Universidade de Brasília
Departamento de Ciência da Computação (2018/2)

Transmissão de Dados – Turma A – Trabalho 1
Professor: Marcos F. Caetano <mfcaetano@unb.br>
Monitoras: Mariana Makiuchi <marimakiuchi@aluno.unb.br>
Resumo: Desenvolvimento de um sistema de chat *online*

1 Introdução

Chats online são definidos como sendo qualquer tipo de comunicação realizada por meio da troca de mensagens de texto em tempo real através da Internet. Atualmente, diversas plataformas e aplicativos oferecem esse tipo de serviço, tais como Whatsapp, Telegram, Skype, Line, Slack e o *chat* do Facebook (Messenger). Além de permitirem a troca de mensagens de texto em tempo real, esses aplicativos oferecem também outros recursos, como por exemplo:

- Chamadas de vídeo ou ligações;
- Envio de arquivos, fotos, vídeos e aúdio;
- Compartilhamento de telas;
- *Chats* em grupo (texto, ligação ou vídeo);
- Bloquear mensagens de um usuário específico;
- Envio de localização GPS.

Os serviços de troca de mensagem em *chats online* se tornaram extremamente populares a ponto de fazerem parte do cotidiano de grande parte da população mundial. Na verdade, em 2016, o aplicativo Whatsapp somado ao Facebook Messenger precisavam processar 60 bilhões de mensagens por dia [1]. Apesar da popularidade atual, o serviço de *chat online* é relativamente antigo e surgiu em 1973 com a criação do sistema Talkomatic, cujas versões mais atuais estão disponíveis em [2].

No sistema Talkomatic e em outros que surgiram no final do século XX, as trocas de mensagens acontecem dentro das salas virtuais de bate-papo. Nestas salas, é possível trocar mensagens com todos aqueles que estão *online* dentro da mesma sala. No sistema Talkomatic, é possível criar salas públicas, nas quais qualquer usuário pode entrar, ou privadas. O usuário que criar uma sala privada deve elaborar uma senha para a sala, que deve ser apresentada por todos os usuários que desejam entrar na sala criada.

Inspirado nos antigos sistemas de *chat online*, é proposto que seja desenvolvido um sistema de salas de bate-papo, que será descrito com maiores detalhes nas próximas secções.

2 Descrição do projeto

De forma geral, o projeto consiste em duas partes (programas) que se comunicam: o lado servidor e o lado cliente. O lado cliente requisita serviços ao servidor que, por sua vez, deve fornecer esses serviços corretamente. No caso do *chat* a ser desenvolvido, os lados cliente e servidor se comportam segundo os diagramas apresentados pelas Figuras 1 e 2, respectivamente.

