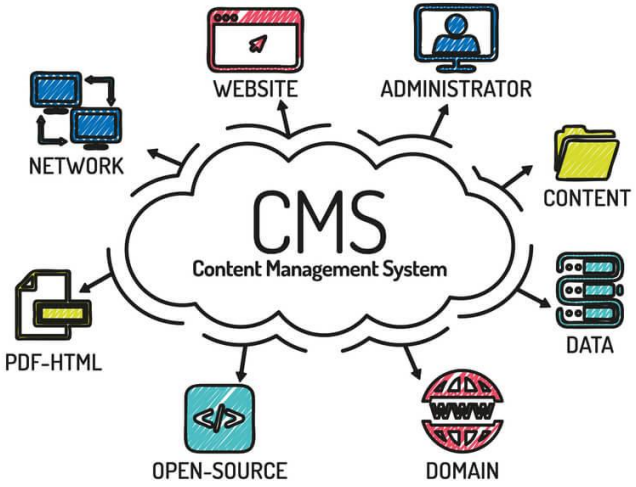
API (Application Programming Interface)

Essa abordagem envolve a criação de uma API (Interface de Programação de Aplicativos) para permitir que diferentes aplicativos e plataformas se comuniquem entre si. Os desenvolvedores criam endpoints de API que permitem que as informações sejam trocadas entre diferentes aplicativos ou serviços. Isso pode incluir desde APIs de mídias sociais que permitem que os usuários postem conteúdo em várias plataformas, até APIs de comércio eletrônico que permitem que diferentes lojas virtuais compartilhem informações sobre produtos e transações.



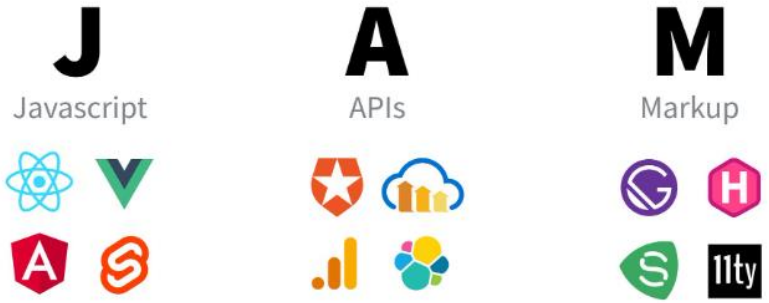
Headless CMS

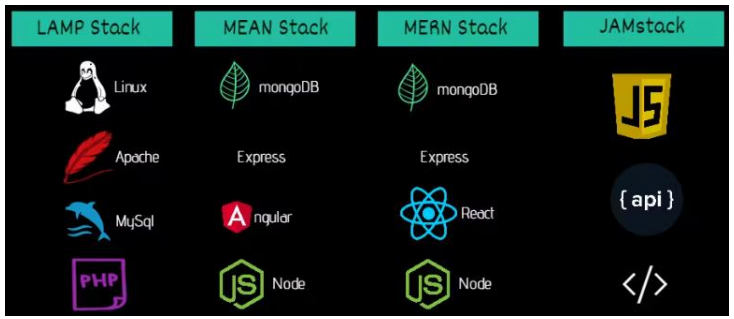
Essa abordagem envolve o uso de um sistema de gerenciamento de conteúdo (CMS) que não tem uma interface de usuário integrada. Em vez disso, o CMS é usado como uma plataforma para armazenar, gerenciar e entregar conteúdo para diferentes aplicativos e dispositivos. Isso pode incluir desde sites e aplicativos web, até dispositivos de IoT (Internet das Coisas) e aplicações de realidade aumentada.



Jamstack

Essa é uma abordagem de desenvolvimento que se concentra em JavaScript, APIs e pré-renderização para criar sites e aplicativos altamente performáticos e seguros. O Jamstack (JavaScript, API, Markup) envolve o uso de pré-renderização para entregar conteúdo estático aos usuários e a API para fornecer conteúdo dinâmico, criando uma experiência de usuário rápida e suave. Além disso, o Jamstack usa tecnologias modernas de segurança, como o CDN (Content Delivery Network) para proteger os sites contra ataques de segurança.



