

Relatório - Redes de Computadores EP1

Tema: Sistema de reservas de Hotel.

Integrante: Icaro Lima de Oliveira N°USP

Integrante: João Neves da Silva Santos

1 - Proposta:

A proposta do projeto foi criar um sistema de reservas de hotel onde o cliente através do Sockets-Client-Reservas-1.0-SNAPSHOT.jar conseguisse realizar uma conexão TCP com o servidor, que executa através do Sockets-Reservas-1.0-SNAPSHOT.jar. Está conexão seria por meio de Sockets (para isto foi utilizada a API nativa java.net que contém as classes Socket e ServerSocket) e a troca de mensagens seria entre o cliente que envia apenas conteúdos específicos como o nome o qual quer ser chamado, data que deseja realizar a reserva, data para qual deseja alterar a reserva ou cancelar a reserva e tudo isso por meio de um “bate-papo” com um ChatBot que é orquestrado pelo servidor.

2 - Especificações:

O projeto foi criado para que o cliente siga um fluxo esperado e previsível, assim facilitando as mensagens esperadas e como devem ser respondidas. O servidor gera todo o texto gerado pelo servidor e simula início das mensagens dos clientes deixando variável somente os campos importantes como o nome a qual o cliente quer ser chamado, a data desejada da reserva, opções de alteração ou cancelamento e etc.

O protocolo segue o objetos MessageServer e MessageClient como base em que o servidor inicia enviando pelo canal de comunicação por meio de uma conexão TCP um objeto MessageServer enquanto o cliente recebe e responde com um objeto MessageClient que são serializáveis, para isso foi utilizado a classe ObjectOutputStream e ObjectInputStream para envio e recebimento destas objetos que representam as mensagens.

3 - Como ler o código, fluxo de funcionamento, testes e suas classes:

MessageServer e MessageClient - Classes que representam as mensagens do servidor e cliente.

O MessageServer é a mensagem que é enviada pelo servidor e o MessageClient é a mensagem que é enviada pelo cliente. As duas classes são semelhantes contendo ambas os seguinte atributos:

1. string sender: nome do remetente da mensagem.
2. string recipient: nome do destinatário da mensagem.
3. boolean close_connection: utilizado para verificação se deve fechar a conexão.
4. string content: conteúdo da mensagem.

O que diferencia as classes é que as mensagens enviadas pelo servidor podem contém diversas linhas de conteúdo (vetor de strings content) enquanto o cliente envia mensagens com somente uma linha (um atributo string content) de conteúdo.

ClientApplicationSocket - Classe que simula o Cliente.

Esta classe é utilizada para o cliente realizar a conexão com o ServerSocket (servidor) e realizar a troca de mensagens. Deve ser executada após o servidor já está de pé (executando), o fluxo desta classe é inicializar um Socket com o endereço de conexão do servidor, se na inicialização do ServerApplicationSocket for passada algum número de porta por parâmetro deve ser passado a mesma porta por parâmetro na execução da classe ClientApplicationSocket (se não for passado parâmetro ambas as classes utilizarão por padrão a porta 4545).

Após inicializar o Socket e realizar a conexão com o servidor a classe entrará em um loop que funcionará da seguinte forma:

1. Lê os dados enviados pelo servidor e realiza o casting para um objeto MessageServer.
2. Recupera os atributos do MessageServer, realiza a impressão no console de todo o conteúdo enviado pelo servidor e verifica se deve fechar a conexão.
3. Caso não feche a conexão deve ler os dados de entrada do teclado para montar o conteúdo do MessageClient e enviar em resposta ao servidor.
4. Volta para o passo 1 e repete o fluxo.

O fluxo sairá do loop em duas ocasiões: se em algum momento o cliente enviar um MessageClient com conteúdo "FIM" ou finalizar um fluxo de reserva, alteração de reserva ou cancelamento de reserva. Ao sair do loop fecha a conexão do lado do cliente e finaliza o programa.

ServerApplicationSocket - Classe que simula o servidor.

O fluxo de funcionamento desta classe que desempenha o papel de servidor é inicialmente subir um ServerSocket em alguma porta (espera que seja passada por parâmetro, se não é utilizado a porta 4545), inicializa o vetor de booleanos calendar que simula o mês de janeiro (utilizado para gestão das reservas e criação da tabela de dias de janeiro no terminal) e inicializa o vetor de strings tickets_calendar que guarda o ticket da reserva realizada na posição correspondente ao dia reservado no vetor calendar (utilizado para validação em alteração ou cancelamento de reservas).