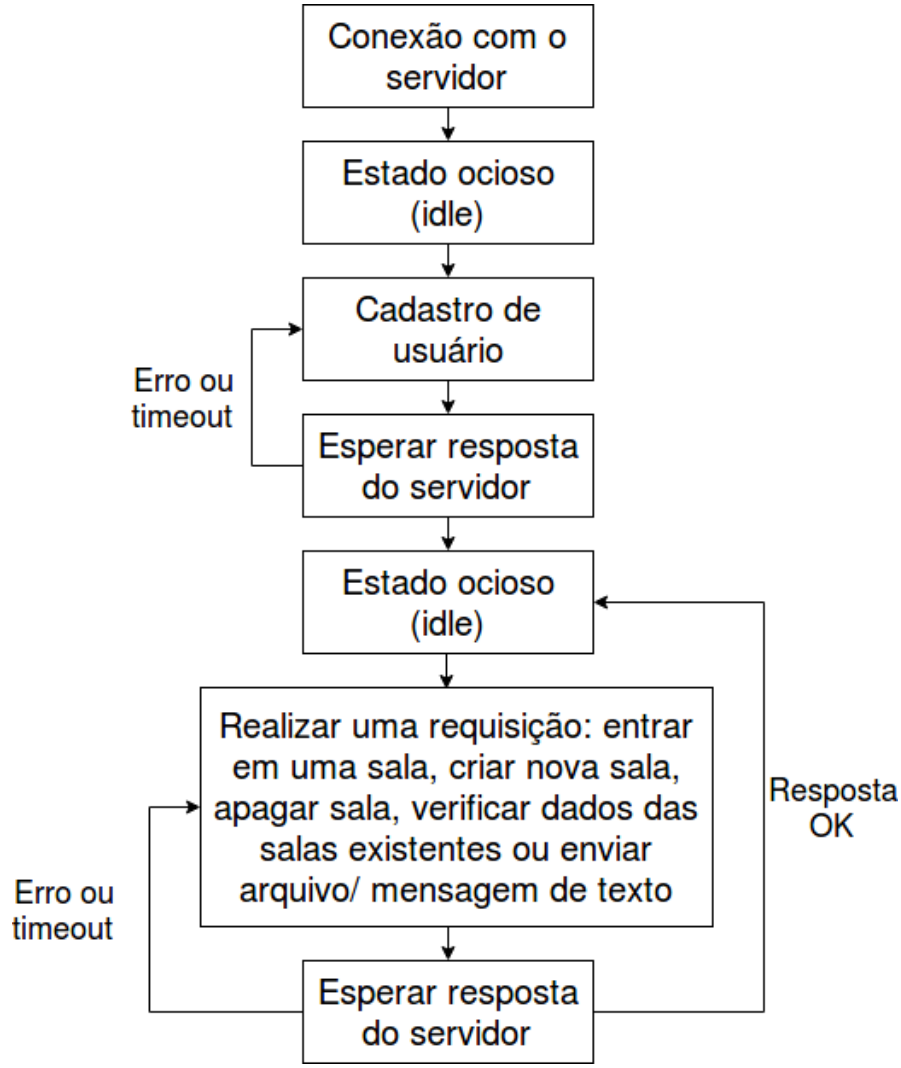

Figura 1: Diagrama do funcionamento do lado cliente da aplicação

Assim, de forma pontual, o projeto a ser desenvolvido deverá ser capaz de realizar as seguintes funcionalidades, considerando tanto o lado cliente quanto o servidor:

1. Cadastro de usuários;
2. Solicitar a lista de salas de bate papo disponíveis e o número de usuários associados a cada uma das salas;
3. Entrar em uma sala existente. Um usuário só pode estar associado apenas a uma sala por vez;
4. Sair de uma sala de bate papo;
5. Listar os usuários atuais de uma sala de bate papo. Só é possivel listar os usuários de uma sala em que o usuário esteja associado;

