

Laboratório 4 - Parte 2

Projeto de aplicação de “bate-papo distribuído ERT”

Funcionamento e testes

Nesta seção será apresentado como a aplicação funciona na prática, com imagens de cada momento que julgamos relevante ser mostrado, principalmente os resultados da execução dos comandos digitados pelo usuário. A seguir, será mostrado uma sequência de possíveis passos a serem seguidos do ponto de vista de execução só do servidor. Em seguida apresentaremos aqueles que relacionam o cliente com o servidor e por fim uma demonstração da troca de mensagens cliente-cliente.

- **Servidor**

1. O servidor ao receber uma conexão com o cliente printa o ip e a porta da mesma.

```
Iniciando o servidor...  
Servidor está pronto para receber conexões.  
Conectado com: ('127.0.0.1', 59216)
```

2. O servidor ao receber uma mensagem imprime em tela informando que recebeu uma mensagem e quem é o remetente e ao enviar uma mensagem informa no terminal que a mensagem foi devolvida e quem foi o destinatário.

```
Iniciando o servidor...  
Servidor está pronto para receber conexões.  
Conectado com: ('127.0.0.1', 59216)  
  
Mensagem recebida de ('127.0.0.1', 59216)!  
Mensagem devolvida para ('127.0.0.1', 59216)!  
  
Mensagem recebida de ('127.0.0.1', 59216)!  
█
```

3. O gerente do servidor ao executar o comando “ativos” obtém o número de clientes que estão conectados ao servidor.

```
ativos  
  
Existem 1 clientes ativos.  
█
```

4. O gerente do servidor ao executar o comando “hist” obtém todos os clientes que passaram pelo servidor.

```
hist  
  
Histórico de conexões: [('127.0.0.1', 59216), ('127.0.0.1', 59318)]  
█
```

5. O gerente do servidor ao executar o comando “fim” finaliza o servidor após todos os clientes terem se desconectado. Também é possível finalizar o servidor utilizando o comando “kill”, mas o servidor, nesse caso, será finalizado de forma abrupta sem esperar que os clientes se desconectem.

```
fim  
  
Existem conexões abertas com clientes.  
Novas conexões não serão aceitas.  
Aguardando a finalização dos clientes existentes...  
█
```

6. No caso do comando “fim”, após todos os clientes se desconectarem, os mesmo irá continuar a finalização como apresentado abaixo.

```
Existem conexões abertas com clientes.  
Novas conexões não serão aceitas.  
Aguardando a finalização dos clientes existentes...  
Mensagem recebida de ('127.0.0.1', 59318)!  
Todos os clientes terminaram de usar o servidor.  
Servidor finalizado!
```

- **Cliente-Servidor**

1. Do ponto de vista do cliente, a tela de home é a primeira a ser mostrada, nomeada com a de “autenticação” na documentação da aplicação. Nela é possível digitar os comandos para logar-se, criar um cadastro, deletar um cadastro ou sair da aplicação.

```
<===== Seja Bem-vindo ao Bate Papo ERT! =====>

<===== HOME =====>

Digite "login" para login.
Digite "criar" para criar seu cadastro.
Digite "deletar" para deletar o seu cadastro.
Digite "sair" para fechar a aplicação.

<=====>
```

2. O cliente ao digitar o comando “criar” pode se cadastrar no servidor e, para isso, será necessário passar um nome único e senha. Caso o usuário crie uma conta com um nick que já é utilizado por outro usuário, surgirá uma mensagem de erro informando que não foi possível realizar a ação.

```
criar
<=====Criando Seu Perfil=====>

Digite um nick: teste
Digite uma senha: 123
```

3. O cliente ao digitar o comando “deletar” é apresentado um espaço para que o usuário possa entrar com os dados de login e senha que deseja deletar. O cadastro só é deletado caso a autenticação esteja correta.

```
deletar
<=====Deletando o Perfil=====>
Digite o seu nick: teste
Digite a sua senha: 123
```

4. Ao digitar “sair”, o usuário sai da aplicação. A última notificação sempre aparece nas telas principais.

```
>>>> Conta deletada com sucesso.

<===== Seja Bem-vindo ao Bate Papo ERT! =====>

<===== HOME =====>

Digite "login" para login.
Digite "criar" para criar seu cadastro.
Digite "deletar" para deletar o seu cadastro.
Digite "sair" para fechar a aplicação.

<=====>

sair
Ficamos triste por sair, mas esperamos vê-lo em breve.
```

