**UNIVERSIDADE DO ESTADO DE SANTA CATARINA**

CURSO DE ENGENHARIA DE SOFTWARE

ANA CRISTINA VASCONCELLOS REINERT

MATEUS COELHO NOSSE

**TRABALHO FINAL DESENVOLVIMENTO DE SISTEMAS PARALELOS E DISTRIBUÍDOS**

IBIRAMA

SETEMBRO/2022

ANA CRISTINA VASCONCELLOS REINERT

MATEUS COELHO NOSSE

**TRABALHO FINAL DESENVOLVIMENTO DE SISTEMAS PARALELOS E DISTRIBUÍDOS**

Trabalho apresentado no curso de graduação da Universidade do Estado de Santa Catarina.

Orientador: Fernando dos Santos

IBIRAMA

SETEMBRO/2022

**SUMÁRIO**

[1 INTRODUÇÃO 4](#_Toc112789193)

[2 REQUISITOS DO SISTEMA 5](#_Toc112789194)

[2.1 REQUISITOS FUNCIONAIS 5](#_Toc112789195)

[3 MENSAGENS 6](#_Toc112789196)

1 INTRODUÇÃO

A proposta para o desenvolvimento desse trabalho trata-se de um serviço de chat para ambientes desktops, onde os usuários poderão se conectar a um canal de comunicação para trocar mensagens entre si. Tendo a ideia de interface inspirada no chat do serviço de mensagens “Omegle”, onde as mensagens trocadas entre os usuários ficam dispostas uma abaixo da outra no mesmo lado da coluna de conversa.

Como o sistema será desenvolvido na linguagem de programação Java, utilizaremos a biblioteca Swing para a criação da interface gráfica. Será necessário que o usuário informe o endereço ip e portas deseja conectar para que então, após confirmação de conexão estabelecida, seja iniciado o chat. O chat funcionará de forma nominada por usernames, sendo estes que o próprio usuário poderá definir e será mostrado para os participantes do chat como forma de identificação desse usuário na rede.

2 REQUISITOS DO SISTEMA

2.1 REQUISITOS FUNCIONAIS

|  |  |
| --- | --- |
| Identificação | Descrição |
| RF01 | O sistema deve permitir que o usuário conecte a um servidor de chat. |
| RF02 | O sistema deve permitir que o usuário desconecte de um servidor de chat. |
| RF03 | O sistema deve informar ao usuário o resultado da busca pelo servidor. |
| RF04 | O sistema deve permitir que o usuário altere seu nome de usuário. |
| RF05 | O sistema deve permitir que o usuário envie mensagens para o servidor. |
| RF06 | O sistema deve limpar o campo de digitação após o envio bem-sucedido da mensagem. |
| RF07 | O sistema deve informar ao usuário caso haja algum erro no envio da mensagem. |
| RF08 | O sistema deve exibir uma lista de todas as mensagens existentes no servidor. |
| RF09 | O sistema deve atualizar a lista de mensagens do servidor após um novo envio bem-sucedido. |
| RF10 | O sistema deve mostrar uma lista de usuários que estão conectados no chat. |
| RF11 | O sistema deve mostrar uma lista de usuários que não estão conectados no chat. |
| RF12 | O sistema deve permitir o usuário atualizar dados de perfil. |

3 MENSAGENS

3.1 ENTRADA NO CHAT

|  |  |  |
| --- | --- | --- |
| **Conteúdo da Mensagem** | **Campos**  Operação  nome | **Conteúdo**  “INSERT”  Nome de usuário |
| **Descrição** | Conectar ao chat | |
| **Retorno** | Não há | |

3.2 DESCONECTAR DO CHAT

|  |  |  |
| --- | --- | --- |
| **Conteúdo da Mensagem** | **Campos**  Operação | **Conteúdo**  “DELETE” |
| **Descrição** | Desconectar do chat | |
| **Retorno** | Não há | |

3.3 ENVIAR UMA MENSAGEM PARA O CHAT

|  |  |  |
| --- | --- | --- |
| **Conteúdo da Mensagem** | **Campos**  Operação  mensagem  horário | **Conteúdo**  “INSERT”  A mensagem enviada  Horário de envio |
| **Descrição** | Envio de uma mensagem | |
| **Retorno** | Não há | |

3.4 LISTAR TODOS OS USUÁRIOS NO CHAT

|  |  |  |
| --- | --- | --- |
| **Conteúdo da Mensagem** | **Campos**  Operação | **Conteúdo**  “LIST” |
| **Descrição** | Busca todas as pessoas do chat e lista | |
| **Retorno** | A String de retorno é formada pela quantidade de pessoas no chat separado pelo status dele no chat, com quebra de linha entre eles. Exemplos:  Ana Cristina – Online  Mateus Coelho – Online  Julio Cesar - Offline | |

3.5 LISTAR TODAS AS MENSAGENS

|  |  |  |
| --- | --- | --- |
| **Conteúdo da Mensagem** | **Campos**  Operação | **Conteúdo**  “LIST” |
| **Descrição** | Busca todas as mensagens do chat e lista | |
| **Retorno** | O retorno da mensagem é um String composta pelos dados que essas possuem, com quebra de linha entres essas informações.  Exemplo:  Ana Cristina;Oi;17:10  Mateus Nosse;Oi, Tudo bem?;17:11  Julio Cesar;Tudo certo;17:13  Se não existirem mensagens cadastradas, o retorno seria o valor 0. | |