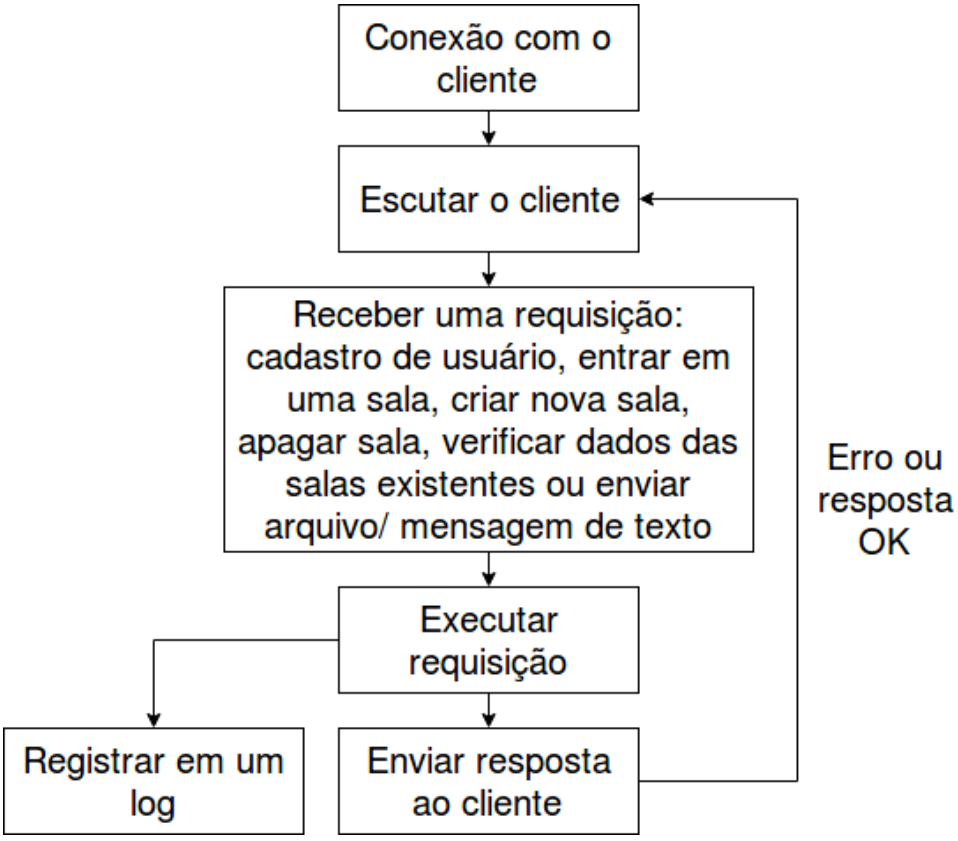

Figura 2: Diagrama do funcionamento do lado servidor da aplicação

6. Criação de salas de bate papo públicas (qualquer usuário pode se associar);
7. Apagar uma sala caso o número de usuários associados a ela seja nulo;
8. Enviar arquivos e messagens para a sala de bate papo a qual o usuário esteja associado. O arquivo/mensagem enviado deverá ser apresentado no terminal de todos os usuários associados a sala;

Cada uma dessas funcionalidades será explicada em detalhes nas próximas subsecções.

Além disso, o desenvolvimento da funcionalidade de criação de salas privadas é definido como uma pontuação extra equivalente a **1 (um) ponto** somado à nota do trabalho. Para obter esse ponto, é necessário que a aplicação atinja as expectativas definidas na secção 4.1 deste roteiro.

2.1 Cadastro de usuários

Após a conexão entre o cliente e o servidor, deve surgir um menu para que o usuário entre no sistema (cadastre-se durante aquela sessão). Nas salas de bate-papo tradicionais, as informações do usuário não são armazenadas entre sessões, mas, caso deseje, o aluno pode implementar uma função para salvar as informações dos usuários de forma persistente entre sessões (utilizando métodos de banco de dados ou apenas cadastro em arquivo) sem adição de ponto extra a sua nota. Independente da escolha feita pelo aluno (manter as informações entre sessões ou não), é necessário que haja algum tipo de **registro dos usuários** acessando o servidor enquanto este estiver ativo.

Assim, o "Cadastro de usuários" no lado cliente consiste em pedir que o usuário insira o nome que será utilizado na sala de bate-papo para que ele tenha acesso ao sistema e, no lado servidor, consiste em salvar esse nome em um arquivo de texto (ou em um banco de dados) e associá-lo ao cliente específico que escolheu o nome para que ele seja utilizado durante a sessão que está aberta (ou de forma persistente para sessões futuras).

Após o usuário se logar no sistema, deve ser mostrado um menu com as opções de requisições a serem executadas. Essas requisições são:

- Buscar informações sobre as salas
- Entrar em e sair de uma sala existente
- Criação de salas públicas e privadas
- Apagar salas

Note que, logo depois do cadastro, a requisição de envio de arquivo ou de mensagem de texto não deve ser possível, uma vez que o usuário não selecionou uma sala para entrar.

2.2 Entrar em e sair de uma sala existente

No lado servidor da aplicação, a entrada em e saída de salas existentes consiste em, respectivamente, incluir ou remover a associação feita entre o usuário e a sala. Ou seja, caso um usuário entre em uma sala, ele passa a poder receber mensagens de todos os usuários cadastrados naquela sala e, caso ele saia da sala, ele não pode mais receber essas mensagens. Quando um usuário sair de ou entrar em uma sala, os arquivos armazenados no servidor (ou o banco de dados do servidor) devem ser atualizados de forma a conter a quantidade correta de usuários em cada sala bem como a informação sobre em que sala está cada um dos usuários cadastrados. Caso contrário, mesmo que o usuário saia da sala, ele continuará recebendo mensagens dos demais usuários cadastrados.

No lado cliente, quando um usuário entrar na sala, deve aparecer, para todos os integrantes daquela sala, a mensagem de que aquele usuário específico entrou na sala e, quando um usuário sair da sala, uma mensagem dizendo que aquele usuário deixou a conversa deve aparecer, também, para todos os integrantes da sala. Ou seja, caso um usuário "Maria" entre na sala "BT 108", a mensagem "Maria entrou na sala" deve surgir para todos os usuários logados na sala "BT 108". De forma análoga, quando Maria sair da sala, a mensagem "Maria saiu da sala" deve ser transmitida para todos os integrantes da sala "BT 108". Para Maria, mensagens do tipo "Bem-vinda à sala BT 108" ou "Você saiu da sala BT 108" também são necessárias.

