

Guia Técnico do Usuário – Aplicativo CuidadoSenior (Versão 2.0)

1. Introdução

O **CuidadoSenior** é um aplicativo multiplataforma desenvolvido para oferecer suporte a idosos e familiares, com funcionalidades de **cadastro de contatos**, **ligações de emergência** e **interface acessível**.

A aplicação foi construída com **React + TypeScript** utilizando o **Vite** como ferramenta de build e empacotada com **Tauri**, framework que permite gerar aplicativos nativos de alto desempenho usando tecnologias web e backend em **Rust**.

Este guia técnico tem como objetivo auxiliar desenvolvedores e usuários avançados a **configurar o ambiente de desenvolvimento**, **executar o projeto localmente** e **gerar o executável (.exe)** do CuidadoSenior.

2. Requisitos de Ambiente

Para executar ou compilar o projeto, é necessário ter o seguinte ambiente configurado:

2.1. Softwares Necessários

Node.js – versão 18 ou superior
<https://nodejs.org>


npm – gerenciador de pacotes (instalado junto com o Node.js)

Git – para clonar o repositório
<https://git-scm.com>

Rust – linguagem utilizada pelo Tauri para compilar o executável
<https://www.rust-lang.org/tools/install>

Tauri CLI – interface de linha de comando para empacotamento

```
npm install -g @tauri-apps/cli
```

Visual Studio Code (recomendado)
 <https://code.visualstudio.com>

2.2. Dependências de Sistema (Windows)

- **Visual Studio Build Tools 2022**
 - Durante a instalação, marque o componente **“Desktop Development with C++”**.
- **Rustup** (instalador do compilador Rust).
- **Permissões administrativas** (necessárias para gerar o executável).

3. Clonando o Projeto

Abra o terminal (PowerShell, CMD ou Git Bash) e execute:

```
# Clonar o repositório
git clone https://github.com/SEU_USUARIO/CuidadoSenior.git

# Entrar na pasta do projeto
cd CuidadoSenior
```

4. Instalação das Dependências

Com o Node.js e o npm configurados, instale as dependências do projeto:

```
npm install
```

Esse comando fará o download de todas as bibliotecas necessárias para o front-end e o ambiente Tauri.

```
{
  "name": "vite_react_shadcn_ts",
  "private": true,
  "version": "0.0.0",
  "type": "module",
  "scripts": {
    "dev": "vite --port 5173",
    "build": "vite build",
    "build:dev": "vite build --mode development",
    "lint": "eslint .",
    "preview": "vite preview"
  },
  "dependencies": {
    "@hookform/resolvers": "^3.10.0",
    "@radix-ui/react-accordion": "^1.2.11",
    "@radix-ui/react-alert-dialog": "^1.1.14",
    "@radix-ui/react-aspect-ratio": "^1.1.7",
    "@radix-ui/react-avatar": "^1.1.10",
    "@radix-ui/react-checkbox": "^1.3.2",
```

```

"@radix-ui/react-collapsible": "^1.1.11",
"@radix-ui/react-context-menu": "^2.2.15",
"@radix-ui/react-dialog": "^1.1.14",
"@radix-ui/react-dropdown-menu": "^2.1.15",
"@radix-ui/react-hover-card": "^1.1.14",
"@radix-ui/react-label": "^2.1.7",
"@radix-ui/react-menubar": "^1.1.15",
"@radix-ui/react-navigation-menu": "^1.2.13",
"@radix-ui/react-popover": "^1.1.14",
"@radix-ui/react-progress": "^1.1.7",
"@radix-ui/react-radio-group": "^1.3.7",
"@radix-ui/react-scroll-area": "^1.2.9",
"@radix-ui/react-select": "^2.2.5",
"@radix-ui/react-separator": "^1.1.7",
"@radix-ui/react-slider": "^1.3.5",
"@radix-ui/react-slot": "^1.2.3",
"@radix-ui/react-switch": "^1.2.5",
"@radix-ui/react-tabs": "^1.1.12",
"@radix-ui/react-toast": "^1.2.14",
"@radix-ui/react-toggle": "^1.1.9",
"@radix-ui/react-toggle-group": "^1.1.10",
"@radix-ui/react-tooltip": "^1.2.7",
"@tanstack/react-query": "^5.83.0",
"class-variance-authority": "^0.7.1",
"clsx": "^2.1.1",
"cmdk": "^1.1.1",
"date-fns": "^3.6.0",
"dexie": "^4.2.1",
"embla-carousel-react": "^8.6.0",
"input-otp": "^1.4.2",
"lucide-react": "^0.462.0",
"next-themes": "^0.3.0",
"react": "^18.3.1",
"react-day-picker": "^8.10.1",
"react-dom": "^18.3.1",
"react-hook-form": "^7.61.1",
"react-resizable-panels": "^2.1.9",
"react-router-dom": "^6.30.1",
"recharts": "^2.15.4",
"sonner": "^1.7.4",
"tailwind-merge": "^2.6.0",
"tailwindcss-animate": "^1.0.7",
"vaul": "^0.9.9",
"zod": "^3.25.76"
},
"devDependencies": {
  "@eslint/js": "^9.32.0",
  "@tailwindcss/typography": "^0.5.16",

