Passo 1: Configuração do Projeto

Crie um novo projeto React com TypeScript usando Vite:	
npm create vite@latest meu-projetotemplate 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	

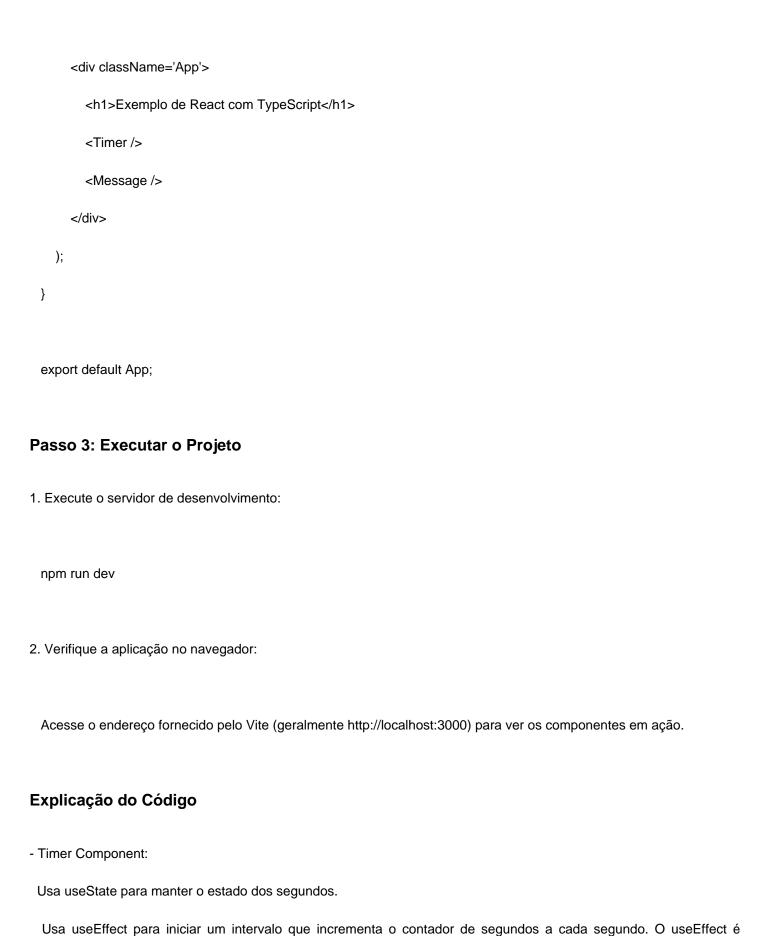
??? tsconfig.json ??? vite.config.ts

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 (

```
<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);
```

return () => clearTimeout(timer);
}, []);
return (
<div></div>
<h3>{message}</h3>
);
}
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 (



reexecutado	toda	vez	que	o valor	do	intervalo	muda,	0	que	faz	com	que	o i	intervalo	anterior	seja	limpo	е	um	novo
intervalo seja	a conf	igura	ado.																	

- 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.