Descrição do sistema:

Será desenvolvido um sistema de chat que suporte múltiplos usuários, onde cada um poderá enviar e receber mensagens de texto simultaneamente entre eles.

Requisitos funcionais do Sistema:

RF-01: O Sistema deve manter usuários;

RF-02: O Sistema deve manter servidor;

RF-03: O Sistema deve manter conexões entre usuário e Servidor;

RF-04: O Sistema deve manter conexões entre usuários;

RF-05: O Sistema deve manter a troca de mensagens de texto entre usuários;

Regras de negócio:

RN-01: A tela de cadastro deve conter os campos de texto: Nome, porta e endereço do servidor;

RN-02: A tela de menu principal do chat deve ter: Uma lista de usuários conectados, nome do usuário atual e um botão para iniciar uma conversa;

RN-03: Se um usuário encerrar a aplicação, o servidor deverá removê-lo da lista de usuários conectados;

RN-04: Se um usuário fechar a tela de chat, o sistema deve encerrar a conversa nos dois usuários;

Diagrama de classes:

GerenciadorMensagens

- dados: String

converteDados(Cliente cliente):String dados EnviaMensagem(String mensagem):void RecebeMensagem():String mensagem

Cliente

- nome: String - Ip: String

- ServerIp: String

- Porta: int

- clientSocket: Socket

- cServerSocket: ServerSocket

- UsuáriosConversa: List<String>

conexaoServ(String ip):void conexaoCliente(String ip, int porta):void

listarUsers(): List<String> encerrar(Socket s):void sinalPresenca():void

GerenciadorConexao

- dados: String - socket: Socket - server: Servidor

run(): void

Servidor

- Ip: String - Porta: int

- serverSocket: ServerSocket

- Usuários: List<String>

iniciarServidor(): void addUsuario(String dados):void

removerUsuario(String

dados):void

returnUsuarios(): String