



# React para iniciantes





<title>

# AGENDA DO </> </> </> MÓDULO

</title>



Agenda do módulo



Anotações



{Introdução ao React}

{Como o React funciona}

{Configuração do ambiente}

{Criando o projeto}

{Configuração do projeto}

{Estrutura do React}



Atualmente, o React é uma das bibliotecas virtuais mais **prestigiadas** do mercado, com mais de 3,4 milhões de projetos.

<title>

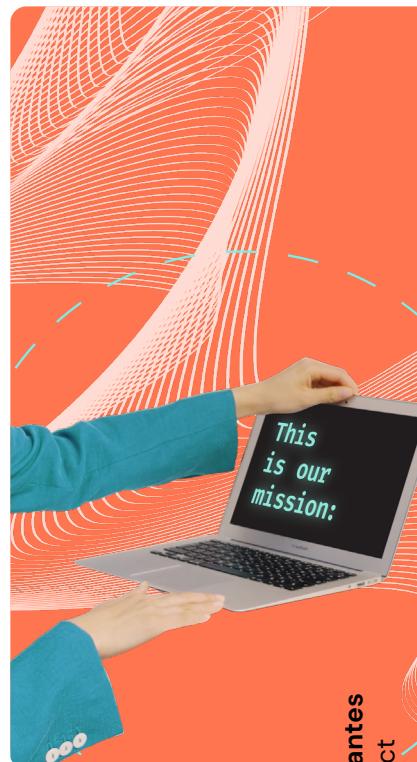
# MISSÃO DO MÓDULO

</title>

Iniciar sua jornada com o React de forma prática.

{01.}

## Introdução ao React



React para iniciantes  
Iniciando no React



# Framework

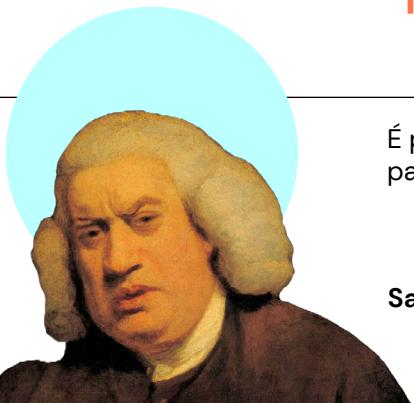
Framework, em tradução livre, significa estrutura. Em programação, essa estrutura é responsável por **trazer mais funcionalidades, códigos e comandos**.

O React se mantém como um framework por conseguir **atender às necessidades do projeto** em todos os pontos, sem precisar de outro framework para reger.

Utilizar um framework é acelerar o desenvolvimento do seu projeto, se for comparado a um desenvolvimento sem a utilização de um.



“



É preciso folhear meia biblioteca  
para fazer um livro.

Samuel Johnson

”



**Use  
alicerces fortes  
em seus códigos.**





# React

O **React** é uma biblioteca JS na teoria, mas como dito anteriormente, com suas novas funcionalidades e também a introdução dos Hooks (algo mais avançado, que será apresentado mais à frente no curso), acaba se tornando um **framework** na prática.

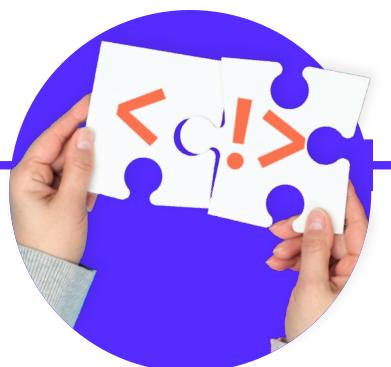
<title>

MÃO + + +  
NA MASSA

</title>



Acesse a documentação oficial do React:  
<https://pt-br.reactjs.org/>





## 2013

O React foi **lançado oficialmente em 2013** por Jordan Walke, que trabalhava no Facebook.

Logo, o React foi criado e é mantido pelo grupo Meta (antigo Facebook).

## 2010

Já em **2010**, o Facebook **implementou o XHP**, que é uma versão do **PHP**, para combater a vulnerabilidade das tentativas de inserção de códigos.

Porém, com a alta demanda de requisições, essa implementação falhou. Dessa forma, Jordan Walke teve o aval de conseguir conciliar o XHP com o JavaScript no browser.

O XHP já permitia a criação de **códigos em bloco**, transformando-os em códigos reutilizáveis, uma funcionalidade que é pilar do React.

## 2011



Seu primeiro protótipo foi criado em **2011**, com o nome **FaxJS**, que tinha como ferramenta um elemento de pesquisa para o Facebook.