```

```

"@tauri-apps/api": "^2.8.0",
"@tauri-apps/cli": "^2.8.4",
"@types/node": "^22.16.5",
"@types/react": "^18.3.23",
"@types/react-dom": "^18.3.7",
"@vitejs/plugin-react-swc": "^3.11.0",
"autoprefixer": "^10.4.21",
"eslint": "^9.32.0",
"eslint-plugin-react-hooks": "^5.2.0",
"eslint-plugin-react-refresh": "^0.4.20",
"globals": "^15.15.0",
"lovable-tagger": "^1.1.9",
"postcss": "^8.5.6",
"tailwindcss": "^3.4.17",
"typescript": "^5.8.3",
"typescript-eslint": "^8.38.0",
"vite": "^5.4.19"
}
}

```

5. Estrutura do Projeto

```
CuidadoSenior/
├── src/                                # Código-fonte React + TypeScript
│   ├── components/                    # Componentes reutilizáveis
│   ├── pages/                         # Telas principais (Home, Contatos,
Emergência)
│   └── assets/                        # Ícones, imagens e sons
├── public/                            # Arquivos públicos
├── src-tauri/                         # Configuração e código backend (Tauri +
Rust)
│   ├── Cargo.toml                    # Configuração do Rust/Tauri
│   └── tauri.conf.json               # Configurações do build e nome do
executável
├── APLICATIVO_EXECUTAVEL/            # Pasta onde o .exe final é armazenado
└── GUIA_USUARIO/                     # Documentos e capturas de tela
```

6. Execução Local (Modo Desenvolvimento)

Para rodar o projeto com Tauri no modo de desenvolvimento (sem empacotar):

```
npm run tauri dev
```

Esse comando:

- Inicia o servidor local Vite;
- Executa o aplicativo em uma janela desktop nativa (sem navegador).

A saída do terminal mostrará logs do Tauri e do Vite, e o aplicativo será aberto automaticamente.

7. Build de Produção (Gerar Executável .exe)

Para gerar o executável do **CuidadoSenior**, siga os passos abaixo:

7.1. Verificar ambiente Rust

Confirme se o Rust está configurado corretamente:

```
rustc --version  
cargo --version
```

Ambos devem retornar um número de versão.

7.2. Gerar o build do aplicativo

No terminal, dentro da pasta do projeto, execute:

```
npm run tauri build
```

O processo irá:

- Compilar o front-end (React + TypeScript) usando o Vite;
- Gerar o backend nativo em Rust;
- Empacotar tudo em um executável .exe pronto para uso.

Após a conclusão, o resultado será gerado em:

```
src-tauri/target/release/
```

7.3. Mover o arquivo final

Renomeie o executável (se desejar) e mova para a pasta padrão:

APLICATIVO_EXECUTAVEL/CuidadoSenior.exe

O arquivo pode ser executado diretamente em qualquer computador Windows, sem necessidade de dependências adicionais.

8. Testes e Execução do Executável

1. Localize o arquivo:
2. APLICATIVO_EXECUTAVEL/CuidadoSenior.exe
3. Clique duas vezes para abrir o aplicativo.
4. Verifique se as telas principais estão acessíveis:
 - o **Tela Inicial** com botões de acesso;
 - o **Tela de Contatos** (cadastro e edição);
 - o **Botão de Ligação de Emergência** funcional;
 - o **Interface responsiva** e acessível.

9. Problemas Comuns (Troubleshooting)

Problema	Causa Provável	Solução
Erro ao rodar npm run tauri build	Falta de Rust instalado	Instale Rust e reinicie o terminal
Visual Studio Build Tools não encontrado	Dependência do compilador C++ ausente	Reinstale Build Tools com "Desktop Development with C++"
tauri.conf.json inválido	Erro de formatação no JSON	Revise as chaves e vírgulas do arquivo
App não abre após build	Antivírus bloqueando o executável	Adicione o app à lista de exceções

10. Contribuição para o Projeto

1. Faça um **fork** do repositório;
2. Crie uma nova branch para sua modificação:
3. git checkout -b minha-feature
4. Faça o commit das alterações:
5. git commit -m "Nova funcionalidade adicionada"
6. Envie para o seu fork:
7. git push origin minha-feature
8. Abra um **Pull Request** descrevendo suas alterações.

11. Suporte e Contato

Para dúvidas técnicas, erros de execução ou sugestões, entre em contato com a equipe de desenvolvimento:

- **Ellen Rodrigues Cardoso** — TIA: 10416041 —
10416041@mackenzista.com.br
- **Erik Dias Gomes da Silva** — TIA: 10416035 —
erikdgds.edgds@gmail.com
- **José Rene Martins Oliveira** — TIA: 10423660 —
renemartinsjf@gmail.com
- **Mateus Galvão de Melo Silva** — TIA: 10424662 —
mateusgalvaomelos@gmail.com
- **Yago Ben Homauad Martins** — TIA: 10424412 —
yagobenh@gmail.com

12. Considerações Finais

O **CuidadoSenior** é resultado da integração entre **tecnologias web modernas (React, TypeScript e Vite)** e o **framework Tauri**, que possibilita empacotamento nativo em **Rust** com excelente desempenho e baixo consumo de memória.

A presente documentação técnica fornece os passos completos para **configurar o ambiente, executar o aplicativo localmente e gerar o executável**, garantindo reprodutibilidade e clareza para novos colaboradores e avaliadores técnicos.