# Passo 1: Configurar o Projeto com Vite e TypeScript

1. Instale o Node.js e npm:
Certifique-se de que você tem o Node.js e o npm instalados. Você pode baixar a versão LTS do Node.js no site
oficial.
2. Crie um Novo Projeto com Vite e TypeScript:
- Abra o terminal ou o prompt de comando.
- Execute o comando abaixo para criar um novo projeto com Vite e TypeScript. Substitua 'meu-projeto' pelo nome do
seu projeto.
npm create vite@latest meu-projetotemplate react-ts
3. Navegue até o Diretório do Projeto:
- Entre no diretório do seu projeto recém-criado.
cd meu-projeto
4. Instale as Dependências:
- Execute o comando abaixo para instalar todas as dependências necessárias.
npm install
5. Inicie o Servidor de Desenvolvimento:

- Após a instalação das dependências, inicie o servidor de desenvolvimento.

npm run dev

- O Vite iniciará um servidor de desenvolvimento e fornecerá um link (geralmente http://localhost:3000) para visualizar seu projeto no navegador.

### Passo 2: Criação dos Componentes com Props

Agora que o projeto está configurado, vamos criar componentes que utilizam props.

1. Modifique o arquivo 'src/App.tsx':

Abra o arquivo 'src/App.tsx' e modifique o código para incluir componentes que recebem props:

```
import React from 'react';
import './App.css';

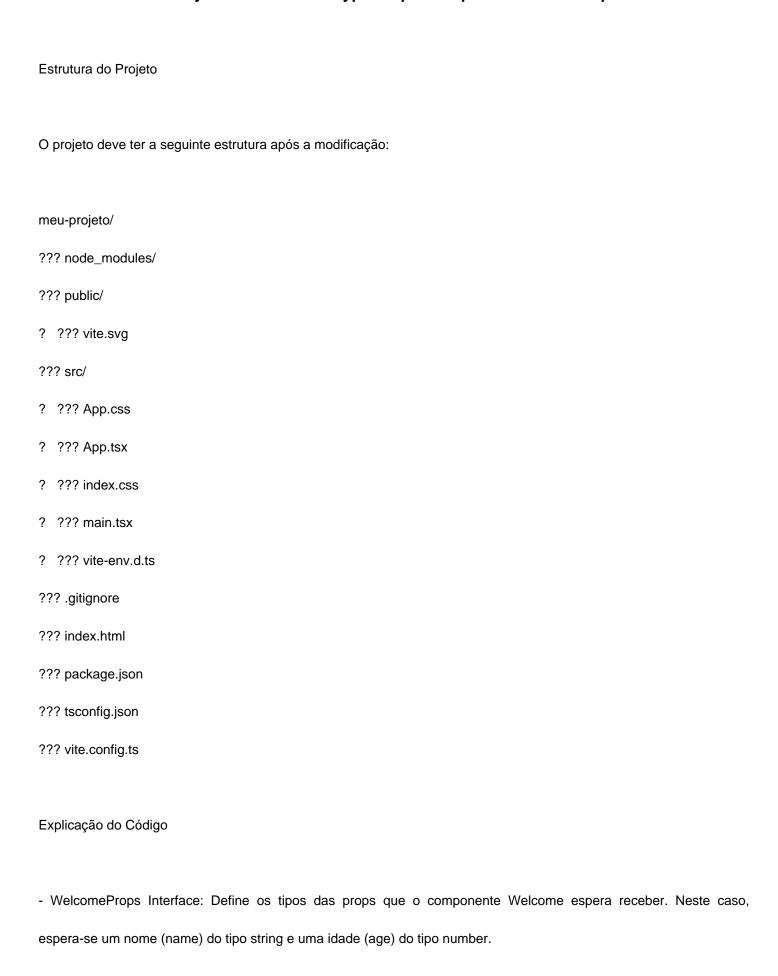
// Definindo uma interface para as props
interface WelcomeProps {
   name: string;
   age: number;
}

// Componente funcional que utiliza props
const Welcome: React.FC<WelcomeProps> = ({ name, age }) => {
    return (
```

```
<div>
      <h1>Hello, \{name\}! < /h1>
      You are {age} years old.
    </div>
  );
};
function App() {
 return (
    <div className='App'>
      <Welcome name='Alice' age=\{25\} />
      <Welcome name='Bob' age=\{30\} />
      <Welcome name='Charlie' age=\{35\} />
    </div>
  );
}
export default App;
```

#### 2. Salve e visualize o resultado:

Com o servidor de desenvolvimento rodando, você verá que cada vez que você salva, a página no navegador atualiza automaticamente para refletir as mudanças. A aplicação deve mostrar três mensagens de boas-vindas personalizadas com base nas props passadas.



- Welcome Componente Funcional: Este componente recebe as props tipadas e as utiliza para renderizar uma mensagem de boas-vindas personalizada.

- App Componente: O componente principal (App) utiliza o componente Welcome várias vezes, passando diferentes valores para as props name e age.

#### Conclusão

Esse projeto simples exemplifica como você pode utilizar componentes com props em React com TypeScript. Ele mostra como definir tipos para props usando interfaces, como passar dados para os componentes e como renderizar esses dados.