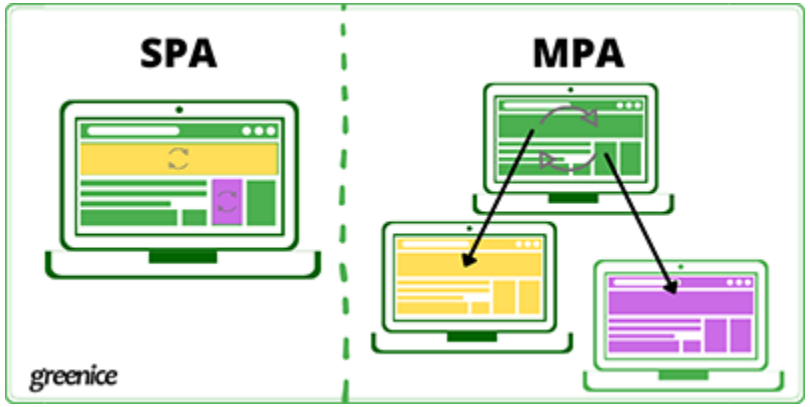


Low-code/no-code

Essa é uma abordagem de desenvolvimento que envolve o uso de plataformas que permitem que usuários com pouca ou nenhuma experiência em programação criem sites e aplicativos. As plataformas de low-code e no-code fornecem ferramentas visuais, como interfaces de arrastar e soltar e modelos pré-fabricados, que permitem que os usuários criem aplicativos com facilidade e rapidez.

SPA (Single-Page Application)

É um tipo de aplicativo web que carrega uma única página HTML e atualiza dinamicamente o conteúdo da página à medida que o usuário interage com o aplicativo, sem precisar carregar uma nova página do servidor. As vantagens do SPA incluem uma experiência de usuário mais rápida e fluida, pois a página não precisa ser carregada novamente a cada interação do usuário, e a capacidade de criar aplicativos altamente interativos e personalizados.



O conceito de Single-Page Application (SPA) não é novo, mas o termo em si foi cunhado em 2005 por Steve Yen, co-fundador da empresa de software OnCenter, durante a criação de um aplicativo web complexo. Desde então, o conceito tem sido amplamente adotado na comunidade de desenvolvimento web, especialmente após a criação do framework React em 2011 pelo Facebook. O React é frequentemente usado para criar aplicativos SPA devido à sua capacidade de atualizar rapidamente o conteúdo da página sem a necessidade de recarregá-la, o que torna a experiência do usuário mais rápida e fluida. Outros frameworks, como Angular e Vue.js, também são usados para criar aplicativos SPA.