5. Após realizar o login, o usuário será direcionado para a tela do menu principal. Nessa tela é possível digitar os comandos para visualizar todos os usuários que estão conectados no servidor e se encontram ativos; mudar o status para ativo ou inativo; iniciar um chat com um usuário ativo; mostrar as últimas 5 notificações; se desconectar do servidor. Na imagem abaixo também é possível observar a mensagem de notificação de login do usuário que foi realizado com sucesso.

```
>>>> Logado com sucesso.

<===== MENU PRINCIPAL =====>

Logado como "teste"
Digite "usuarios" para visualizar usuários ativos.
Digite "ativar" para tornar seu status ativo.
Digite "inativar" para tornar seu status inativo.
Digite "chat: nome_login" para abrir o chat de conversa com o usuário "nome_login"
. Por exemplo: chat:fabio_Junior19
Digite "avisos" mostrar as 5 últimas notificações.
Digite "logout" para se deslogar da sua conta.

<=====>

█
```

6. Ao digitar “usuarios”, é disponibilizado para o usuário, no topo da tela de menu principal, a listagem de usuários ativos no sistema, inclusive ele mesmo.

```
>>>> USUÁRIOS ATIVOS
      lady_gaga
      teste

<===== MENU PRINCIPAL =====>

Logado como "teste"
Digite "usuarios" para visualizar usuários ativos.
Digite "ativar" para tornar seu status ativo.
Digite "inativar" para tornar seu status inativo.
Digite "chat: nome_login" para abrir o chat de conversa com o usuário "nome_login"
. Por exemplo: chat:fabio_Junior19
Digite "avisos" mostrar as 5 últimas notificações.
Digite "logout" para se deslogar da sua conta.

<=====>

█
```

7. Ao digitar o comando “ativar” ou “inativar” o status do usuário será redefinido e uma notificação será apresentada informando tal mudança.

```
>>>> Seu status foi redefinido.

<===== MENU PRINCIPAL =====>

Logado como "teste"
Digite "usuarios" para visualizar usuários ativos.
Digite "ativar" para tornar seu status ativo.
Digite "inativar" para tornar seu status inativo.
Digite "chat: nome_login" para abrir o chat de conversa com o usuário "nome_login"
. Por exemplo: chat:fabio_Junior19
Digite "avisos" mostrar as 5 últimas notificações.
Digite "logout" para se deslogar da sua conta.

<=====>

█
```

8. Ao digitar o comando “avisos” as últimas 5 notificações recebidas serão apresentados na tela.

```
>>>> Seu status foi redefinido.

-----
NOTIFICAÇÕES
>>>> Houve um problema. Não foi possível logar na conta. Tente novamente.
>>>> Conta criada com sucesso.
>>>> Logado com sucesso.
>>>> Seu status foi redefinido.
>>>> Seu status foi redefinido.
-----

<===== MENU PRINCIPAL =====>

Logado como "teste"
Digite "usuarios" para visualizar usuários ativos.
Digite "ativar" para tornar seu status ativo.
Digite "inativar" para tornar seu status inativo.
Digite "chat: nome_login" para abrir o chat de conversa com o usuário "nome_login"
. Por exemplo: chat:fabio_Junior19
Digite "avisos" mostrar as 5 últimas notificações.
Digite "logout" para se deslogar da sua conta.

<=====>

█
```

9. Ao digitar o comando “logout”, o usuário será deslogado da sua conta e redirecionado para a tela de home.

```
>>>> Desconectado com sucesso

<===== Seja Bem-vindo ao Bate Papo ERT! =====>

<===== HOME =====>

Digite "login" para login.
Digite "criar" para criar seu cadastro.
Digite "deletar" para deletar o seu cadastro.
Digite "sair" para fechar a aplicação.

<=====>

█
```

- **Cliente-Cliente**

1. Ao digitar o comando “chat:” seguido do nome do usuário, o usuário será direcionado para a tela do chat, abrindo um canal de comunicação com o usuário com quem se deseja conversar.

```
>>>> Logado com sucesso.

<===== MENU PRINCIPAL =====>

Logado como "teste"
Digite "usuarios" para visualizar usuários ativos.
Digite "ativar" para tornar seu status ativo.
Digite "inativar" para tornar seu status inativo.
Digite "chat: nome_login" para abrir o chat de conversa com o usuário "n
ome_login". Por exemplo: chat:fabio_Junior19
Digite "avisos" mostrar as 5 últimas notificações.
Digite "logout" para se deslogar da sua conta.

<=====

chat:lady_gaga
```

