



Introdução ao React

[O que é React?](#)

[Escreva componentes com código e marcação](#)

[Interatividade](#)

[Uso de frameworks](#)

[Multiplataformas](#)

[Atualizações](#)

[Vantagens em se utilizar React](#)

[Desvantagens do React](#)

[Evolução do React](#)

[Referências](#)

O que é React?

- Biblioteca para construções de interfaces (web, mobile, tv, VR);
 - Construído pelo Facebook/Grupo Meta.
 - Quando acoplada com outras bibliotecas, consegue rodar em ambientes variados;
 - Permite que os desenvolvedores construam aplicativos web e móveis de maneira eficiente;
 - Reatividade e a capacidade de atualização automática da interface quando os dados mudam;
 - React permite que você construa interfaces de usuário a partir de peças individuais chamadas componentes.
-
- Não importa se você está trabalhando sozinho ou em equipe, a experiência de usar o React é consistente para todos.
 - O React permite que você integre facilmente componentes de diferentes fontes, sejam eles escritos por você mesmo, por outras pessoas ou por equipes

diferentes;

- É fácil reutilizar e combinar componentes no React;
- O processo de juntar componentes no React é como montar um quebra-cabeça;
 - Cada peça pode ser feita por alguém diferente;
 - Todas se encaixam perfeitamente para formar uma imagem completa.

Escreva componentes com código e marcação

- Os componentes do React são funções em JavaScript;
- "Aprender React é aprender a programar";
- Mistura código de marcação com JavaScript;
 - Sintaxe de marcação é chamada JSX (JavaScript XML);
 - É uma extensão de sintaxe do JavaScript popularizada pelo React;
 - Colocar a marcação JSX próxima à lógica de renderização relacionada torna os componentes do React fáceis de criar, manter e excluir.

Interatividade

- Os componentes do React recebem dados e retornam o que deve aparecer na tela;
- Você pode passar novos dados para eles em resposta a uma interação.
 - Usuário digita em um campo de entrada;
 - O React então atualizará a tela para corresponder aos novos dados.
- Integrar React em uma página HTML existente permite renderizar componentes interativos React em qualquer parte da página, sem a necessidade de reconstruir toda a página em React.

Uso de frameworks

- O React é uma biblioteca que permite a montagem de componentes, mas não prescreve como fazer roteamento e busca de dados;
- Pode-se utilizar para uma solução mais completa frameworks como:
 - [Next.js](#)

- Remix

Multiplataformas

- O React permite que você construa tanto aplicativos web quanto aplicativos nativos usando as mesmas habilidades;
 - **React.js**
 - **React Native**
 - Versão do React focada em desenvolver aplicativos móveis nativos;
 - Dispositivos iOS e Android;
 - Permite aos desenvolvedores criar aplicativos com interface de usuário semelhante à dos aplicativos nativos, enquanto utiliza a mesma base de código JavaScript da web.
 - **React VR**
 - Criação de experiências imersivas em realidade virtual (VR) usando as mesmas ferramentas e habilidades do React padrão.
- Curva de aprendizado facilitada para equipes que constroem aplicações web e mobile.

Atualizações

- O React aborda as mudanças com cuidado;
 - Testado em áreas críticas para o negócio, com mais de um bilhão de usuários;
 - Mais de 100.000 componentes React na Meta ajudam os testes.
- Melhoria contínua e manutenção frequente da biblioteca.
 - A equipe do React está sempre pesquisando como melhorar o React;
 - Algumas pesquisas levam anos para ter retorno;
 - O React tem um alto padrão para levar uma ideia de pesquisa para a produção;
 - Apenas abordagens comprovadas se tornam parte do React.

Vantagens em se utilizar React

- **Componentização:** O React permite a criação de componentes reutilizáveis, o que facilita a manutenção e a escalabilidade do código;
- **Reatividade:** Quando os dados mudam, o React atualiza automaticamente a interface do usuário, tornando-a responsiva e eficiente;
- **Comunidade Ativa:** O React tem uma comunidade grande e ativa, o que significa que você pode encontrar muitos recursos, bibliotecas e soluções para problemas comuns;
- **Renderização no Lado do Servidor (Server-Side Rendering - SSR):** O React suporta SSR, o que melhora o desempenho e a experiência do usuário;
- **Integração com Ferramentas e Bibliotecas:** O React pode ser facilmente integrado com outras bibliotecas e ferramentas, como Redux, GraphQL e etc;
- **Performance:** Outra grande vantagem do React é sua performance excepcionalmente rápida. Ele utiliza uma técnica chamada Virtual DOM (DOM virtual) que permite que ele atualize apenas as partes necessárias da página quando ocorrem mudanças, resultando em tempos de carregamento muito mais rápidos e menor consumo de recursos do computador.