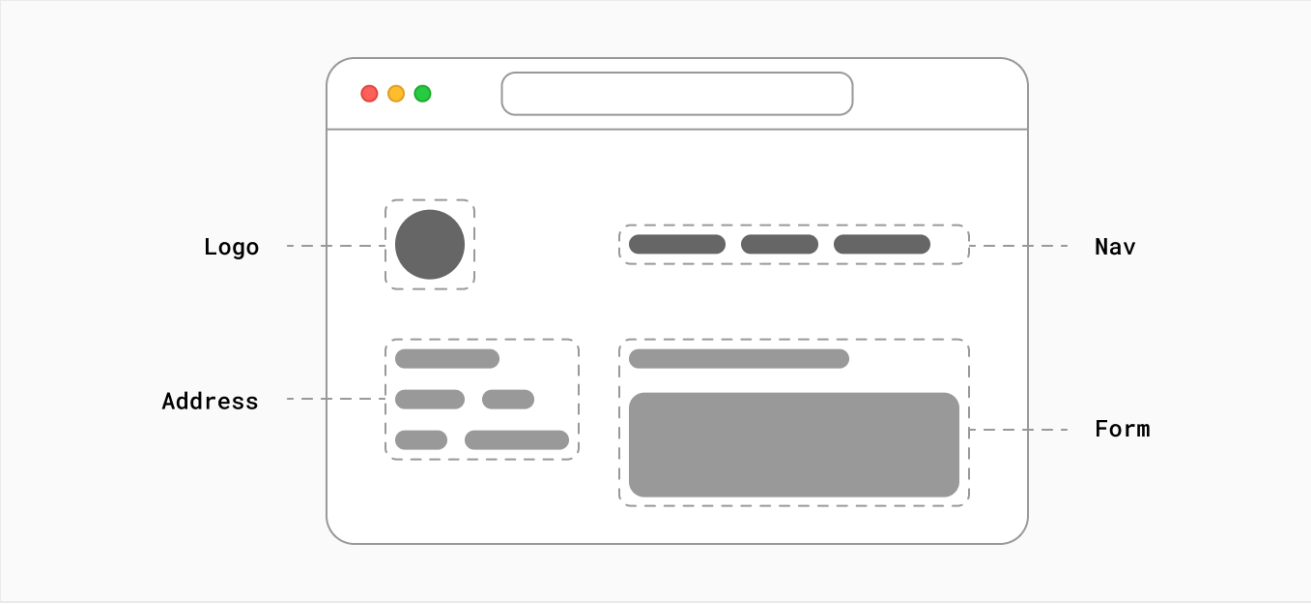
O React é uma biblioteca JavaScript altamente modular, o que significa que há muitas bibliotecas e frameworks complementares que podem ser usados junto com o React para diferentes abordagens de desenvolvimento web. A seguir são apresentadas algumas bibliotecas e frameworks de React que são comumente usados para diferentes abordagens:

- Para SPA (Single-Page Applications): React Router, Redux, MobX, Relay, Apollo Client.
- Para CSR (Client-Side Rendering): Create React App, Next.js, Remix, Gatsby, React Static, Vite.
- Para SSR (Server-Side Rendering): Next.js, Remix, After.js, Nuxt.js, Vite.
- Para SSG (Static Site Generation): Gatsby, Next.js, Remix , Jekyll, Hugo.

O que é React?

React é uma biblioteca JavaScript para construir interfaces de usuário interativas .

Por interfaces de usuário, queremos dizer os elementos que os usuários veem e interagem na tela.



Por biblioteca, queremos dizer que o React fornece funções úteis para construir a interface do usuário, mas deixa para o desenvolvedor onde usar essas funções em seu aplicativo.

Parte do sucesso do React é que ele é relativamente sem opinião sobre os outros aspectos da construção de aplicativos. Isso resultou em um ecossistema próspero de ferramentas e soluções de terceiros.

Isso também significa, no entanto, que construir um aplicativo React completo desde o início requer algum esforço. Os desenvolvedores precisam gastar tempo configurando ferramentas e reinventando soluções para requisitos comuns de aplicativos.

Antes do React, haviam outras tecnologias e técnicas usadas para atualizar o conteúdo das páginas web sem a necessidade de recarregar a página inteira. Algumas dessas técnicas incluíam o uso de JavaScript puro ou bibliotecas como jQuery, que permitiam atualizar o conteúdo de uma página sem a necessidade de carregá-la novamente. No entanto, essas técnicas tinham suas limitações em termos de complexidade, desempenho e escalabilidade.