Após estas inicializações entra no loop em que aguarda solicitações de conexões de clientes, caso seja realizada alguma o mesmo aceita a conexão e passa o Socket o do cliente para a ControllerServer, classe que estende Thread, e em seguida usa o método start() para iniciar de forma assíncrona o método run() da classe em uma nova linha de execução para assim a classe ServerApplicationSocket na Thread main voltar ao loop em que aguarda e aceita novas conexões de clientes enquanto a classe ControllerServer trata a conexão do cliente.

```
icaro@BOOK-R84VHARNK MINGW64 ~/Documents/Faculdade/Redes/EP1/Sockets-Reservas/out (main)
$ java com/redesdecomputadores/ep1/server/ServerApplicationSocket

/SSSSSS /SS /SS
/SS_ SS
| SS \_ / /SSSSSS /SSSSSS /SS /SS/SS /SSSSSS /SSSSSS /SSSSSS
| SSSSSS /SS_ SS /SS_ SS| SS /SS/ SS /SS_ SS /SS_ SS /SS_ SS
\_ SS| SSSSSSS| SS \_ / SS/SS/| SS| SS | SS| SS \ SS| SS \_ /
/SS \ SS| SSSSSS/| SS SSS/| SS| SS | SS| SS | SS| SS
| SSSSSS/| SSSSSS| SS \ $/ | SS| SSSSSS| SSSSSS/| SS
```

Sendo assim o loop na classe ServerApplicationSocket segue o seguinte fluxo:

1. Aceita requisição de conexão feita pelo cliente criando um Socket que representa a conexão do cliente.
2. Cria uma instância de ControllerServer em que passa o Socket.
3. Executa o método start() para tratar a solicitação do cliente em uma nova linha de execução.
4. Volta para o passo 1 e repete o fluxo.

ControllerServer- Classe que direciona o cliente para o serviço requisitado.

Em ControllerServer é feita a triagem inicial onde inicia realmente a troca de mensagens entre cliente e servidor. Primeiramente, é solicitado o nome que o cliente quer ser chamado (Screenshot 2) e o serviço que ele deseja utilizar (Screenshot 3), nesta ordem descrita:

```
icaro@BOOK-R84VHARNK MINGW64 ~/Documents/Faculdade/Redes/EP1/Sockets-Reservas/out (main)
$ java com/redesdecomputadores/ep1/client/ClientApplicationSocket
/$$$$$$ /$$ /$$ /$$$$$$$ /$$
/$$ _ $$ /$$ /$$ /$$$$$ /$$
| $$ \_ /| $$$$$$ /$$$$$ /$$$$$ | $$ _ $ /$$$$$ /$$$$$
| $$ | | | $$$$ $ /$$$$$ | $$/
| $$ | | | $$$$ $ /$$$$$ | $$/
| $$ | | | $$$$ $ /$$$$$ | $$/
| $$ | | | $$$$ $ /$$$$$ | $$/
| $$$$/| $$ | $$| $$$$$$ | $$$$/
\_____/|_/ |_/ \_____/ \_____/

06-12-2023 12:13:43 >> Iniciando configurações do socket do cliente para realizar conexão com servidor...
06-12-2023 12:13:43 >> Endereço de porta não fornecido. Conectando em 127.0.0.1 na porta padrão: 4545
06-12-2023 12:13:43 >> Iniciando conexão com servidor no endereço: 127.0.0.1/4545
06-12-2023 12:13:43 >> Solicitação de conexão bem sucedida!
-----CHATBOT-----
06-12-2023 12:13:43 >> ChatBot: Olá, sou o ChatBot que irá te auxiliar na alteração ou realização da sua reserva!
06-12-2023 12:13:43 >> ChatBot: Primeiramente, como gostaria de ser chamado?
06-12-2023 12:13:43 >> ChatBot: Lembrando que caso queira finalizar o chat a qualquer momento é só enviar 'FIM'.
06-12-2023 12:13:43 >> Usuário50929: Gostaria de ser chamado por:
```

