

## **Descrição do Teste:**

Neste teste, você desenvolverá uma aplicação em React que se comunica com um servidor WebSocket. O objetivo é criar um chat em tempo real, onde múltiplos usuários podem se conectar e trocar mensagens.

## **Requisitos:**

### **1. Interface do Usuário:**

- Uma tela de login onde o usuário pode digitar um nome de usuário.
- Uma tela de chat onde o usuário pode ver as mensagens enviadas e recebidas, e um campo de input para enviar novas mensagens.
- As mensagens devem ser exibidas com o nome do usuário que enviou, a mensagem e o horário de envio.

### **2. Funcionalidades:**

- Implementar a conexão com um servidor WebSocket.
- Enviar mensagens para o servidor WebSocket.
- Receber mensagens do servidor WebSocket e exibir na tela de chat em tempo real.
- Notificar quando um novo usuário entra no chat ou quando um usuário sai.

### **3. Servidor WebSocket:**

- Você pode utilizar um servidor WebSocket simples com Node.js e a biblioteca ws para testes locais, ou pode utilizar um serviço WebSocket existente.

### **4. Estilização:**

- A estilização deve ser mínima, mas funcional. Utilize CSS ou bibliotecas como Styled Components ou Material-UI.

### **5. Código e Organização:**

- Utilize as melhores práticas de desenvolvimento em React.
- Organize seu código de forma clara e modular.
- Inclua comentários explicativos onde necessário.
- Utilize hooks do React para gerenciar estado e efeitos colaterais.

## **Critérios de Avaliação:**

### **1. Funcionalidade:**

- A aplicação funciona conforme especificado.

- Consegue enviar e receber mensagens em tempo real.

## **2. Qualidade do Código:**

- Código bem organizado e comentado.
- Uso apropriado dos hooks do React.
- Tratamento de erros e validações.

## **3. Interface do Usuário:**

- Interface limpa e intuitiva.
- Experiência do usuário suave e sem bugs aparentes.

## **4. Documentação:**

- README detalhado explicando como configurar e rodar a aplicação.
- Comentários no código para explicar as partes principais.

### **Instruções:**

#### **1. Setup Inicial:**

- Crie uma aplicação React utilizando create-react-app ou outra ferramenta de sua escolha.
- Configure o projeto para utilizar WebSockets.

#### **2. Implementação:**

- Crie a tela de login e a tela de chat conforme especificado nos requisitos.
- Implemente a comunicação com o servidor WebSocket.
- Teste a aplicação localmente.

#### **3. Entrega:**

- Suba o código em um repositório no GitHub ou outra plataforma de sua escolha.
- Inclua instruções detalhadas sobre como rodar a aplicação no arquivo README.
- Envie o link do repositório para avaliação.