

Projeto A3 Sistemas Distribuídos

Integrantes:

Lucca Bugatti Vasconcelos

RA: 125111345841

Matheus de Brito Milani

125111355174

Matheus Schiehl

Ra 125111352669

Introdução

Criamos este projeto com base no pdf da A3 de sistemas distribuídos, implementamos um sistema de bate papo entre usuários em diferentes máquinas utilizando programação em Java Orientado a Objetos com Sockets.

Manual de Usabilidade

Todo o nosso código é auto explicativo e possui diversos comentários explicando a funcionalidade do sistema.

O primeiro passo para utilizar o nosso sistema é selecionar o RUN do arquivo Servidor.java isto iniciará o nosso servidor que deve encaminhar uma mensagem de “Servidor funcionando.”

SAÍDA TERMINAL PROBLEMAS CONSOLE DE DEBURAÇÃO

```
PS C:\Users\lbuga\Downloads\Chat em java> & 'C:\Users\lbuga\AppData\Local\Programs\AdoptOpenJDK\jdk-11.0.10.9-hotspot\bin\java.exe' -cp 'C:\Users\lbuga\AppData\Roaming\Code\User\workspaceStorage\e15ddfa104328711e51ca8c2afaa875a\redhat.java\src\main\java\com\lbuga\chat\Server\Server.java' -Djava.class.path='C:\Users\lbuga\AppData\Roaming\Code\User\workspaceStorage\e15ddfa104328711e51ca8c2afaa875a\redhat.java\src\main\java\com\lbuga\chat\Server\Server.java' 'Servidor'
```

Servidor funcionando

Assim que o RUN do arquivo Servidor.java for concluído, o segundo passo será iniciar o RUN do arquivo Usuario.java, o terminal retornará a opção de inserir um nome para o usuário.

```
Experimente a nova plataforma cruzada PowerShell https://aka.ms/pscore6
```

```
PS C:\Users\lbuga\Downloads\Chat em java> & 'C:\Users\lbuga\AppData\Local\hotspot\bin\java.exe' '-cp' 'C:\Users\lbuga\AppData\Roaming\Code\User\workspace\afaa875a\redhat-java\jdt_ws\Chat em java_986ca42b\bin' 'Usuario'
```

```
Digite seu nome: 
```

Assim que o nome for introduzido, o servidor encaminhará a mensagem de “Novo usuário foi conectado.” Junto com a mensagem “Nome do usuário entrou no chat.” para os outros usuários que já estiverem conectados no servidor.

```
SAÍDA  TERMINAL  PROBLEMAS  CONSOLE DE DEPURAÇÃO

PS C:\Users\lbuga\Downloads\Chat em java> & 'C:\Users\lbuga\AppData\Local\Microsoft\Windows\apps\hotspot\bin\java.exe' '-cp' 'C:\Users\lbuga\AppData\Roaming\Code\User\workspace\afaa875a\redhat.java\jdt_ws\Chat em java_986ca42b\bin' 'Servidor'
Servidor funcionando
19:31 Um novo usuário foi conectado!
```

```
SAÍDA  TERMINAL  PROBLEMAS  CONSOLE DE DEPURAÇÃO

PS C:\Users\lbuga\Downloads\Chat em java> & 'C:\Users\lbuga\AppData\Local\Microsoft\Windows\apps\hotspot\bin\java.exe' '-cp' 'C:\Users\lbuga\AppData\Roaming\Code\User\workspace\afaa875a\redhat.java\jdt_ws\Chat em java_986ca42b\bin' 'Usuario'
Digite seu nome: Matheus
19:32 SERVIDOR: Lucca entrou no chat.
```

Dentro do bate papo, toda a sua funcionalidade segue os padrões de plataformas de comunicação.

```
SAÍDA  TERMINAL  PROBLEMAS  CONSOLE DE DEPURAÇÃO

afaa875a\redhat.java\jdt_ws\Chat em java_986ca42b\bin' 'Usuario'
Digite seu nome: Matheus
19:32 SERVIDOR: Lucca entrou no chat.
Oieeeee ← Minha mensagem
19:32 Lucca: Tudo bem?
```

Funcionalidades:

-Chat: O programa é uma plataforma de conversa que funciona por terminal e suporta vários usuários ao mesmo tempo