2.3 Buscar informações sobre as salas

Esta funcionalidade consiste em, no lado do cliente, fornecer os nomes das salas existentes na sessão em curso, o número de pessoas que estão dentro daquela sala virtual e o tipo de acesso à sala (se é pública ou privada). Já, no lado do servidor, essa funcionalidade se resume a buscar essas informações armazenadas em memória e enviá-las à aplicação cliente.

Lembre-se de que, durante o cadastro dos usuários logados na sessão, é necessário associar os nomes dos usuários à cada aplicação cliente (socket) e também à sala em que esse usuário se encontra (se for o caso).

2.4 Criação de salas públicas

Para criar uma sala, o lado cliente deve requisitar sua criação mediante a apresentação de um nome ou outro identificador para essa sala. O lado servidor é responsável, então, por armazenar, em seus registros, o nome da sala e a quantidade de usuários que estão nela. Além disso, a sala criada fica disponível para que seja selecionada por quaisquer usuários que desejam entrar nela.

2.5 Apagar salas

A função de apagar salas existentes só pode ser permitida quando a sala estiver vazia (o número de usuários dentro dela deve ser nulo). Assim, no lado cliente, é requisitado que a sala seja deletada e, no lado servidor, o registro da sala é deletado apenas caso seu número de integrantes seja nulo. Se o número de integrantes não for nulo, o usuário deve ser avisado que não foi possível deletar a sala.

2.6 Envio de arquivos e de mensagens instantâneas de texto

O envio de arquivos e de mensagens de texto deve ocorrer apenas quando o usuário estiver associado a uma sala. Ao enviar arquivos ou mensagens de texto, todos os usuários logados na mesma sala devem receber o arquivo, então, diz-se que o envio de pacotes ocorre de forma *broadcast* dentro de uma lista de usuários. Esse envio é controlado pelo servidor, que detém a informação sobre quais usuários estão logados em cada uma das salas existentes.

O envio de mensagens deve ocorrer de forma que as mensagens apareçam diretamente na tela dos usuários presentes na sala, ou seja, no terminal da aplicação cliente. Já o envio de arquivos pode ocorrer de forma que os arquivos sejam transferidos para o diretório das aplicações clientes que os recebem. Ou seja, o servidor fica responsável por receber um arquivo de um cliente A e por enviar esse arquivo para os clientes B, C, D, F e G que estejam logados na mesma sala. Da mesma forma, uma mensagem de texto enviada pelo cliente A deve ser reproduzida nos terminais dos clientes B, C, D, F e G. A figura 3 mostra um esquemático do fluxo de informação dentro de uma mesma sala de bate-papo.

