

Iniciando com React

Link para o Notion: <https://cherry-client-b8f.notion.site/Iniciando-com-React-258911d84e0d80489fbfd5086316f99d?pvs=73>

O que é o React?

React é uma biblioteca JavaScript de código aberto, criada pelo Facebook, usada para construir interfaces de usuário interativas em aplicações web e móveis. É conhecida por sua abordagem baseada em componentes, que permite criar partes reutilizáveis da interface, tornando o desenvolvimento mais eficiente e organizado.

Por que utilizar o Vite?

O Vite faz um papel fundamental no setup inicial de um projeto em React, executando tarefas como:

- Ele usa o esbuild (super rápido, escrito em Go) para transformar JSX/TSX em JavaScript e aplicar transformações básicas (como importação de CSS e assets).
- O Vite intercepta os imports e os serve via servidor local, aplicando Hot Module Replacement (HMR) para React (com @vitejs/plugin-react).
- Na versão do site para produção, o Vite:
 - Transpila JSX/TSX → usando esbuild internamente, que é chamado pelo plugin @vitejs/plugin-react.
 - Bundling com Rollup → junta todos os módulos em arquivos finais, aplicando code splitting (divisão em chunks).
 - Tree-shaking → remove código não utilizado (ex.: funções da lib que não foram importadas).
 - Minificação → usa o esbuild para minificar.
 - Otimização de assets → imagens, CSS, fontes, etc. são tratados pelo Rollup e injetados nos bundles finais.
 - Geração do dist/ → com HTML, JS e CSS prontos para deploy.

Como iniciar um novo projeto utilizando Vite?

1. No terminal, digite:

```
npm create vite@latest
```

2. O terminal irá pedir sua confirmação para instalar, pressione a tecla Enter.

```
Need to install the following packages:
create-vite@7.1.1
Ok to proceed? (y)
```

3. Digite o nome do projeto

```
Project name:
| vite-project
```

4. Selecione entre as opções de Framework, o React, com a seta do teclado.

```
Select a framework:
| O Vanilla
| O Vue
| ● React
| O Preact
| O Lit
| O Svelte
| ...
```

5. Selecione a opção somente Javascript

```
Select a variant:
|  O TypeScript
|  O TypeScript + SWC
|  ● JavaScript
|  O JavaScript + SWC
|  O React Router v7 ↗
|  O TanStack Router ↗
|  ...
```

6. Agora execute os comandos listados pelo Vite.

```
Done. Now run:

cd nome-projeto
npm install
npm run dev
```

Alternativa mais rápida

Você também pode criar de uma maneira mais curta, utilizando o template de projeto com o React.

```
npm create vite@latest vite-project -- --template react
```

Sendo:

1. `npm create vite@latest`

- `npm create` é um atalho do npm para executar scripts de criação de projetos.
- `vite@latest` especifica que você quer usar a versão mais recente do **Vite**, que é um bundler moderno e rápido para projetos web.

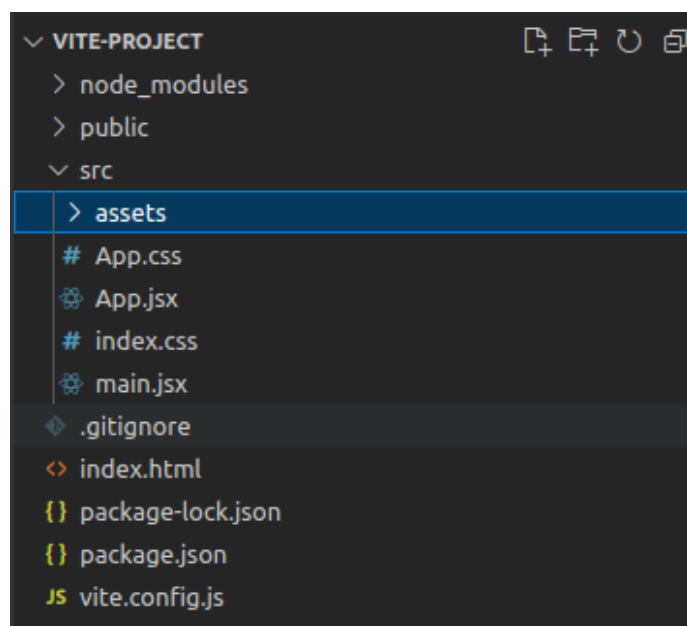
2. `vite-project`

- É o **nome da pasta** onde seu projeto será criado.
- Se a pasta não existir, ela será criada.

3. `-- --template react`

- O primeiro `--` é necessário para separar os argumentos do `npm` dos argumentos que você quer passar para o Vite.
- `-template react` diz ao Vite para usar o **template do React**, ou seja, ele já vai criar o projeto com configuração inicial de React (JSX, estrutura de pastas, etc.).

Estrutura de pastas



Papel de cada pasta/arquivo

node_modules/

- Criada após instalar dependências com `npm install`.
- Contém todas as bibliotecas do projeto (React, Vite, etc.).
- **Nunca** editar ou versionar no Git.

public/

- Arquivos estáticos acessados **diretamente** via URL, sem serem processados pelo Vite.
- Exemplos: `favicon.ico`, imagens grandes, PDFs.
- Se você colocar `public/meu-arquivo.png`, pode acessar em `http://localhost:5173/meu-arquivo.png`.

index.html

- Único HTML da aplicação (React vai "tomar conta" dele).
- Nele existe um `<div id="root"></div>` onde o React **monta** a aplicação.
- O `<script type="module" src="/src/main.jsx"></script>` carrega o JavaScript inicial.

src/

Aqui vive todo o **código da aplicação**.

- **main.jsx**
 - Arquivo de entrada.
 - Monta o React no `#root` do `index.html`.
- **App.jsx**
 - Componente principal (geralmente guarda a estrutura base, rotas etc.).
- **App.css**
 - Estilos específicos para o `App.jsx`.
- **index.css**
 - Estilos globais (reset de CSS, variáveis, fontes).
- **assets/**
 - Onde você coloca imagens, ícones, fontes e até arquivos `.js` ou `.ts` de utilitários.
 - Diferente do `public/`, aqui os arquivos são **processados pelo Vite**, o que permite otimizações (ex.: minificação, cache busting).
- **package.json**
 - Lista todas as dependências do projeto (`react`, `react-dom`, `vite` etc.).
 - Contém scripts, por exemplo:

```
{
  "scripts": {
    "dev": "vite",
    "build": "vite build",
    "preview": "vite preview"
  }
}
```

- **vite.config.js**
 - Configurações do Vite (plugins, alias de importação etc.).
- **eslint.config.js**

- Configuração do ESLint (padronização de código, dicas de boas práticas).
- **.gitignore**
 - Diz ao Git quais arquivos/pastas ignorar (ex.: `node_modules` , `.env`).

Referências

<https://react.dev/>

<https://vite.dev/>

<https://www.goodcore.co.uk/blog/reactjs-vs-angular/>