

Relatório - Projeto de Redes de Computadores

Equipe: Lucas Albuquerque Lisboa (18110495) e José Rubens da Silva Brito (18110471)

Professor: Leandro Melo de Sales

Tecnologias Utilizadas:

- JDK 11;
- Visual Code Studio 1.52.1;

Descrição do Projeto: Implementação de um jogo com temática da disciplina de Redes de Computadores, em que o usuário possui até 3 possibilidades de erro para tentar acertar a palavra correspondente. Esta implementação consistiu na utilização de threads, de tal forma que possibilitou a ocorrência simultânea de várias partidas.

Protocolos:

<i>Descrição</i>	<i>Cliente</i>	<i>Servidor</i>
Iniciar Conexão	<pre>Socket socketNewClient = new Socket("localhost", 1234);</pre>	<pre>Socket connectionSocket = socketServer.accept();</pre>
Enviar Letra	<pre>sentence = inFromUser.readLine() ; outputServer.writeByt es(sentence + '\n');</pre>	<pre>readCliente = inFromGame.readLine(); readCliente = readCliente.toLowerCas e();</pre>
Retornar Estado Atual da Palavra	<pre>System.out.println(in FromServer.readLine());</pre>	<pre>for (int i = 0 ; i < gameSentence.length(); i++){ outToGame.writeUTF(let ters.get(i) + " "); } outToGame.writeByte('\ n');</pre>
Retornar Resultado	<pre>msg = inFromServer.readLine (); System.out.println(ms g);</pre>	<pre>String msg; if(count_lives < 3) { msg = "Parabens! Voce ganhou!"; } else{</pre>

	<pre> if (msg.equals("Parabéns! Você ganhou!") msg.equals("Que pena! Você perdeu!")) { break; } </pre>	<pre> outToGame.writeBytes("Palavra era:"+gameSentence+'\n'); msg = "Que pena! Você perdeu!"; outToGame.writeBytes(msg + '\n'); </pre>
Encerrar Conexão	<pre> inFromServer.close(); outputServer.close(); inFromUser.close(); socketNewClient.close(); </pre>	<pre> inFromGame.close(); outToGame.close(); </pre>

Setup:

- Possuir o pacote JDK 11 instalado na Máquina;
- Realizar o download da aplicação;
 - `git clone https://github.com/lucasalisboa/Projeto_Redes---Jogo_da_Forca.git`
- Após realizar o download, rodar primeiro a classe `Server.java` e logo em seguida rodar a classe `Cliente.java`, pode executar mais de uma vez simultaneamente a classe `Ciente.java`;
- Iniciar o jogo;

Implementações Futuras:

- Interface Gráfica;
- Ampliar banco de palavras selecionadas, de forma inteligente.

Dificuldades na Implementação:

- Organizar dentro de um laço o envio de informações, a ponto de ficar equivalente ao lado do receptor da mensagem (que deverá ter um leitor equivalente para recebê-la);