2. Screenshot da execução do ClientApplicationSocket em que recebe as primeiras mensagens do servidor.

```
-----CHATBOT-----
06-12-2023 12:13:43 >> ChatBot: Olá, sou o ChatBot que irá te auxiliar na alteração ou realização da sua reserva!
06-12-2023 12:13:43 >> ChatBot: Primeiramente, como gostaria de ser chamado?
06-12-2023 12:13:43 >> ChatBot: Lembrando que caso queira finalizar o chat a qualquer momento é só enviar 'FIM'.
06-12-2023 12:13:43 >> Usuário50929: Gostaria de ser chamado por: Icaro
06-12-2023 12:21:46 >> ChatBot: Perfeito, prazer em atendê-lo Icaro
06-12-2023 12:21:46 >> ChatBot: Para quais dos serviços abaixo seria seu contato:
1 - Realizar reserva.
2 - Alterar ou cancelar data da reserva.
06-12-2023 12:21:46 >> ChatBot: Digite o número da opção que deseja.
06-12-2023 12:21:46 >> Icaro: opção número:
```

3. Screenshot após cliente inserir seu nome e receber a segunda mensagem com opções de serviços disponíveis.

Cada um dos dois serviços são realizados por classes diferentes. Na realização de reservas é utilizado a classe ReservationService, na alteração ou cancelamento de reserva é utilizado a classe ChangeReserveService. Logo, dependendo da opção que o cliente enviar no conteúdo da mensagem será feito um fluxo diferente, após enviar a opção o fluxo de realização de reserva, alteração ou cancelamento continua nas respectivas classes.

ReservationService - Classe que realiza o serviço de reservas.

A instância desta classe é inicializada dentro da classe ControllerService, nela é passada como parâmetro o Socket do cliente, ObjectInputStream e ObjectOutputStream (classes utilizadas para realizar troca de dados do tipo objeto via Sockets).

Nesta inicialização da classe é enviado a mensagem do servidor com os próximos passos para realização da reserva bem como os dias disponíveis.

```
06-12-2023 13:16:22 >> Icaro: Opção número: 1
06-12-2023 13:16:26 >> ChatBot: Ok, você escolheu o serviço de realizar reservas.
06-12-2023 13:16:26 >> ChatBot: No momento só temos reservas para Janeiro/2024, para qual dia disponível seria o início da reserva?
06-12-2023 13:16:26 >>
    Janeiro de 2024
Dom  Seg  Ter  Qua  Qui  Sex  Sáb
[xx] [01] [02] [03] [04] [xx] [06]
[07] [08] [09] [10] [11] [12] [13]
[14] [15] [16] [17] [18] [19] [20]
[21] [22] [23] [24] [25] [26] [27]
[28] [29] [30] [31] [xx] [xx] [xx]
06-12-2023 13:16:26 >> ChatBot: As datas marcadas com '[xx]' estão indisponíveis, as demais datas com seu respectivo dia estão disponíveis.
06-12-2023 13:16:26 >> Icaro: Entendido! O dia de início da reserva seria:
```

4. Screenshot após inicialização da classe ReservationService e continuação do fluxo de reservas pelo terminal.

Com a tabela do mês de Janeiro e os dias disponíveis é esperado do cliente o envio da data que deseja realizar a reserva. Após a seleção de uma data válida e disponível e confirmada a reserva do dia em questão e também é gerado um “ticket” que deve ser guardado pelo usuário, o ticket será necessário em fluxos de alteração da data de reserva e cancelamento da reserva, em ambos os casos é utilizado para realizar a validação do usuário e não deixar que outros clientes alterem ou cancele reservas de outros clientes.

```
06-12-2023 15:49:00 >> Icaro: Entendido! O dia de início da reserva seria: 10
06-12-2023 15:49:04 >> ChatBot: Sua reserva para o dia 10 de Janeiro de 2024 foi efetuada com sucesso!
06-12-2023 15:49:04 >> ChatBot: Seu ticket da reserva é: dzqwcvrq42. Lembre-se de guarda-lo para caso queira alterar ou cancelar sua reserva.
06-12-2023 15:49:04 >> ChatBot: Espero te-lo ajudado e o aguardamos em nosso hotel em 010/01/2024. Encerrando ChatBot...
06-12-2023 15:49:04 >> Conexão com servidor encerrada.

icaro@BOOK-R84VVHARNK MINGW64 ~/Documents/Faculdade/Redes/EPI/Sockets-Reservas/out (main)
$ |
```

ChangeReserveService - Classe que realiza o serviço de alteração de data e cancelamento de reservas

- ```

/##### /$ /$
/$_ $ /$ /$
| $ \ /##### /$ /$ /$ /$ /$ /$ /$ /$
| ##### /$ _ $| $ /$ /$ /$ _ $ /$ _ $ /$ _ $
\ _ $| ##### $ \ / $/$/| $| $| $| $ \ $| $ \
/$$ \ $| $ _/| $ \ $$/| $| $| $| $| $| $| $
| #####| ##### $ \ $/| $| ##### $$$$/| $
\ _/ \ _/| \ _/ \ _/ \ _/ \ _/

