



# Estrutura de funcionamento do React

Ponto de montagem (mounting point)

Arquivo de script principal

---

## Ponto de montagem (mounting point)

*Levando em consideração uma aplicação criada com o Vite*

- O arquivo `index.html` é o primeiro arquivo que a nossa aplicação realiza o carregamento;
  - É o ponto de montagem para uma aplicação React.
- Nele, você encontrará uma tag `<div>` com o atributo `id="root"` ;
  - Raiz do DOM onde o React irá inserir e exibir todo o conteúdo da aplicação.
- Chama o script do arquivo `main.tsx | main.jsx` :
  - Importação de um script com `type=module` , para realizar a importação e exportação de módulos (arquivos JavaScript separados) de forma nativa nos navegadores.

```
<script type="module" src="/src/main.tsx"></script>
```

## Arquivo de script principal

- `main.jsx | main.tsx` OU `index.jsx | main.tsx`

- Realizada a importação do `ReactDOM`;
  - Biblioteca que permite a integração da estrutura básica do React com o DOM (*Document Object Model*);
  - Representação do HTML de uma página no JavaScript;
  - Ele gerencia a renderização dos componentes React no navegador, atualizando o DOM conforme necessário;
  - Responsável por sincronizar o estado do aplicativo React com a representação visual na página web.
- Quando é realizada a importação do ReactDOM, estamos integrando o React para funcionar no ambiente web através dos navegadores;
- `ReactDOM.createRoot(document.getElementById('root'))`
  - `ReactDOM.createRoot()`
    - O método `createRoot()` é uma função fornecida pelo `ReactDOM`;
    - Cria uma raiz de renderização para o aplicativo React;
    - Essa raiz é o ponto de partida para a renderização assíncrona, que é uma técnica que melhora o desempenho da atualização do DOM;
    - Em vez de usar o método tradicional `ReactDOM.render()`, que é síncrono, o `createRoot()` permite que o React realize atualizações de forma assíncrona, otimizando a experiência do usuário.
  - `document.getElementById('root')`
    - Aqui, estamos obtendo o elemento HTML com o ID `'root'`;
    - Esse é o mesmo elemento que encontramos no arquivo `index.html` onde o React monta o aplicativo;
    - O React irá renderizar os componentes dentro deste elemento.
  - `root.render(...)`
    - O método `render()` é usado para renderizar componentes React dentro da raiz de renderização criada;
    - Está sendo renderizado o componente `<App />` dentro de um `<React.StrictMode>`;
    - O `<React.StrictMode>` é um componente especial do próprio React

- Ajuda a identificar problemas potenciais no código durante o desenvolvimento;
- Ele não afeta a produção;
- Verifica práticas inseguras e desencoraja o uso de APIs obsoletas.

```
ReactDOM.createRoot(document.getElementById('root')).render(  
  <React.StrictMode>  
    <App />  
  </React.StrictMode>,  
)
```



O operador de exclamação (!) força a conversão de um valor que pode ser nulo ou indefinido em um valor não nulo. No exemplo dado, ele garante que o elemento com o ID "root" existe antes de renderizar o aplicativo React.