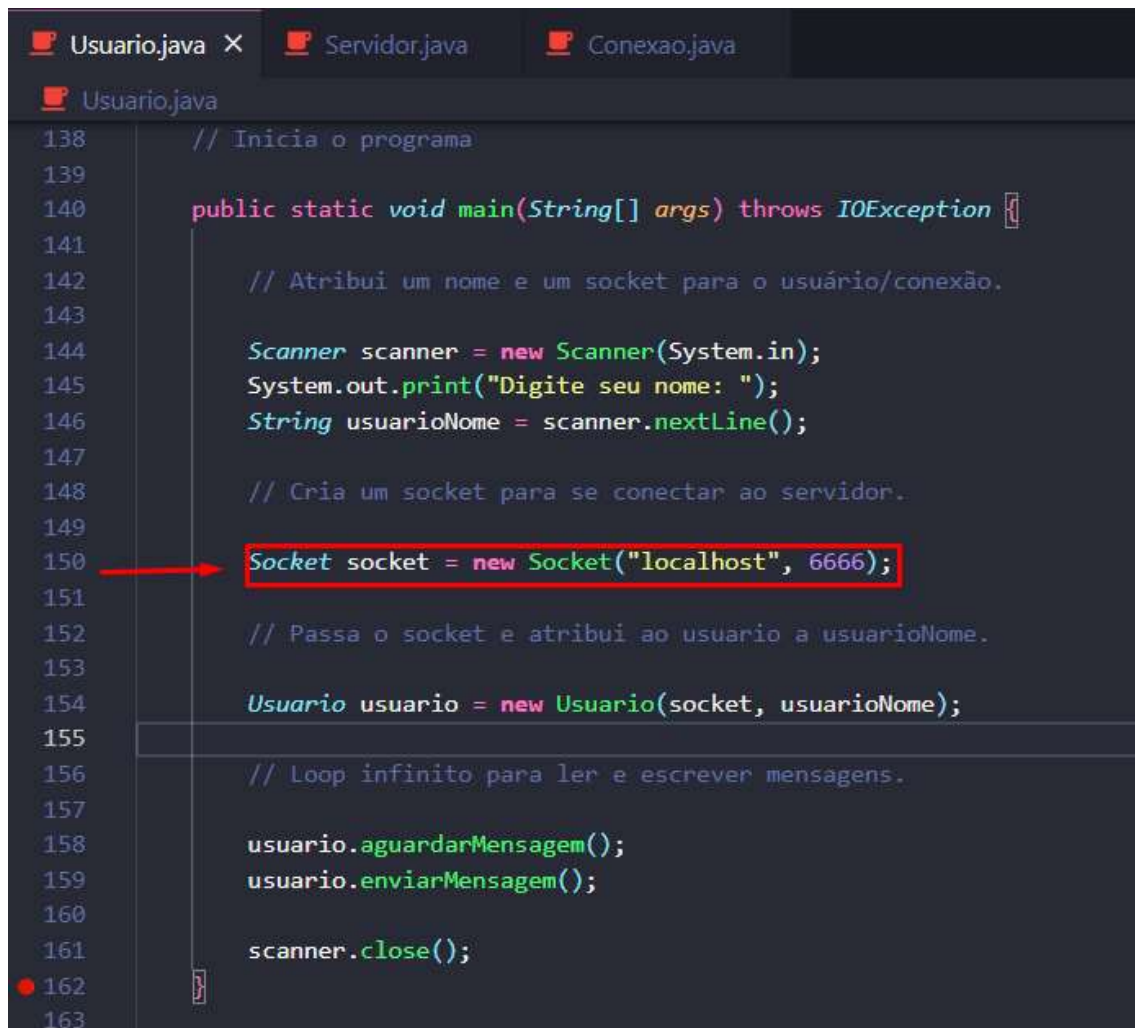
-Limpar o chat: Quando o usuário tiver vontade de limpar as mensagens do terminal, envie /limpar e o terminal vai ser limpo.

-Sair do chat: Quando o usuário tiver vontade de sair do chat, digite /sair e o programa irá fechar. Outros integrantes serão avisados que você saiu do chat.

-Horários das mensagens: Como em qualquer aplicativo de conversa, os horários das mensagens são essenciais. Toda vez que um usuário enviar a mensagem vai aparecer o horário de envio, o nome de quem enviou e a mensagem desejada.

-Número de usuários: O chat suporta várias pessoas simultaneamente na conversa, toda vez que alguém entrar ou sair todos os usuários online serão avisados

Para conectar diversas máquinas no bate papo, é necessário alterar o campo "Localhost" apresentado abaixo para o endereço ipv4 da máquina que está hospedando o servidor.



```
138 // Inicia o programa
139
140 public static void main(String[] args) throws IOException {
141
142     // Atribui um nome e um socket para o usuário/conexão.
143
144     Scanner scanner = new Scanner(System.in);
145     System.out.print("Digite seu nome: ");
146     String usuarioNome = scanner.nextLine();
147
148     // Cria um socket para se conectar ao servidor.
149
150     Socket socket = new Socket("localhost", 6666);
151
152     // Passa o socket e atribui ao usuario a usuarioNome.
153
154     Usuario usuario = new Usuario(socket, usuarioNome);
155
156     // Loop infinito para ler e escrever mensagens.
157
158     usuario.aguardarMensagem();
159     usuario.enviarMensagem();
160
161     scanner.close();
162
163 }
```

Desta forma, a próxima etapa é encaminhar o arquivo Usuario.java para as respectivas máquinas onde os usuários desejam conectar-se. O servidor suporta diversas máquinas/usuários.