**Texto

Descrição gerada automaticamente com confiança média**

**UNIVERSIDADE ESTATUAL DE SANTA CATARINA**

**CEAVI - CENTRO DE EDUCAÇÃO SUPERIOR DO ALTO VALE DO ITAJAÍ**

**DEPARTAMENTO DE ENGENHARIA DE SOFTWARE**

**Afonso Uéslei Böing**

**Lucas Ramthum Vegini**

**Trabalho 4 – Chat multithread**

**Ibirama**

O objetivo do trabalho será desenvolver um chat multithread utilizando socket, para isso teremos um servidor central que irá trocar mensagens entre os clientes do sistema.

Requisitos funcionais do sistema.

1. O sistema não deverá permitir dois clientes com o mesmo nome de usuário.
2. O sistema deverá permitir que o usuário envie mensagem para todos os usuários.
3. O sistema deverá permitir que os clientes troquem mensagens exclusivas entre eles.
4. O sistema deverá informar o cliente quando um novo usuário entrar no chat.
5. O sistema deverá informar quando alguém deixar o chat.
6. O sistema deverá permitir que o usuário saia do chat.
7. O sistema deverá permitir que o usuário que saiu do chat possa voltar pra ele.

**Classe cliente**

**CLIENTE ENTRANDO NO SERVIDOR**

|  |  |
| --- | --- |
| **Conteúdo da mensagem** | ‘ValidaNome;{NOME} |
| **Descrição** | Cliente tenta entrar no servidor e para isso primeiramente é necessário validar o nome, caso o nome seja valido a thread do cliente começa a rodar. |
| **Retorno** | Caso o usuário não mande um campo preenchido: “error;Para executar essa operação é necessário preencher todos os campos de texto”  Caso o nome já esteja sendo usado: “error;Esse nome já está sendo usado no sistema”;  Caso sucesso: “success; Bem-vindo ao chat ” |

**ENVIAR MENSAGEM PARA TODOS**

|  |  |
| --- | --- |
| **Conteúdo da mensagem** | ‘todos;{Mensagem} |
| **Descrição** | Cliente envia mensagem para todos do chat. |
| **Retorno** | Caso o usuário não mande um campo preenchido: “error;Para executar essa operação é necessário preencher todos os campos de texto”  Caso sucesso: “success” |

**ENVIAR MENSAGEM PARA ALGÉM EM ESPECÍFICO**

|  |  |
| --- | --- |
| **Conteúdo da mensagem** | ‘individual;{Nome do usuário};{Nome do usuário que receberá a mensagem};{Mensagem} |
| **Descrição** | Cliente envia uma mensagem para alguém específico do chat. |
| **Retorno** | Caso o usuário não mande um campo preenchido: “error;Para executar essa operação é necessário preencher todos os campos de texto”  Caso sucesso: “success” |

**SAIR DO CHAT**

|  |  |
| --- | --- |
| **Conteúdo da mensagem** | ‘sair;{Nome de usuário} |
| **Descrição** | Cliente envia sairá do chat; |
| **Retorno** | Caso o usuário não mande um campo preenchido: “error;Para executar essa operação é necessário preencher todos os campos de texto”  Caso sucesso: “success” |

**Classe servidor**

**SERVIDOR ENVIA MENSAGEM PARA TODOS**

|  |  |
| --- | --- |
| **Conteúdo da mensagem** | Caso seja o usuário que enviou a mensagem: ‘mensagem; você enviou para todos {mensagem}  Caso seja os outros usuários: ‘mensagem; {Nome usuário} enviou para todos {mensagem} |
| **Descrição** | Servidor envia mensagens para todos os usuário. |
| **Retorno** | Servidor não espera um retorno; |

**SERVIDOR ENVIA MENSAGEM PARA ALGUÉM ESPECÍFICO**

|  |  |
| --- | --- |
| **Conteúdo da mensagem** | Caso seja o usuário que enviou a mensagem: ‘mensagem; você enviou para: {Nome de usuário}{mensagem}  Caso seja usuário que receberá a mensagem: ‘mensagem; {Nome do usuário que enviou} enviou para você: {mensagem} |
| **Descrição** | Servidor envia mensagens para usuário de destino. |
| **Retorno** | Servidor não espera um retorno; |

**SERVIDOR COMUNIDA OS CLIENTES QUE ALGUÉM DEIXOU O CHAT**

|  |  |
| --- | --- |
| **Conteúdo da mensagem** | “saiu;{Nome de usuário} |
| **Descrição** | Servidor envia para todos os usuários ativos que alguém deixou o chat. |
| **Retorno** | Servidor não espera um retorno; |