```
- 06-12-2023 13:15:47 >> Iniciando Servidor...
- 06-12-2023 13:15:47 >> Não foi fornecido número de porta para SocketServer. Subindo servidor na porta: 4545
- 06-12-2023 13:15:47 >> Servidor inicializado no endereço: 0.0.0.0/0.0.0.0:4545. Aguardando conexões de clientes....
- 06-12-2023 13:15:51 >> Nova conexão com cliente - Endereço: /127.0.0.1:52181
- 06-12-2023 13:15:51 >> Reservada Thread-0 para cliente: 52181
- 06-12-2023 15:25:38 >> Encerrada conexão com Cliente: 52181. Thread-0 liberada.

- ```
06-12-2023 15:35:52 >> Icaro: opção número: 2
06-12-2023 15:35:54 >> ChatBot: Ok, você escolheu o serviço de alteração ou cancelamento de reservas. Lembre-se que
irá precisar do ticket gerado no momento da reserva.
06-12-2023 15:35:54 >> ChatBot: No momento só temos reservas para Janeiro/2024, para qual data de Janeiro está agend
ado sua reserva?
06-12-2023 15:35:54 >>
    Janeiro de 2024
Dom  Seg  Ter  Qua  Qui  Sex  Sáb
[xx] [01] [02] [03] [04] [xx] [06]
[07] [08] [09] [xx] [11] [12] [13]
[14] [15] [16] [17] [18] [19] [20]
[21] [22] [23] [24] [25] [26] [27]
[28] [29] [30] [31] [xx] [xx] [xx]

06-12-2023 15:35:54 >> Icaro: A data da minha reserva é:
```

- Neste momento é pedido ao usuário que seja inserido a data da sua reserva que irá ser alterada ou cancelada, também é impresso o mês de Janeiro para auxiliar o usuário.

```

06-12-2023 15:50:45 >> Icaro: A data da minha reserva é: 10
06-12-2023 15:50:47 >> ChatBot: Perfeito, agora precisamos validar sua reserva. Por favor, envie o ticket gerado no momento da reserva feita para 10/Janeiro/2024.
06-12-2023 15:50:47 >> Icaro: Meu ticket é: dzqwcvg42
06-12-2023 15:51:08 >> ChatBot: Ticket validado com sucesso! Agora selecione o serviço que deseja realizar para sua reserva:
    1 - Alterar data da reserva.
    2 - Cancelar reserva.
06-12-2023 15:51:08 >> ChatBot: Por favor, envie o número do serviço que deseja (1 ou 2).
06-12-2023 15:51:08 >> Icaro opção: |

```

8. Screenshot da seleção da data que foi realizada a reserva. Em seguida solicitado o ticket pelo servidor e enviado o ticket pelo cliente.

Com o envio da sua data de reserva é validado o ticket, se for validado com sucesso e disponibilizado os serviços para a data de reserva que foi validada. Seguindo o fluxo para a opção de alteração da data é será necessário a escolha de uma nova data de reserva e o cancelamento da data anterior, assim é alterada a reserva e o ticket não é gerado novamente apenas transferido da data anterior para a data atual sendo possível realizar novamente alteração ou cancelamento da reserva.

```

06-12-2023 15:51:08 >> Icaro Opção: 1
06-12-2023 15:53:32 >> ChatBot: ok, você selecionou a opção de alteração da data de reserva. Insira a nova data para qual deseja alterar a reserva.
06-12-2023 15:53:32 >>
    Janeiro de 2024
    Dom Seg Ter Qua Qui Sex Sáb
    [xx] [01] [02] [03] [04] [xx] [06]
    [07] [08] [09] [xx] [11] [12] [13]
    [14] [15] [16] [17] [18] [19] [20]
    [21] [22] [23] [24] [25] [26] [27]
    [28] [29] [30] [31] [xx] [xx] [xx]
06-12-2023 15:53:32 >> ChatBot: As datas marcadas com '[xx]' estão indisponíveis, as demais datas com seu respectivo dia estão disponíveis.
06-12-2023 15:53:32 >> Icaro: Desejo alterar a minha reserva para: 18
06-12-2023 15:53:40 >> ChatBot: Perfeito! Reserva do dia 10/Janeiro/2024 alterada para 18/Janeiro/2024.
06-12-2023 15:53:40 >> ChatBot: Seu ticket de reserva continua sendo o mesmo da data anterior, lembre-se de guardá-lo.
06-12-2023 15:53:40 >> ChatBot: Encerrando ChatBot. Espero vê-lo em breve, até!
06-12-2023 15:53:40 >> Conexão com servidor encerrada.
icaro@BOOK-R84VVHARNK MINGW64 ~/Documents/Faculdade/Redes/EP1/sockets-Reservas/out (main)

