

# Exemplo de Projeto React com TypeScript

## Passo 1: Configuração do Projeto

1. Crie um novo projeto React com TypeScript usando Vite:

```
npm create vite@latest meu-projeto -- --template react-ts
```

```
cd meu-projeto
```

```
npm install
```

```
npm run dev
```

2. Estrutura do Projeto:

Certifique-se de que a estrutura básica do projeto é semelhante a esta:

```
meu-projeto/
```

```
??? node_modules/
```

```
??? public/
```

```
??? src/
```

```
?   ??? App.tsx
```

```
?   ??? index.css
```

```
?   ??? main.tsx
```

```
?   ??? components/
```

```
?     ??? Timer.tsx
```

```
?     ??? Message.tsx
```

```
??? .gitignore
```

```
??? index.html
```

```
??? package.json
```

## Exemplo de Projeto React com TypeScript

??? tsconfig.json

??? vite.config.ts

### Passo 2: Implementação dos Componentes

#### 1. Componente Timer (Timer.tsx):

Este componente exibirá um temporizador que atualiza a cada segundo usando useEffect.

Código:

```
import { useState, useEffect } from 'react';
```

```
function Timer() {  
  const [seconds, setSeconds] = useState(0);
```

```
  useEffect(() => {  
    const interval = setInterval(() => {  
      setSeconds(prev => prev + 1);  
    }, 1000);
```

```
    return () => clearInterval(interval);  
  }, []);
```

```
  return (
```

## Exemplo de Projeto React com TypeScript

```
<div>

  <h2>Tempo Decorrido: {seconds} segundos</h2>

</div>

);

}
```

```
export default Timer;
```

### 2. Componente Message (Message.tsx):

Este componente exibirá uma mensagem que muda após 5 segundos usando useEffect.

Código:

```
import { useState, useEffect } from 'react';

function Message() {

  const [message, setMessage] = useState('Bem-vindo ao projeto!');

  useEffect(() => {

    const timer = setTimeout(() => {

      setMessage('Obrigado por visitar!');

    }, 5000);

  });

}
```

## Exemplo de Projeto React com TypeScript

```
    return () => clearTimeout(timer);  
  }, []);
```

```
    return (  
      <div>  
        <h3>{message}</h3>  
      </div>  
    );  
  }  
}
```

```
export default Message;
```

### 3. Componente Principal (App.tsx):

O componente principal App importará e renderizará os componentes Timer e Message.

Código:

```
import './App.css';  
  
import Timer from './components/Timer';  
  
import Message from './components/Message';  
  
function App() {  
  return (  

```

## Exemplo de Projeto React com TypeScript

```
<div className='App'>

  <h1>Exemplo de React com TypeScript</h1>

  <Timer />

  <Message />

</div>

);

}

export default App;
```

### Passo 3: Executar o Projeto

1. Execute o servidor de desenvolvimento:

```
npm run dev
```

2. Verifique a aplicação no navegador:

Acesse o endereço fornecido pelo Vite (geralmente <http://localhost:3000>) para ver os componentes em ação.

### Explicação do Código

- Timer Component:

Usa `useState` para manter o estado dos segundos.

Usa `useEffect` para iniciar um intervalo que incrementa o contador de segundos a cada segundo. O `useEffect` é

## Exemplo de Projeto React com TypeScript

reexecutado toda vez que o valor do intervalo muda, o que faz com que o intervalo anterior seja limpo e um novo intervalo seja configurado.

### - Message Component:

Usa `useState` para manter a mensagem atual.

Usa `useEffect` para definir um temporizador que altera a mensagem após 5 segundos. O `return` limpa o temporizador se o componente for desmontado antes do tempo expirar.

### - App Component:

Gerencia o estado e renderiza `Timer` e `Message`, compondo a interface da aplicação.