O Desafio: "Chatbot de Atendimento Simulado"

Queremos que você construa um **protótipo fullstack** (Backend + Frontend) de um sistema de chat. O **objetivo** é simular uma tarefa real de desenvolvimento, avaliando sua capacidade de implementar novas funcionalidades e criar uma solução completa.

🚀 Tecnologias Obrigatórias

Backend: Python 3+ com Django (ou Django REST Framework).

Frontend: React.

Banco de Dados: Pode utilizar o SQLite (padrão do Django).

Controle de Versão: Git.

📋 Requisitos Funcionais

1. **O "Login Mockado" (Simulado)**
   - NÃO precisa criar um sistema de login e senha com autenticação complexa.
   - No frontend (React), crie uma forma simples (ex: botões ou um dropdown) para o usuário "assumir" um de dois perfis: "Usuário A" ou "Usuário B".
   - O estado de qual usuário está "ativo" deve ser gerenciado no React.
2. **A Tela de Chat**
   - O usuário (A ou B) deve ter uma interface de chat onde pode digitar e enviar uma mensagem.
   - Ao clicar em "Enviar", o frontend deve enviar a mensagem e o identificador do usuário ativo (ex: "A" ou "B") para a API no backend.

- A API (Django) deve salvar essa mensagem no banco de dados, vinculando-a corretamente ao usuário que a enviou.
- A API deve então retornar uma resposta mockada (simulada), como: "Obrigado por seu contato. Em breve responderemos." (diferentes para cada usuário)
- A tela do chat deve exibir tanto a mensagem enviada pelo usuário quanto a resposta recebida do backend.

3. A Tela de Histórico

- Crie uma segunda página/rota no React (ex: /historico).
- Esta página deve buscar e exibir apenas o histórico de mensagens (perguntas e respostas) do usuário que está atualmente selecionado (A ou B).
- Se o "Usuário A" estiver ativo, a página deve mostrar apenas as mensagens de A. Se o usuário trocar para "Usuário B" no frontend, esta página deve atualizar e mostrar apenas as mensagens de B.

📦 **O que você deve entregar**

- Um link para um repositório público Git (GitHub, GitLab, etc.) contendo seu projeto.
- Um arquivo README.md claro, contendo:
  - Instruções simples de como configurar e rodar o projeto localmente (instalação de dependências, comandos para rodar o backend e o frontend).
  - Uma breve explicação de suas decisões técnicas (ex: como você decidiu estruturar seus models no Django ou gerenciar o estado no React).

Essas informações devem ser enviadas para o e-mail: marcelosilva@4blue.com.br

💡 **O que será Avaliado**

- Qualidade do Código: Organização, clareza e boas práticas em Python/Django e React.
- Lógica de Negócio: Sua capacidade de implementar a filtragem de dados por usuário (o requisito principal do histórico).
- Modelagem de Dados: Como você estruturou seus models no Django.
- Funcionalidade: O aplicativo cumpre os requisitos descritos?
- Documentação: A clareza do seu README.md.