2. Em seguida, o usuário “lady_gaga” irá receber uma notificação informando que o usuário “teste” iniciou um canal de chat com ele. O mesmo pode ignorar tal notificação ou entrar na sala de chat.

```
>>>> teste iniciou um chat com você! Digite "chat:teste para abrir o chat."

<===== MENU PRINCIPAL =====>

Logado como "lady_gaga"
Digite "usuarios" para visualizar usuários ativos.
Digite "ativar" para tornar seu status ativo.
Digite "inativar" para tornar seu status inativo.
Digite "chat: nome_login" para abrir o chat de conversa com o usuário "nome_login". Por
exemplo: chat:fabio_Junior19
Digite "avisos" mostrar as 5 últimas notificações.
Digite "logout" para se deslogar da sua conta.

<=====


```

3. Caso o usuário "lady_gaga" deseje abrir o chat iniciado, deverá digitar "chat:teste" dando o comando para que a aplicação abra o chat com o usuário "teste"

```
>>>> teste iniciou um chat com você! Digite "chat:teste para abrir o chat."

<===== MENU PRINCIPAL =====>

Logado como "lady_gaga"
Digite "usuarios" para visualizar usuários ativos.
Digite "ativar" para tornar seu status ativo.
Digite "inativar" para tornar seu status inativo.
Digite "chat: nome_login" para abrir o chat de conversa com o usuário "nome_login". Por
exemplo: chat:fabio_Junior19
Digite "avisos" mostrar as 5 últimas notificações.
Digite "logout" para se deslogar da sua conta.

<=====>

chat:teste
```

4. Uma vez aberto o canal de comunicação, os usuários visualizarão a tela de chat podendo, assim, digitar mensagens entre si ou utilizarem palavras reservadas como "fechar" e "apagar" para fechar a janela de chat e retornar ao menu principal ou apagar as mensagens enviadas, respectivamente.

```
>>>> Chat aberto com sucesso.

<===== CHAT COM: lady_gaga =====>

Digite "fechar" para voltar ao menu principal.
Digite "apagar" para apagar mensagens do chat.
Digite sua mensagem a ser enviada para este usuário.

<=====>


```


5. Abaixo segue um exemplo da troca de mensagens feita entre os usuários “teste” e “lady_gaga”.

<pre>>>>> Chat aberto com sucesso. <===== CHAT COM: lady_gaga =====> Digite "fechar" para voltar ao menu principal. Digite "apagar" para apagar mensagens do chat. Digite sua mensagem a ser enviada para este usuário. <=====> █</pre>	<pre>>>>> Chat aberto com sucesso. <===== CHAT COM: teste =====> Digite "fechar" para voltar ao menu principal. Digite "apagar" para apagar mensagens do chat. Digite sua mensagem a ser enviada para este usuário. <=====> █</pre>
<pre>>>>> Chat aberto com sucesso. <===== CHAT COM: lady_gaga =====> Digite "fechar" para voltar ao menu principal. Digite "apagar" para apagar mensagens do chat. Digite sua mensagem a ser enviada para este usuário. <=====> teste:Oie! lady_gaga:Olá! Como Você vai? █</pre>	<pre>>>>> Chat aberto com sucesso. <===== CHAT COM: teste =====> Digite "fechar" para voltar ao menu principal. Digite "apagar" para apagar mensagens do chat. Digite sua mensagem a ser enviada para este usuário. <=====> teste:Oie! lady_gaga:Olá! Como Você vai? █</pre>

Pontos importantes e modificações

Durante o desenvolvimento desta aplicação, nos deparamos com vários obstáculos, considerando a documentação proposta no material da entrega passada. A fim de superar tais problemas, algumas modificações foram feitas no que ficou definido no passado. Sendo assim, abaixo segue algumas dessas mudanças e também algumas decisões importantes a nível de estruturação e implementação, assim como funcionalidades não pretendidas previamente mas que se tornaram necessárias.

