

Especificação do Trabalho II – ELC1018 Sistemas Distribuídos
Programação de uma aplicação Salas de Chat utilizando Java RMI
Trabalho DEVE ser realizado em dupla

O trabalho consiste em implementar uma aplicação **Salas de Chat** utilizando Java RMI.

A aplicação distribuída Salas de Chat deve ter um servidor central (classe `ServerChat`) e permitir diversos clientes (classe `UserChat`) e diversas salas (classe `RoomChat`).

Obedeça rigorosamente a esta especificação. A funcionalidade do seu trabalho será testada junto a clientes e ou servidores desenvolvidos por outros grupos, escolhidos aleatoriamente.

Requisitos Funcionais Absolutos:

RFA1: As classes do servidor central, dos clientes e das salas devem implementar as interfaces `IServerChat`, `IUserChat` e `IRoomChat`, respectivamente, conforme abaixo especificado.

```
public interface IServerChat extends java.rmi.Remote {
    public ArrayList<String> getRooms();
    public void createRoom(String roomName);
}
public interface IUserChat extends java.rmi.Remote {
    public void deliverMsg(String senderName, String msg);
}
public interface IRoomChat extends java.rmi.Remote {
    public void sendMsg(String usrName, String msg);
    public void joinRoom(String userName, IUserChat user);
    public void leaveRoom(String usrName);
    public String getRoomName();
    public void closeRoom();
}
```

RFA2: O servidor deve manter uma lista de salas, que deve ser declarada como `private Map<String, IRoomChat> roomList`. O servidor deve garantir que não deve haver duas salas com o mesmo nome.

RFA3: No servidor não deve haver limite de salas, tampouco de usuários por sala.

RFA4: Cada sala (classe `RoomChat`) deve manter uma lista de usuários (`userList`), que deve ser declarada como `private Map<String, IUserChat> userList`.

RFA5: No início, todo cliente, identificado pelo seu nome (`usrName`), deve contatar o servidor e solicitar a lista de nomes de salas via método remoto `getRooms()`. A lista de nomes de salas deve ser exibida na interface do usuário (GUI), para permitir a escolha da sala.

RFA6: Sempre que um usuário desejar entrar numa sala já existente ele deve solicitar a referência ao objeto remoto ao RMI Registry usando o nome da sala e, após conhecer o objeto, deve invocar o método remoto `joinRoom()` da respectiva sala, passando seu nome e sua referência remota.

RFA7: Caso o usuário não encontre no servidor a sala desejada ele deve poder solicitar a criação de uma nova sala. Isto deve ser feito através da invocação ao método remoto `createRoom(String roomName)` do servidor. A vinculação do usuário a esta sala não deve ser automática. Ele deve solicitar a entrada invocando o método remoto `joinRoom()` da sala.

RFA8: Após pertencer a uma sala, o usuário deve enviar mensagens de texto à sala através da invocação ao método remoto `sendMsg(String usrName, String msg)` da sala.

RFA9: Para receber mensagens, o processo do usuário deve implementar um método remoto `deliverMsg(String senderName, String msg)`.

RFA10: O controlador da sala é quem deve controlar o envio das mensagens aos membros da sala.

RFA11: Os usuários devem sair da sala invocando o método remoto `leaveRoom(String userName)` da sala.

RFA12: Uma sala só deve poder ser fechada pelo servidor. O servidor deve fechar a sala invocando o método remoto `closeRoom()` do controlador de sala. Caso haja usuários na sala, antes de ser finalizado o controlador da sala deve enviar uma mensagem *“Sala fechada pelo servidor.”* aos usuários.

RFA13: Após fechar a sala o servidor deve eliminar a sala da lista de salas. Cada usuário deve fazer o mesmo ao receber a mensagem *“Sala fechada pelo servidor.”* do controlador.

RFA14: A formatação da GUI para o usuário e servidor é de livre escolha, mas deve contar no mínimo com um quadro para visualização das mensagens, com a possibilidade de seleção da sala pelo usuário e servidor, além de conter botões apropriados para os principais comandos (SEND, CLOSE, JOIN e LEAVE).

RFA15: O servidor deve ser registrado no registro de RMI (`rmiregistry`) com o nome “Servidor” e usar a porta “2020” para escutar clientes. O registro deve executar na máquina do servidor.