



Setup Completo: React + Vite + TypeScript + Tailwind


Guia passo a passo para configurar um ambiente de desenvolvimento React moderno e completo.

Pré-requisitos do Ambiente

1. Node.js e npm

-  **Node.js 22+** instalado
-  **npm 9+** (vem com Node.js)
- Verificar versões: `node --version` e `npm --version`

2. Editor de Código

-  **Cursor** ou **VS Code**
 - **Extensões obrigatórias:**
 - ES7+ React/Redux/React-Native snippets
 - Tailwind CSS IntelliSense
 - Prettier - Code formatter
 - ESLint
-

Setup do Projeto

3. Criar Projeto Base

```
bash

# Opção 1: Novo projeto
npm create vite@latest nome-do-projeto -- --template react-ts
cd nome-do-projeto

# Opção 2: Aproveitar projeto existente (se já tiver)
# Continuar com os próximos passos
```

4. Instalar Dependências Principais

```
bash

# HTTP Requests e Roteamento
npm install axios react-router-dom

# API Fake para desenvolvimento
npm install -D json-server
```

5. Configurar Tailwind CSS

bash

Instalar Tailwind v3 (estável)

`npm install -D tailwindcss@^3.4.0 postcss autoprefixer`

Criar arquivos de configuração

`npx tailwindcss init -p`

Configurações Essenciais

6. Configurar `tailwind.config.js`

javascript

```
/** @type {import('tailwindcss').Config} */
export default {
  content: [
    './index.html',
    './src/**/*.{js,ts,jsx,tsx}',
  ],
  theme: {
    extend: {},
  },
  plugins: [],
}
```

7. Configurar CSS Principal (`src/index.css`)

CSS

```
@tailwind base;
@tailwind components;
@tailwind utilities;
```

8. Teste Visual (opcional)

Substituir `src/App.tsx` temporariamente:

tsx

```
function App() {
  return (
    <div className="min-h-screen bg-gradient-to-br from-blue-500 to-purple-600 flex items-center justify-center">
      <div className="bg-white rounded-lg shadow-xl p-8 max-w-md w-full mx-4">
        <h1 className="text-3xl font-bold text-gray-800 text-center mb-4">
          🛒 Projeto React
        </h1>
        <p className="text-gray-600 text-center">
          Ambiente configurado com sucesso!
        </p>
        <div className="mt-6 flex justify-center">
          <button className="bg-blue-500 hover:bg-blue-600 text-white px-6 py-2 rounded-lg transition-colors">
            Começar Desenvolvimento
          </button>
        </div>
      </div>
    </div>
  )
}

export default App
```

✅ Teste Final

9. Executar Projeto

```
bash

npm run dev
```

Resultado esperado:

- ✅ Servidor local rodando (ex: <http://localhost:5173>)
- ✅ Tailwind funcionando (gradiente e estilos aplicados)
- ✅ Sem warnings no terminal
- ✅ TypeScript funcionando

🎯 Stack Final Instalada

Core:

- **React 19.1+** - Framework principal
- **TypeScript 5.8+** - Tipagem
- **Vite 7.1+** - Build tool ultra-rápido

Styling & UI:

- **Tailwind CSS 3.4+** - Framework CSS utility-first
- **PostCSS + Autoprefixer** - Processamento CSS

Routing & HTTP:

- **React Router 7.8+** - Roteamento SPA
- **Axios 1.11+** - HTTP requests

Development:

- **json-server 1.0-beta** - API fake para testes
- **ESLint + Prettier** - Qualidade de código
- **TypeScript types** - Tipagem completa



Scripts Disponíveis

bash

`npm run dev` # Servidor de desenvolvimento

`npm run build` # Build para produção

`npm run preview` # Preview do build

`npm run lint` # Verificar código

`npm run format` # Formatar código



Para o Próximo Projeto

1. Copie este guia
2. Execute os comandos em ordem
3. Ajuste nome do projeto e dependências específicas
4. Tempo total: ~5 minutos



Próximos Passos Após Setup

1. **Estruturar pastas:** `/src/components`, `/src/pages`, `/src/contexts`, `/src/services`
 2. **Configurar roteamento** com React Router
 3. **Criar componentes base:** Header, Footer, Layout
 4. **Configurar json-server** com `db.json`
 5. **Implementar Context API** para estado global
-

✨ *Ambiente React moderno pronto em poucos comandos!*