

## Projeto de Redes de Computadores

Este projeto tem o objetivo da criação de um *chat* seguindo o modelo cliente e servidor com a utilização de sockets para o envio e recebimento de mensagens, desenvolvido em Node.js.

### Principais Funcionalidades

A aplicação segue um modelo Cliente/Servidor, onde o seu objetivo é enviar mensagem para uma sala de conversa de modo que todos os usuários daquela sala recebam a mensagem. Todo envio de mensagem pelo lado do cliente é enviado para o *back-end* da aplicação onde será emitido um socket chamado *message* com o objeto da mensagem contendo o nome do usuário, hora/data e mensagem. **As funcionalidades da aplicação são:**

- É possível ingressar em salas existentes, como a Sala Pública (*default*);
- Criar novas salas personalizadas para poder utilizar o *chat* (as salas ficam salvas em um arquivo .txt no *back-end* da aplicação);
- Durante a conversação é possível o envio de arquivo de imagens (única ou múltipla), basta clicar no ícone de clipe e selecionar os arquivos;
- É possível programar para que uma determinada mensagem seja apagada dado um tempo em segundos. Para usar essa função basta digitar a mensagem na caixa de texto e clicar no botão com ícone de relógio, onde será solicitado o tempo em segundos para que a mensagem expire. Após esse tempo passar, todos os que receberam a mensagem não irão vê-la mais e será exibida a frase: “Mensagem Expirada”.

### O que poderia ser implementado a mais

Por algumas dificuldades durante a condução do projeto, onde foi gasto muito tempo em algumas funcionalidades citadas acima, acabou não dando tempo de implementar:

- Criação de salas com senha;
- Possibilidade de enviar arquivos de vídeo ou pdf;
- Programar mensagens que expiram para o envio de imagens;

### Dificuldades Encontradas

Houve dificuldade para usar as funções de socket.io, onde foi necessário ler a documentação e ver alguns vídeo para saber como implementa-la na linguagem adotada. Houve um gasto de tempo considerável a respeito de como enviar arquivo de imagens (única e múltipla) de modo que todos os usuários recebessem o que foi enviado. Outra funcionalidade sendo considera a mais dificultosa, foi fazer a mensagem expirar após o tempo dado pelo usuário, de modo que ela expirasse para todos os usuários, pois como as mensagens não eram salvas, era necessário uma referência de quem enviou e uma identificação da mesma. Para isso foi criado no lado do servidor um socket chamado *UpdateMessage* que faria a lógica para isso e então fluiu o desejado.