

✓ RELATÓRIO CONSOLIDADO - IMPLEMENTAÇÃO DE CHAT CONVERSACIONAL

Data: 15 de novembro de 2025 **Versão Testada:** v3.6.1

Prezado,

Conforme solicitado, analisei a funcionalidade de continuação de conversa e preparei um relatório consolidado com recomendações cirúrgicas para implementar o chat conversacional.

🎯 VEREDITO: FUNCIONALIDADE INEXISTENTE, MAS IMPLEMENTAÇÃO SIMPLES

O sistema atual **não possui** a funcionalidade de chat conversacional. A interação é "one-shot" (uma pergunta, uma resposta). No entanto, a arquitetura atual (React, tRPC, SSE) permite uma implementação **simples e cirúrgica**.

🚀 RECOMENDAÇÕES CIRÚRGICAS (MUDANÇAS MÍNIMAS)

1. Adicionar Campo de Input no Frontend:

- **Arquivo:** `client/src/components/StreamingPromptExecutor.tsx`
- **Mudança:** Adicionar um `<textarea>` e um botão "Enviar" abaixo da resposta da IA.

2. Manter Histórico da Conversa no Estado do Componente:

- **Arquivo:** `client/src/components/StreamingPromptExecutor.tsx`
- **Mudança:** Usar `useState` para manter um array de mensagens (ex: `const [messages, setMessages] = useState([])`).

3. Enviar Histórico da Conversa para o Backend:

- **Arquivo:** `client/src/hooks/useStreamingPrompt.ts`
- **Mudança:** Modificar a chamada da API para enviar o histórico de mensagens junto com a nova mensagem do usuário.

4. Adaptar Backend para Receber Histórico:

- **Arquivo:** `server/routers/prompts.ts`
- **Mudança:** Modificar a rota `execute/stream` para receber um array de mensagens (histórico) e concatená-lo ao prompt antes de enviar para a IA.

📊 IMPACTO ESTIMADO

Métrica	Estimativa
Tempo de Desenvolvimento	2-4 horas
Linhas de Código	~50-100 linhas
Arquivos Modificados	3-4 arquivos
Risco de Regressão	Baixo

CONCLUSÃO

A implementação do chat conversacional é uma **evolução natural e de baixo custo** para o sistema. As recomendações acima permitem uma implementação rápida e com mínimo impacto no código existente.

Aguardando a implementação para a validação final!