

Aluno: Denilson Bulhões da Rosa Silva

Aluno: José Arthur Lopes Sabino

### **Principais funcionalidades:**

A aplicação gerencia um sistema de bate papo. Ao entrar e escolher qual sala desejar participar poderá visualizar mensagens enviadas por outros membros do chat, assim como, enviar mensagens que serão visualizadas por todos clientes que também façam parte da sala.

### **O que poderia ser implementado a mais:**

Implementar transferências de mensagens de áudio e de arquivo, deixando o chat ainda mais completo. Além disso poderia ser feito uma interface gráfica, mas como nunca trabalhamos com isso, nos focamos nas funcionalidades de redes.

### **Principais dificuldades:**

A dificuldade principal foi separar as salas de modo que mensagens enviadas não acabassem indo para sala errada. Algo que aconteceu muito ao decorrer do desenvolvimento do sistema foi mensagens enviadas por exemplo na “Sala 1” sendo encaminhadas para todas as outras salas. Uma outra dificuldade encontrada foi como saber que um cliente se desconectou da sala sem aviso prévio, pois o cliente pode simplesmente fechar a aplicação e deixar-nos desperdiçando recurso naquela thread. Mas tratamos isso com um try, except do python, e notificamos a todos da sala que o cliente se desconectava.

### **Aplicação dos conceitos de redes:**

Utilizamos web-socket juntamente a operações multi-thread, para podermos ter vários clientes se conectando ao nosso servidor. Primeiramente nosso servidor abre uma thread para cada cliente, e registra a sala em que ele está conectado, fazendo com que não sejam desperdiçados recursos de rede enviando mensagens para salas que não deveriam processá-las/recebê-las. Utilizamos os conceitos de bind e listen do lado do servidor, para ficar aguardando conexões, e quando recebemos uma conexão direcionamos para as checagens, verificamos o nome e sala que o cliente nos informa, e se forem dados válidos abrimos a thread e colocamos o cliente na sala desejada, informando-o através de uma mensagem. Após isso ele pode usar o chat de forma livre, utilizando os conceitos de send e receive, tanto do lado do cliente (recebe as mensagens dos outros clientes através do servidor e envia mensagens para o servidor para que ele redirecione para os outros clientes) quanto do lado do servidor (recebe as mensagens do cliente e envia a de outros clientes para ele).