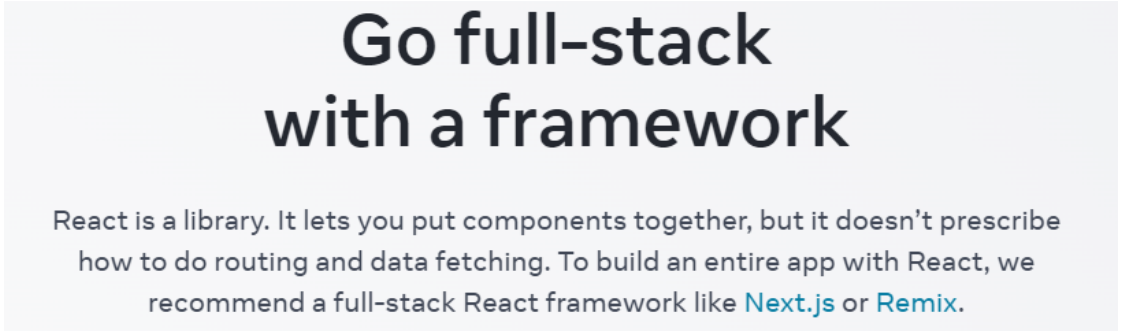
Outra abordagem usada anteriormente era o AJAX (Asynchronous JavaScript and XML), que permitia carregar e enviar dados para um servidor em segundo plano, sem a necessidade de recarregar a página. O AJAX permitiu o desenvolvimento de aplicativos web mais dinâmicos e interativos, mas ainda exigia muito código JavaScript para gerenciar a atualização de dados na página.

O React, no entanto, introduziu uma nova abordagem para atualização de conteúdo em tempo real na página, utilizando a técnica de renderização de componentes. Com o React, é possível criar componentes independentes que podem ser atualizados de forma isolada sem afetar o resto da página, tornando a atualização do conteúdo mais eficiente e escalável.

Além disso, o ReactJS oferece uma ampla gama de ferramentas e bibliotecas que ajudam no desenvolvimento de SPAs, incluindo o React Router, que permite que os desenvolvedores gerenciem as rotas do aplicativo, e o Redux, que ajuda a gerenciar o estado do aplicativo.

O ReactJS também pode ser usado para criar páginas que não sejam Single-Page Applications (SPAs). Embora o ReactJS tenha sido projetado para o desenvolvimento de SPAs, ele também pode ser usado para criar páginas web mais tradicionais.

Por exemplo, você pode usar o ReactJS para criar componentes específicos em uma página da web e incorporá-los em uma página existente. Isso pode ser útil se você quiser aproveitar as vantagens do ReactJS, como a facilidade de reutilização de componentes, mas não deseja criar um aplicativo SPA completo.



O que é Remix?

Remix é um framework de desenvolvimento web para criar aplicativos modernos e escaláveis usando JavaScript e React. É baseado em abordagens como SSR (Server Side Rendering), SSG (Static Site Generation) e CSR (Client Side Rendering), e permite que os desenvolvedores criem aplicativos que carregam rapidamente, fornecem uma experiência de usuário responsiva e são facilmente escaláveis.

O Remix é altamente recomendado para projetos que exigem uma abordagem híbrida de renderização, em que a renderização ocorre tanto no servidor quanto no lado do cliente, a fim de fornecer os benefícios de ambas as abordagens. Além disso, o Remix oferece uma experiência de desenvolvimento intuitiva, testável e focada em componentes, que ajuda a acelerar o processo de desenvolvimento.

A abordagem recomendada para o desenvolvimento com Remix é seguir as práticas recomendadas de desenvolvimento web, como dividir o aplicativo em componentes reutilizáveis, manter o estado do aplicativo gerenciado centralmente, usar gerenciadores de estado e manter o código limpo e organizado.

O que é Next.js?

Next.js é uma estrutura React que fornece blocos de construção para criar aplicativos da web.

Por estrutura, queremos dizer que o Next.js lida com as ferramentas e a configuração necessárias para o React e fornece estrutura, recursos e otimizações adicionais para seu aplicativo.