## 2015

Em **2015**, a Netflix e o Airbnb passaram a utilizar o React, com isso, sua popularidade aumentou, tornando-o um **framework muito utilizado e conhecido pelo mundo**.

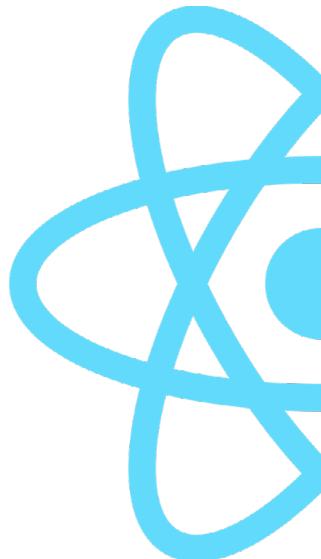
Em **2015** também foi lançada sua **versão mobile**, conhecida como **React Native**. Essa versão permite ao programador a criação de aplicativos nativos para Android e iOS.

Além de funcionar muito bem com o **ES6** e o **TypeScript** (que veremos mais à frente), funciona também com o **JSX**.



## Componetização

A vantagem de separar algumas partes por componentização é que podemos usar **componentes em diversas páginas**, reutilizando código.



## Compatibilidade

Além disso, com a **compatibilidade** com outras linguagens de programação como Ruby on Rails, Python, entre outros, torna-se um framework que **se conecta com diversas linguagens back-end**.

## React Dev Tools

O **React Dev Tools** é uma **expansão** para o Google Chrome muito utilizada pelos desenvolvedores.

## React Hot Loader

O **React Hot Loader** é um **plugin** que permite que o componente seja recarregado de forma síncrona sem a perda do estado.



O React é **muito prático** para fazer **sites completos**, onde você pode usar o mesmo menu em quase todas as páginas, pois podemos **criar um componente que atenda a todas as páginas**.

Além disso, o React possui **gerenciamento de estado**, que é quando você pode mudar o estado de uma parte do código de forma interativa. Uma boa aplicação do React é para uma **SPA**, que significa aplicação de página única.

## SPA

Uma SPA é uma página que vai **carregar o essencial da aplicação com diversas interações**. É uma ferramenta simples, com pouco código e muito poderosa.

Exemplos interessantes de SPA são: ToDo List, Gmail e Airbnb, onde, ao procurar um imóvel, caso utilize filtros, você não é levado para outra página, mas sim ao carregamento da página com novos resultados.

# {02.}

# Como o React funciona



Uma SPA tende a **renderizar** o essencial da aplicação apenas uma vez para tornar a página mais rápida. Conforme o usuário for utilizando aquela página, outras informações vão sendo carregadas por demanda.

Quando usamos o Gmail, por exemplo, temos um vislumbre do que é uma SPA.

Existem componentes estáticos na tela como a barra de menu, barra lateral, header etc.

## Airbnb e SPA: como funciona?

Quando o usuário vai pesquisar por um local, aparecem diversas opções. A partir dos filtros, o usuário vai selecionando o que mais lhe agrada.

Com isso, a página também mantém certos componentes para preservar o tempo de recarregamento e atualiza somente a parte das opções que serão filtradas.

Mas como todo tipo de aplicação, a SPA também possui algumas **desvantagens**.

### 1 Falta de segurança

Por ser uma SPA, podem existir mais **brechas na segurança** que o normal, além de ser mais difícil de testar. Isso traz instabilidade e facilita possíveis ataques de hackers.



## 2 SEO

SEO é a **otimização de mecanismos de busca**. Basicamente, por ser uma SPA, não geramos uma URL para cada conteúdo, o que **prejudica a escalabilidade nas buscas**.

## 3 Primeiro carregamento demorado

O conteúdo que precisa ser renderizado pode levar **mais tempo que o normal**. É por isso que leva mais tempo para a página carregar por completo.

---

Os princípios de um framework são criados para **evitar que desviem do propósito original daquela ferramenta**.

Com isso, seus criadores resolveram conceber os princípios básicos para que a comunidade de desenvolvedores não altere seu propósito e consiga contribuir de forma linear com a ideia original.

O **primeiro princípio**, e talvez o mais importante deles, é a **componentização** (ou composição).

Quando se está trabalhando em conjunto, todos os que estão envolvidos devem compreender e conseguir contribuir para que o projeto saia de acordo com o que foi sugerido.

