

PROMPT COMPLETO PARA GERAR O FRONTEND DO APLICATIVO (LLM)

⚠️ Instruções para a IA:

Crie o frontend completo de um aplicativo de chat baseado na interface do ChatGPT da OpenAI, adaptado para funcionar como Web App, Aplicativo Mobile (via PWA ou wrapper como Capacitor/React Native Web), e Desktop. A interface deve ser 100% responsiva e moderna, mantendo compatibilidade com:

- **Mobile** (smartphones e tablets),
- **Desktop** (navegadores e apps instaláveis),
- **Modo PWA** com manifest e service worker (base pronto),
- **Modo escuro/claro alternável**,
- **Design semelhante ao ChatGPT da OpenAI**.

🔧 Tecnologias obrigatórias:

- React + Tailwind CSS
- `react-markdown` para renderização de texto com Markdown
- `framer-motion` para animações
- `lucide-react` para ícones modernos
- Modularização de componentes (`/components`)

📁 Estrutura esperada de pastas:

bash

Copiar código

`src/`

```
|— App.jsx
|— main.jsx
|— index.css
|— assets/      # Ícones e imagens se necessário
|— components/
```

```
|
| |— Sidebar.jsx
| |— ChatArea.jsx
| |— MessageBubble.jsx
| |— InputArea.jsx
|— layouts/      # Layout base (opcional)
|— services/     # APIs futuras (placeholder)
|— hooks/        # useDarkMode (opcional)
|— utils/        # formatText.js (opcional)
```

Funcionalidades obrigatórias:

1. Sidebar lateral fixa (visível apenas em desktop) com:

- Botão de nova conversa
- Lista de conversas fictícias (mock)
- Botões: Perfil, Ajuda, Configurações
- Botão para alternar modo escuro/claro

2. Área de mensagens (ChatArea):

- Título fixo superior
- Lista de mensagens com bolhas alternadas (usuário x IA)
- Animação de entrada suave (`framer-motion`)
- Markdown ativado nas mensagens (com `react-markdown`)

3. Input de mensagem (InputArea):

- Textarea expansível com suporte a `Enter` para enviar, `Shift+Enter` para nova linha
- Ícone de envio
- Ícone de upload (sem funcionalidade real ainda)

4. Modo escuro/claro:

- Usa classe `dark` do Tailwind

- Armazenamento em `useState` (ou opcionalmente `localStorage`)
- Layout adaptado para ambos os modos

5. **100% responsivo:**

- Mobile-first
- Sidebar oculta em telas menores
- Input e mensagens com bom espaçamento e adaptabilidade

6. **PWA** (base pronta):

- Criar `manifest.json`
- Adicionar `serviceWorker.js` (pode ser vazio ou pronto para expansão futura)

O que não deve ser incluído agora:

- Backend/API real (apenas simulação com `setTimeout`)
- Sistema de login/autenticação
- Armazenamento de conversas real
- Upload funcional

Instalar automaticamente estas dependências no projeto gerado:

bash

Copiar código

```
npm install react-markdown framer-motion lucide-react
npm install -D tailwindcss postcss autoprefixer
npx tailwindcss init -p
```

Observações finais:

- Use boas práticas de estruturação e responsividade.
 - Comentários curtos no código são bem-vindos para explicação dos blocos.
 - Arquivo `index.css` deve conter os `@tailwind base;`, `components`, `utilities` e `font-family` personalizada.
 - Pode simular mensagens da IA com `setTimeout` após o envio de mensagem.
-