Blocos de construção de um aplicativo da Web

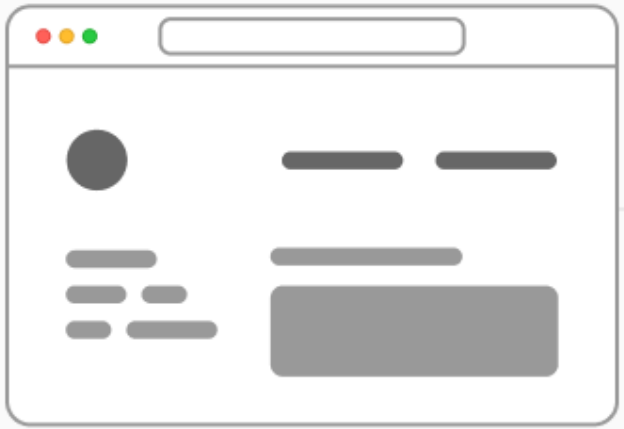
Há algumas coisas que você precisa considerar ao criar aplicativos modernos. Como:

- Interface do usuário - como os usuários irão consumir e interagir com seu aplicativo.
- Roteamento - como os usuários navegam entre diferentes partes do seu aplicativo.
- Busca de dados - onde seus dados residem e como obtê-los.
- Renderização - quando e onde você renderiza conteúdo estático ou dinâmico.
- Integrações - quais serviços de terceiros você usa (CMS, autenticação, pagamentos etc.) e como você se conecta a eles.
- Infraestrutura - onde você implanta, armazena e executa o código do seu aplicativo (Serverless, CDN, Edge, etc).
- Desempenho - como otimizar seu aplicativo para usuários finais.
- Escalabilidade - como seu aplicativo se adapta à medida que sua equipe, dados e tráfego crescem.
- Experiência do desenvolvedor - a experiência de sua equipe na criação e manutenção de seu aplicativo.

Para cada parte de seu aplicativo, você precisará decidir se criará uma solução sozinho ou usará outras ferramentas, como bibliotecas e estruturas.

Client

Server



Next.js

UI with React

CSR

SSR/SSG

Routing

Data Fetching

...

Favorite tools

Database

CMS

Ecommerce

Auth

...

Você pode usar o React para criar sua interface do usuário e adotar gradualmente os recursos do Next.js para resolver os requisitos comuns do aplicativo, como roteamento, busca de dados, integrações - tudo isso enquanto melhora a experiência do desenvolvedor e do usuário final.

Seja você um desenvolvedor individual ou parte de uma equipe maior, você pode aproveitar o React e o Next.js para criar aplicativos da Web totalmente interativos, altamente dinâmicos e de alto desempenho.

Interfaces de usuário de renderização

Para entender como o React funciona, primeiro precisamos de um entendimento básico de como os navegadores interpretam seu código para criar interfaces de usuário (UI) interativas.