No caso, os componentes feitos separadamente devem dar certo de forma conjunta.

1



O **segundo princípio** é sobre a **abstração comum**.

O React é uma ferramenta que consegue implementar bastantes abstrações personalizadas. O problema é que, por conta disso, surgem componentes que podem ter funcionalidades incompatíveis com o React, além de componentes similares.

E para não ocorrer esse tipo de competição e incompatibilidade, é preferível implementar os recursos como parte do React.

2

O **terceiro princípio** é a **saída de emergência**.

O React é muito pragmático, além de ser mantido pelo grupo Meta.

3

As atualizações vão de acordo com a necessidade do grupo, porém, como se torna uma ferramenta muito utilizada, acaba que deve se adequar a outras funcionalidades para que não caia em desuso e continue popular.

O **último princípio** que mencionaremos aqui, lembrando que há outros, será sobre a estabilidade.

4

Estabilidade no sentido de que haverá mudanças na ferramenta, porém sua essência continuará a mesma.

A cada atualização que o React traz, ele volta com novas funcionalidades, respeitando os outros princípios e melhorando a ferramenta.



<title>

# PULO DO GATO

</title>

- O uso do JSX não é obrigatório, mas é recomendável!

## JSX

A **JSX** é uma sintaxe do JS. O uso da JSX é recomendável, mas não obrigatório.

Com o uso dela, você consegue renderizar de forma **mais concisa** as lógicas de interatividade, como a mudança de estado.

O JSX não é interpretado pelo navegador, sendo assim, precisa de um **transpilador** para transformar tudo em JavaScript, inserindo o HTML.

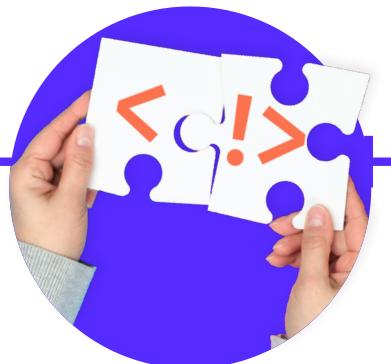




<title>

# MÃO + + NA MASSA

</title>



Abra o código dos exemplos em JS e JSX  
e analise as diferenças e semelhanças em:

- Tamanho do código;
- Sintaxe;
- Funções;
- Variáveis.

## {03.}

# Configuração do ambiente



Para começar a trabalhar com React, é necessário ter algumas ferramentas. As principais são o **VS Code**, o **Terminal do Windows** e o **NodeJS**.

<title>

# PULO DO GATO

</title>

- Não use o VS Code Web.

Para instalar o terminal do Windows, acesse este link:

[Clique aqui!](#)

Para instalar o Node, acesse este link:

[Clique aqui!](#)





## Para usuários de Mac

O brew é um gerenciador de pacotes comumente utilizado por usuários do Mac.

Caso ainda não o tenha, acesse o link.

[Clique aqui!](#)

## Passo a passo Linux

Para usuários de Linux, acesse o seguinte link:

[Clique aqui!](#)

## NVM

O **NVM** significa **Node Version Manager**, é um gerenciador de versões do Node.

## NPM

Já o **NPM** significa **Node Package Manager**, que é um gerenciador de pacotes do Node, servindo para a instalação de bibliotecas, início de projeto, entre outros.



<title>

# MÃO + + NA MASSA

</title>

Baixe e instale os aplicativos sugeridos para o seu sistema operacional.



## {04.}

# Criando o projeto

## Vite

O **Vite** é uma **build tool** (ferramenta de construção) originalmente criada para ser utilizada com outro framework. Porém, com suas atualizações, ela também pode ser usada para criar aplicações em React.



<title>

# PULO DO GATO

</title>

- O Vite é recomendado para projetos pequenos.



## Ele se atualiza!

Com o Vite o desenvolvedor cria uma estrutura mais simples e com o padrão atual dos browsers.

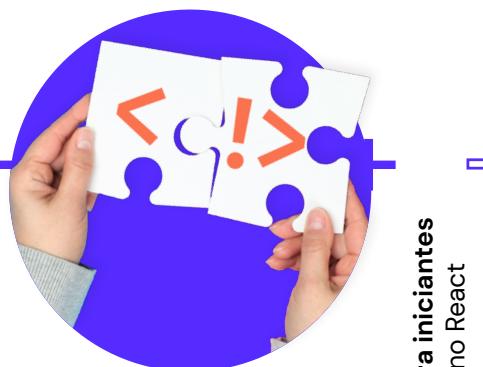
Pelo Vite, você pode criar uma aplicação usando React com base no JS ou escolher criar uma aplicação com base em TS (TypeScript).