- Inicialmente não havíamos pretendido ter um comando do tipo “kill” disponível para o gerente do servidor, contudo, devido a alguns problemas que fomos encontrando ao longo do desenvolvimento da aplicação, julgamos importante a sua existência.
- Na arquitetura de software houveram poucas modificações, alterando somente o fato de que, do ponto de vista cliente-servidor, no cliente não há a implementação de parte da camada de dados, ela ficou por inteiro do lado do servidor.
- Um ponto muito relevante é que dado a decisão de limpar a tela várias vezes, por conta das atualizações que ocorrem nessa tela, fica difícil a leitura do que se digita por mais que o terminal pegue tudo que o usuário digite na entrada padrão.
- Em relação a deleção de cadastros, é válido ressaltar que não é necessário que haja o login do usuário para que a ação ocorra, um ponto que diferia da decisão da

documentação inicial. É importante explicitar que a opção de deletar cadastro foi colocada na tela de credencial por uma questão de contexto.

- Caso o usuário faça o logout e não saia da aplicação, as mensagens que o mesmo trocou até o momento não serão perdidas. Assim, se o usuário abrir um chat com os usuários novamente, irá ver as últimas mensagens trocadas.
- Alguns métodos entre o cliente-servidor tiveram o formato das mensagens trocadas alteradas, como por exemplo o método “authAccount” que passamos a incluir a porta de comunicação a ser passada na requisição e o método “getMyStatus” que incluímos a senha senha na requisição.
- É Importante dizer, que a camada de interface do cliente e a camada de processamento do cliente acaba por se comunicar através de trocas de JSON na maior parte das vezes. Porém não é a única forma de ambas se comunicarem.
- Enquanto o usuário está em um chat com outro usuário, a aplicação não fica impedida de receber novas mensagens de outros usuários. Tais mensagens serão armazenadas e futuramente apresentadas quando o usuário abrir o chat com o usuário em questão.

Dificuldades e a experiência

Um obstáculo inicial que encontramos com este trabalho foi estruturar como seria o funcionamento na prática de uma arquitetura peer-to-peer dado que, apesar de termos estudado durante a disciplina de modo teórico, não vimos sua implementação num caso mais “real”. Embora essa dificuldade tenha existido no início, depois de algumas discussões e com base no que aprendemos fluiu um pouco mais facilmente como isso seria feito.

Uma dificuldade enfrentada foi com os inputs no terminal durante a troca de mensagens com os demais usuários. Os mesmos são constantemente “deletados” por conta do dinamismo das telas que ficam atualizando, para sempre apresentar as novas mensagens que chegam. Vale ressaltar que os inputs que foram digitados não são de fato descartados, apenas somem do terminal. Apesar da longa busca por uma solução para este problema, ela não foi encontrada. Outro desafio foi definir qual a porta que o usuário iria disponibilizar quando assumisse o lado passivo da comunicação, uma vez que existe a possibilidade da porta escolhida já estar sendo utilizada. Para tal desafio, contornamos criando um método que testa um determinado range de portas até encontrar uma porta disponível.

Inicialmente os usuários estavam trocando entre si as mensagens de conversa e também mensagens de sucesso desse envio. No entanto, identificamos uma situação em que estava ocorrendo deadlock e, assim, os usuários não conseguiam mais se comunicar entre si. Tal obstáculo foi contornado retirando tais mensagens de sucesso.

Por fim, outro problema foi com o envio de mais de uma mensagem por chamada do send. Tal obstáculo foi contornado realizando uma análise em todas as mensagens recebidas, caso seja identificada mais de uma mensagem as mesmas serão separadas.

Ao executar este tipo de trabalho, foi possível que entendêssemos melhor e mais a fundo como funcionam os sistemas distribuídos nas arquiteturas híbridas e, conseqüentemente, a peer-to-peer. Por mais que existam várias abstrações de middleware que poderiam nos ter facilitado na implementação, realizar a codificação ainda a nível de socket nos proveu o conhecimento de como tais abstrações poderiam funcionar por trás. De certo modo, em alguns momentos “implementamos” lógicas de programação que se assemelham a ideia do RPC e do sistema de fila de mensagens - abstrações que podem ser providas por middlewares.

Deter este tipo de conhecimento pode nos garantir um melhor entendimento das abstrações mencionadas, bem como outras de mesma categoria que venhamos a ter contato no futuro. Além disso, pudemos colocar em prática outros conteúdos importantes dentro da Computação, como redes e computação concorrente.