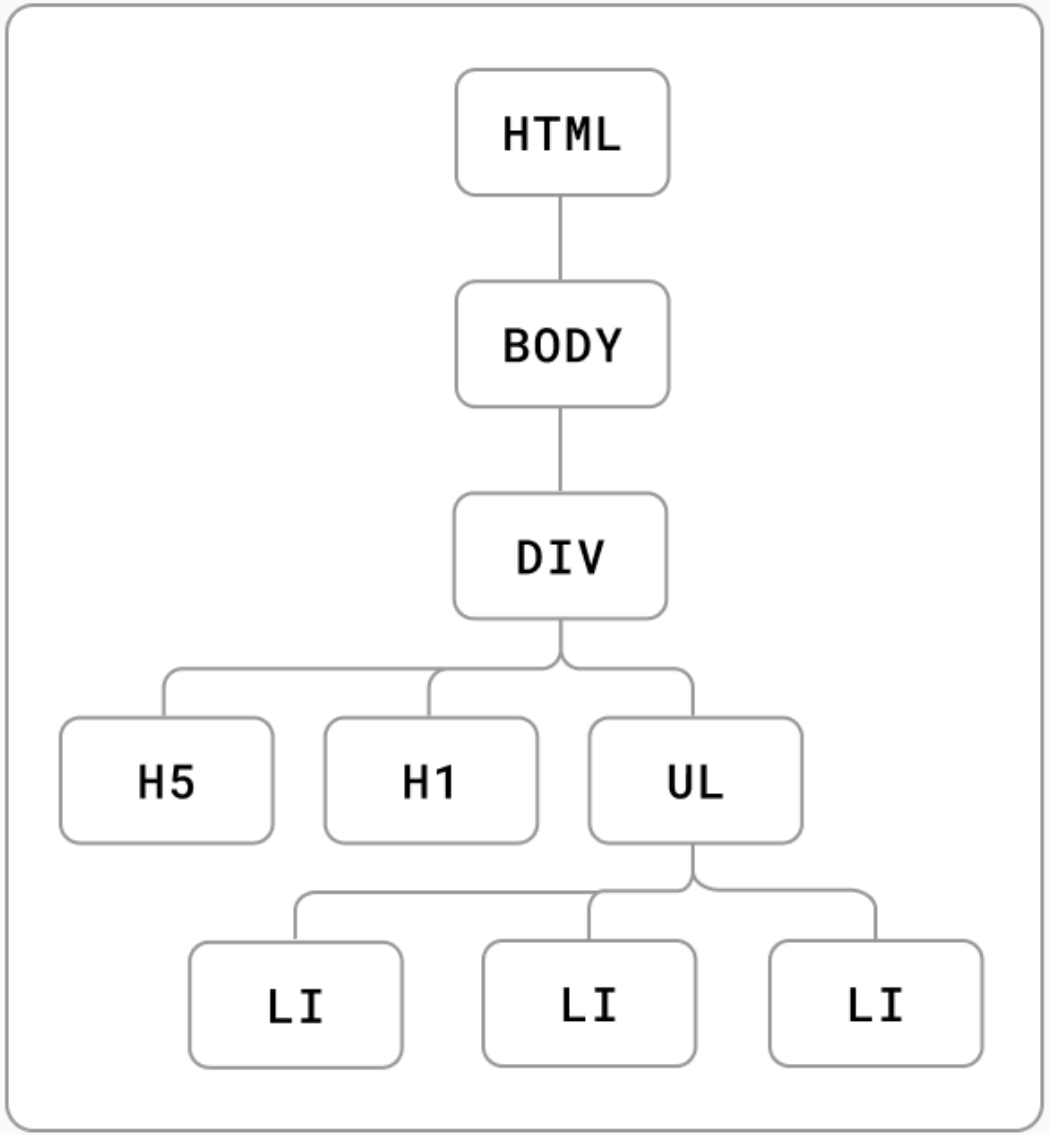
Quando um usuário visita uma página da web, o servidor retorna um arquivo HTML para o navegador que pode ter a seguinte aparência:

HTML

index.html

```
1  <html>
2    <body>
3      <div>
4        <h5>Meet the</h5>
5        <h1>Engineers</h1>
6        <ul>
7          <li>A. Lovelace</li>
8          <li>G. Hopper</li>
9          <li>M. Hamilton</li>
10       </ul>
11     </div>
12   </body>
14 </html>
15
16
```