```

9. Screenshot da opção de alteração da reserva em que é pedido uma nova data. Após seleção da nova data cliente mantém o mesmo ticket e a reserva é alterada para data selecionada.

Assim se encerra o fluxo de alteração da data de reserva. Já para o cancelamento da data, é necessário na etapa descrita na Screenshot 8 ser selecionado a opção “2 - Cancelar Reserva” em que a data da sua reserva é cancelada e fica novamente disponível para ser reservada.


```

06-12-2023 16:07:57 >> Icaro: A data da minha reserva é: 18
06-12-2023 16:08:01 >> ChatBot: Perfeito, agora precisamos validar sua reserva. Por favor, envie o ticket gerado no
momento da reserva feita para 18/Janeiro/2024.
06-12-2023 16:08:01 >> Icaro: Meu ticket é: dzqwcvg42
06-12-2023 16:08:40 >> ChatBot: Ticket validado com sucesso! Agora selecione o serviço que deseja realizar para sua
reserva:
    1 - Alterar data da reserva.
    2 - Cancelar reserva.
06-12-2023 16:08:40 >> ChatBot: Por favor, envie o número do serviço que deseja (1 ou 2).
06-12-2023 16:08:40 >> Icaro Opção: 2
06-12-2023 16:08:45 >> ChatBot: Ok, você selecionou a opção de cancelamento de reserva.
06-12-2023 16:08:45 >> ChatBot: Sua reserva para 18/Janeiro/2024 foi cancelada com sucesso.
06-12-2023 16:08:45 >> ChatBot: Encerrando ChatBot. Espero ve-lo em breve, até!
06-12-2023 16:08:45 >> Conexão com servidor encerrada.

icaro@BOOK-R84VVHARNK MINGW64 ~/Documents/Faculdade/Redes/EP1/Socketes-Reservas/out (main)
$

```

10. Screenshot da etapa de escolha da opção de cancelamento da reserva e o cancelamento sendo feito com sucesso.

DateTimeLog - Classe com máscara para geração de data e hora com minutos e segundos.

Está classe no projeto só contém um método para geração da data e hora com minutos e segundos atuais para que seja gerado tanto nos logs do servidor quanto no fluxo de conversas com o ChatBot.

4 - Como compilar ou executar:

O projeto contém o script de execução dos arquivos.jar dos projetos (cliente e servidor) logo não há a necessidade de compilação e execução via terminal, porém caso seja desejado, para realizar a compilação rodar o seguinte comando a partir da pasta raiz (Socketes-Reservas ou Socketes-Client-Reservas) do projeto em um terminal bash:

- `javac -d out src/main/java/com/redesdecomputadores/ep1/*/*.java`

O comando é o mesmo para o projeto cliente ou servidor. Agora para executá-los acesse pelo terminal bash a pasta out com “cd out” e execute o seguinte comando:

- Projeto Cliente: `java com/redesdecomputadores/ep1/main/ClientApplicationSocket`
- Projeto Servidor: `java com/redesdecomputadores/ep1/main/ServerApplicationSocket`

Agora se deseja apenas executar os projetos por meio dos arquivos .jar nas pastas target dos projetos (Socketes-Client-Reservas-1.0-SNAPSHOT.jar do projeto cliente e Socketes-Reservas-1.0-SNAPSHOT.jar do projeto servidor) execute o script **run.bat** (feito para execução em windows) na pasta raiz do projeto dando um duplo clique.

Lembre-se de primeiro executar o run.bat do Socketes-Reservas que representa o servidor para depois executar quantas instâncias desejar do Socketes-Client-Reservas que representa o cliente.

5 - Como interagir com o projeto:

Após executar através dos scripts run.bat o servidor e cliente basta apenas seguir o fluxo informado pelo ChatBot que informa e instrui as opções e como deve ser feito as interações.