<title>

# MÃO + + + NA MASSA

</title>

Crie um projeto do Vite no VS Code, seguindo os passos apresentados nesta aula.





# {05.}

# Configuração do projeto

## Plugins

Plugins são **ferramentas adicionais** que auxiliam e personalizam a aplicação ou o código para trazer melhorias e evitar erros simples.

Também existem plugins que auxiliam no desenvolvimento, como atalhos e dicas para o seu código ficar mais limpo e legível.

---

Para os projetos que serão realizados neste curso, precisaremos de **três plugins** para auxiliar no desenvolvimento.

Os plugins que serão utilizados nos projetos são **CSS Modules**, **vscode-styled-components** e **Color Highlight**.

Para baixar esses plugins, basta ir a Extensões no VSCode. Digite o nome de cada um dos plugins e instale-os.



<title>

# PULO DO GATO

</title>

- Roteirize todas as ações que os usuários devem fazer ao clonar o projeto e tentar rodar na máquina local.



## ESlint e Prettier

O **ESlint** é uma ferramenta plugável capaz de localizar e corrigir possíveis erros de sintaxe e problemas que estejam atrapalhando o andamento da aplicação.

Já o **Prettier** é um formatador de código para tornar o código mais limpo e padronizado, muito utilizado no VSCode e um bom complemento para o ESlint, pois vão agindo em conjunto para manter a estrutura do código e sua arquitetura mais limpa.



<title>

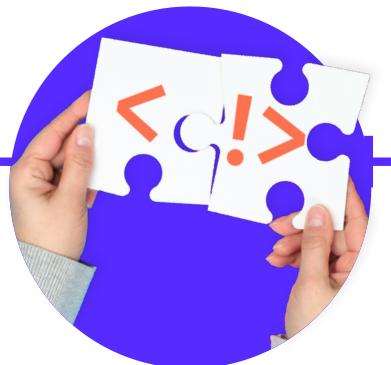
# MÃO + + NA MASSA

</title>

Instale os plugins:

- CSS Modules
- Vscode-styled-component
- Color Highlight
- ESlint
- Prettier

Comece um novo projeto no GitHub e crie um rascunho para o Readme.

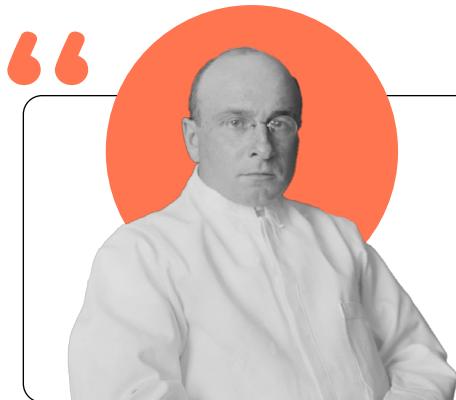


## {06.}

# Estrutura do React



Para criar um novo projeto, você pode utilizar o comando **npx react-create-app nome do projeto ou npm init vite@latest.**



O Homem que não soube organizar um mundo para si mesmo é um estranho no mundo que ele mesmo criou.

Alexis Carrel



Caso você tenha uma aplicação com múltiplas páginas, você terá que criar uma rota para conseguir navegar entre elas.



<title>

# PULO DO GATO

</title>

- Separe o seu projeto em partes de forma inteligente, focando em reaproveitar códigos e deixar o projeto escalável.



<title>

# MÃO + + NA MASSA

</title>

- Inicie um novo projeto.
- Configure as pastas.
- Faça o projeto rodar.





<title>

# DESAFIO CONQUER

</title>



```
if (challenge)
// start
```

<body>

Sem consultar em lugar nenhum,  
responda ao nosso quiz!

Você encontra o link no Material  
de Apoio deste módulo.

</body>



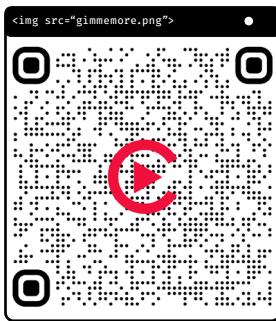
<title>

# QUERO MAIS

</title>



Guia de informações  
sobre aplicativos de  
página única.



Guia de informações  
sobre aplicativos de  
página única.



```
if (interested)
// gimmemore
```





## Anotações

- 
- 
- 
- 
- 
- 