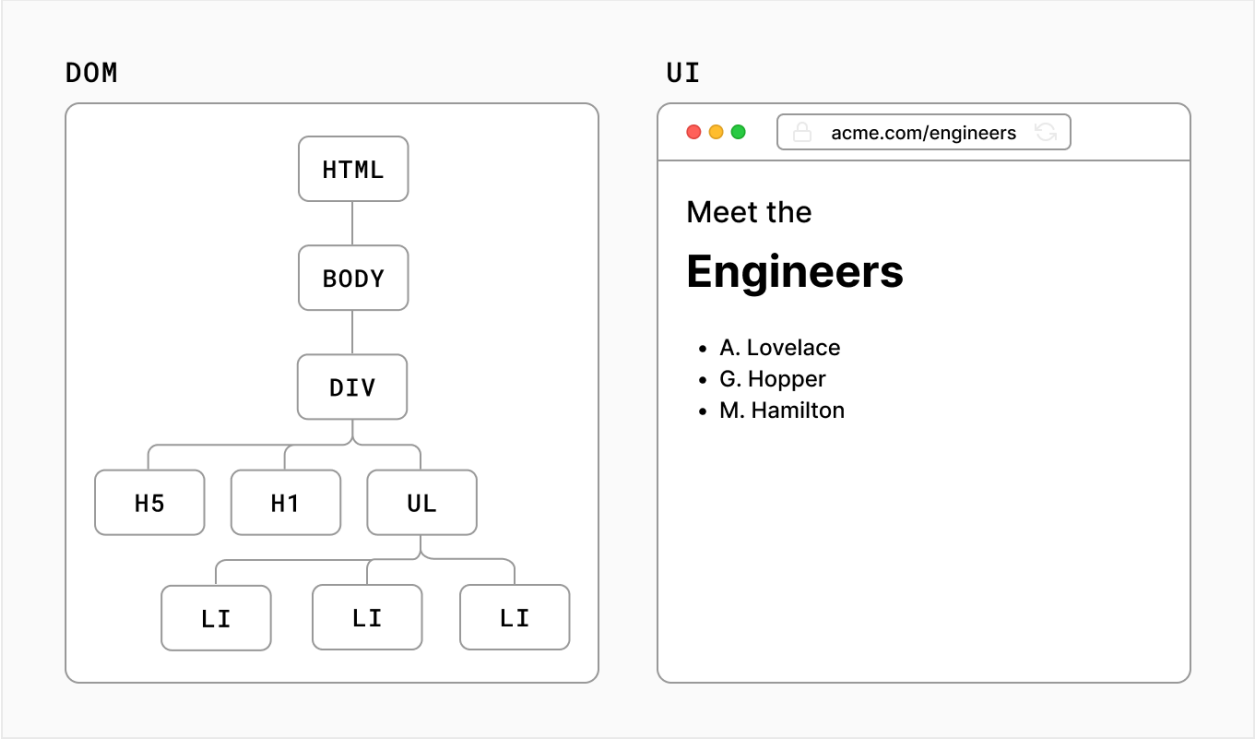
DOM



O navegador então lê o HTML e constrói o Document Object Model (DOM).

O que é o DOM?

O DOM é uma representação de objeto dos elementos HTML. Ele atua como uma ponte entre seu código e a interface do usuário e possui uma estrutura semelhante a uma árvore com relacionamentos pai e filho.



Você pode usar métodos DOM e uma linguagem de programação, como JavaScript, para ouvir eventos do usuário e manipular o DOM selecionando, adicionando, atualizando e excluindo elementos específicos na interface do usuário. A manipulação de DOM permite não apenas direcionar elementos específicos, mas também alterar seu estilo e conteúdo.

O que é SPA?

a) Uma nova tecnologia para desenvolvimento de bancos de dados

b) Uma técnica de CEO para melhorar a performance de aplicações web

c) Um padrão arquitetural para desenvolvimento de aplicações web

d) Uma abordagem de desenvolvimento web que consiste em criar páginas web altamente interativas, que não precisam ser recarregadas durante o uso.

O que é SPA?

a) Uma nova tecnologia para desenvolvimento de bancos de dados

b) Uma técnica de CEO para melhorar a performance de aplicações web

c) Um padrão arquitetural para desenvolvimento de aplicações web

d) Uma abordagem de desenvolvimento web que consiste em criar páginas web altamente interativas, que não precisam ser recarregadas durante o uso.

Uma abordagem de desenvolvimento web que consiste em criar páginas web altamente interativas, que não precisam ser recarregadas durante o uso. Como discutido neste chat, SPA (Single Page Application) é uma abordagem de desenvolvimento web que consiste em criar aplicações web altamente interativas, que atualizam o conteúdo dinamicamente sem precisar recarregar a página.

O que é React?

a) Uma linguagem de programação

b) Um framework de desenvolvimento web

c) Uma biblioteca de JavaScript para criação de interfaces de usuário

d) Um banco de dados

O que é React?

a) Uma linguagem de programação

b) Um framework de desenvolvimento web

c) Uma biblioteca de JavaScript para criação de interfaces de usuário

d) Um banco de dados

O React é uma biblioteca de JavaScript utilizada para criar interfaces de usuário em aplicações web.