Desvantagens do React

- **Curva de Aprendizado:** Para iniciantes, a curva de aprendizado do React pode ser íngreme, especialmente se você não está familiarizado com conceitos como JSX e estados;
- **Complexidade:** À medida que o aplicativo cresce, a complexidade dos componentes e do gerenciamento de estados pode aumentar;
- **Boilerplate Code:** O React exige algum código inicial para configurar o ambiente de desenvolvimento, o que pode ser considerado boilerplate;
- **Decisões de Arquitetura:** O React não impõe uma arquitetura específica, o que significa que você precisa tomar decisões sobre como estruturar seu aplicativo;
- **Problemas de SEO:** conteúdo das SPAs é frequentemente dinâmico e gerado por JavaScript no lado do cliente, o que pode afetar a indexação de mecanismos de pesquisa como o Google.



Boilerplate é um termo usado na programação para se referir a **código padrão ou repetitivo** que precisa ser escrito em várias partes de um projeto. É como seguir uma receita básica toda vez que você cozinha algo, mesmo que os ingredientes principais mudem. No desenvolvimento de software, o boilerplate pode incluir configurações iniciais, estruturas básicas de arquivos, importações comuns e outras tarefas que são necessárias, mas não diretamente relacionadas à lógica específica do aplicativo. Simplificando, é o “enche-linguiça” que você precisa escrever antes de realmente começar a criar a funcionalidade principal do seu programa. 🚀

Evolução do React

- A biblioteca React foi criada em 2013 pelo Facebook;
 - Rapidamente ganhou destaque na comunidade de desenvolvedores;
 - Objetivo principal de tornar mais fácil a construção de aplicações web complexas.
- Amplamente utilizada para desenvolver aplicações web modernas e single-page applications (SPA);
 - Toda a aplicação é carregada de uma vez;
 - As mudanças de conteúdo são gerenciadas dinamicamente sem a necessidade de recarregar a página.
- A principal inovação do React foi o conceito de **componentes**;
- Com o tempo, o ecossistema do React cresceu significativamente. Surgiram muitas bibliotecas e ferramentas relacionadas, como o **Redux**, **React Router**, **Styled Components** e **Material-UI**;
- O **React** foi lançado em **2013** e já incluía a capacidade de criar **componentes funcionais** e **componentes de classe**;
 - Funções JavaScript que retornavam elementos JSX.
- React 16 (Fiber);
 - Trouxe uma reescrita completa do mecanismo de reconciliação e renderização;
 - Melhorias significativas no desempenho, especialmente para aplicativos com muitos componentes.

- Hooks (React 16.8+);
 - Os **Hooks** permitem que componentes funcionais tenham **estado, efeitos colaterais** e outras funcionalidades anteriormente disponíveis apenas em componentes de classe;
 - Simplificou o código e tornou a lógica do componente mais legível.
- Context API (React 16.3+);
 - Permite o compartilhamento de estado entre componentes sem a necessidade de **Redux** ou outras bibliotecas de gerenciamento de estado;
- Suspense e Lazy Loading (React 16.6+);
 - **Suspense** permite o **carregamento assíncrono de componentes e recursos**;
 - Com o **Lazy Loading**, os componentes são carregados apenas quando necessários, melhorando o desempenho e reduzindo o tempo de carregamento inicial.
- Server-Side Rendering (React 18).
 - O SSR envolve renderizar páginas da web no servidor antes de enviá-las para o navegador do cliente;
 - O servidor gera a marcação HTML final para uma página da web e enviando-a ao cliente.

Referências


React

React is the library for web and native user interfaces. Build user interfaces out of individual pieces called components written in JavaScript. React is designed to let you seamlessly combine

 <https://react.dev/>

Descubra React: a biblioteca criada pelo Facebook para facilitar a criação de páginas

Criada pelo time do Facebook e hoje usada também no Instagram e Whatapp e pensada para a criação de interfaces de usuário mais acessíveis, com facilidade

 <https://www.dio.me/articles/descubra-react-a-biblioteca-criada-pelo-facebook-para-facilitar-a-criacao-de-paginas>

A Evolução do React: o que você precisa saber para tomar a melhor decisão para o seu projeto

Você já ouviu falar de React? Se você está envolvido no desenvolvimento web, provavelmente sim! Mas será que você sabe tudo o que precisa sobre essa incrível biblioteca JavaScript? Desde a sua criação em 2013 até os dias atuais, o React evoluiu bastante. E é por isso que estamos aqui hoje: para disc



<https://www.linkedin.com/pulse/evolução-do-react-o-que-você-precisa-saber-para-tomar-melhor/?originalSubdomain=pt>