Verdadeiro ou falso : você pode atualizar o conteúdo da página manipulando o DOM.

Verdadeiro

Falso

Verdadeiro ou falso : você pode atualizar o conteúdo da página manipulando o DOM.

Verdadeiro

Falso

Aprenda mais em:

Introdução ao DOM

Comece a visualizar e alterar o DOM

Destaque e inspecione os nós DOM

Referências

- Documentação oficial do React: https://reactjs.org/docs/getting-started.html
- Site do Next.js (SSR/SSG para React): https://nextjs.org/
- Artigo da Smashing Magazine sobre React e SSR: https://www.smashingmagazine.com/2020/07/react-server-side-rendering/
- Artigo da Google Developers sobre PWAs: https://developers.google.com/web/progressive-web-apps
- Livro "Pro React 16" de Adam Freeman
- Livro "Learning React" de Alex Banks e Eve Porcello

Extra – Além de React

React, Vue e Angular são três dos principais frameworks JavaScript usados para o desenvolvimento de aplicações web. Embora tenham semelhanças, também existem diferenças significativas entre eles, como:

1. Arquitetura: O React é uma biblioteca, enquanto o Angular e o Vue são frameworks. Isso significa que o React oferece menos recursos nativos do que os outros dois, mas pode ser usado com outras bibliotecas e frameworks. O Angular é uma estrutura completa e inclui muitos recursos nativos, enquanto o Vue está em algum lugar no meio, oferecendo recursos nativos suficientes para facilitar o desenvolvimento.

2. Linguagem: O React é baseado em JavaScript, enquanto o Angular usa TypeScript, que é uma extensão do JavaScript. O Vue é escrito em JavaScript, mas também suporta TypeScript.
3. Renderização: O React é usado principalmente para renderização do lado do cliente (CSR), enquanto o Angular e o Vue suportam tanto a renderização do lado do servidor (SSR) quanto a renderização do lado do cliente.
4. Curva de aprendizado: O React tem uma curva de aprendizado mais íngreme do que o Vue e o Angular, pois requer conhecimento em bibliotecas adicionais, como Redux. O Angular tem uma curva de aprendizado íngreme, pois é uma estrutura completa que requer conhecimento em TypeScript e muitos outros recursos nativos. O Vue tem uma curva de aprendizado mais suave do que os outros dois, pois é uma estrutura intermediária entre uma biblioteca e um framework completo.

5. Comunidade e ecossistema: O React e o Angular têm grandes comunidades e ecossistemas, com muitas bibliotecas e ferramentas disponíveis. O Vue tem uma comunidade menor, mas em crescimento, com muitas ferramentas e bibliotecas disponíveis.

Em resumo, a escolha do React, Vue ou Angular dependerá das necessidades específicas do projeto, do conhecimento prévio do desenvolvedor e das preferências pessoais. Todos os três são poderosos e podem ser usados para criar aplicativos web escaláveis, mas cada um tem suas próprias características únicas que os tornam adequados para diferentes casos de uso.

A escolha entre React, Vue ou Angular dependerá do tipo de projeto de desenvolvimento e das necessidades específicas do projeto.

- React: Aplicativos web com interface do usuário complexa e interativa. Projetos que exigem muitas atualizações em tempo real. Projetos que se concentram em componentes reutilizáveis.

- Vue: Pequenos projetos e prototipagem rápida. Aplicativos web com interface do usuário complexa e interativa. Projetos em que a velocidade e a facilidade de desenvolvimento são importantes.

- Angular: Grandes projetos empresariais. Aplicativos web com interface do usuário complexa e interativa. Projetos que exigem muitos recursos nativos, como autenticação de usuário e roteamento.

No entanto, é importante lembrar que essas são apenas diretrizes gerais e que cada projeto é único e pode ter necessidades específicas diferentes. Portanto, a escolha entre React, Vue ou Angular deve ser baseada em uma avaliação cuidadosa das necessidades do projeto e das habilidades e preferências da equipe de desenvolvimento.