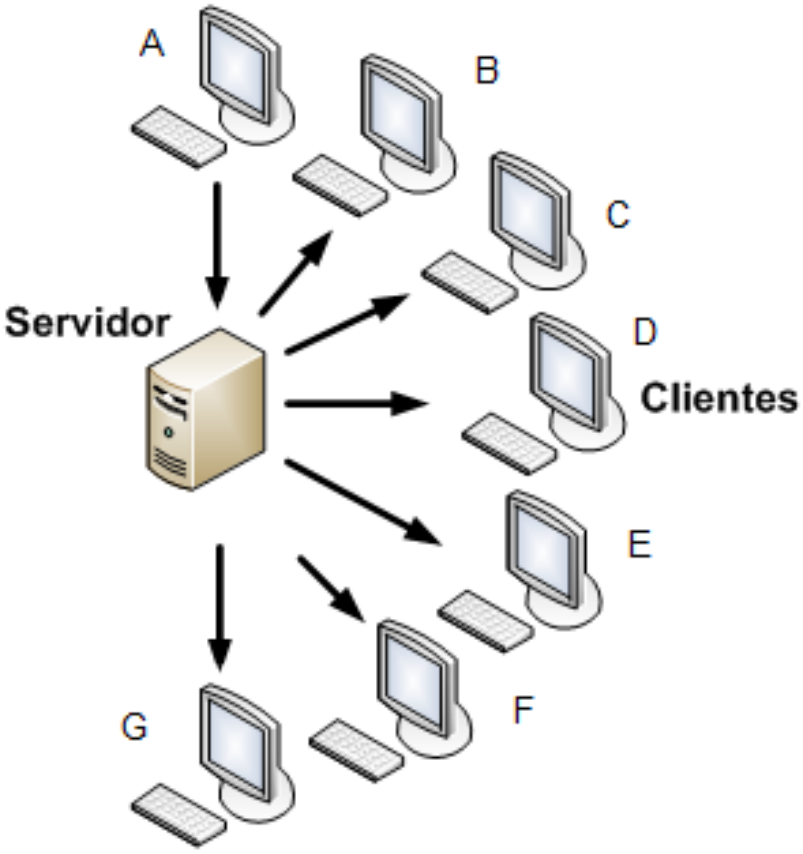

Figura 3: Esquemático do fluxo de informação em uma sala de bate-papo em que A envia um arquivo para os demais integrantes da sala

3 Considerações Importantes

Como a comunicação em salas de bate-papo envolve diversos usuários (clientes) que podem realizar requisições (envio de arquivos/mensagens, entrar em salas, apagar ou criar salas) simultâneas, o servidor precisa ser **capaz de atender essas múltiplas requisições** (vindas de máquinas diferentes) ao mesmo tempo. Dessa forma, a utilização de *threads* torna-se indispensável para a realização deste trabalho.

Deixa-se claro que **não é necessário** que o trabalho possua interface gráfica. Se for de desejo do aluno, a implementação de tal interface pode ser feita, porém, **nenhuma pontuação extra será associada a tal interface** dado que o objetivo do trabalho não é a implementação desta. É necessário, porém, que haja uma interface mínima de texto no lado do cliente, para que um usuário possa comunicar ao servidor o que ele deseja fazer.

O servidor deverá ser capaz de armazenar as requisições requisitadas em um formato de log, que é um arquivo texto contendo as seguintes informações: data e hora da requisição, o endereço IP do cliente que realizou a requisição, tipo de requisição (cadastro, entrar em uma sala, apagar uma sala, envio de mensagem de texto ou de arquivos, etc) e a resposta dada pelo servidor à requisição efetuada. O arquivo de log é muito importante em sistemas computacionais, uma vez que ele permite a identificação de erros do sistema e ele mantém um registro para fins de segurança. No caso de requisições do tipo "envio de mensagem de texto" ou "buscar informações sobre as salas", o conteúdo da mensagem de texto e as informações sobre as salas virtuais **não devem** ser armazenados no log.

4 Avaliação

A avaliação consiste em 2 etapas:

- Relatório de desenvolvimento do projeto.
  - Apresentação detalhada sobre o projeto, que deve ser desenvolvido a partir das especificações deste roteiro;
  - Apresentação de problemas encontrados, bem como explicações sobre o porquê dos problemas e sobre suas possíveis soluções;
  - O comportamento do protocolo da camada de aplicação desenvolvido deve estar explicado na forma de uma máquina de estados finitos;
  - Deverão ser adicionados *Screenshots* e explicações a respeito do funcionamento das funções (ou métodos) implementadas;
  - Podem ser adicionados *links* de vídeos hospedados na internet demonstrando o funcionamento do projeto;
  - Não serão aceitos trabalhos enviados fora do prazo;
- Apresentação em sala de aula.
  - Trabalhos não apresentados não serão considerados;
  - As apresentações devem ter duração de no máximo **10 minutos**;
  - As datas de realização das apresentações serão definidas e informadas via Moodle.

4.1 Pontuação extra: criação de salas privadas

A criação de salas privadas é análoga à criação de salas públicas definidas na secção 2.4. A única diferença entre esses dois tipos de sala é que, na criação de salas privadas, é necessário definir uma senha, que será estritamente necessária para que outros usuários possam entrar nesse tipo de ambiente. Ou seja, para que seja garantido o ponto extra, é necessário que a aplicação seja capaz de:

- Criar salas privadas, isto é, protegidas por uma senha;
- Permitir que usuários entrem nestas salas mediante o fornecimento dessa senha;
- Permitir que usuários troquem arquivos e mensagens de texto normalmente quando estiverem dentro dessas salas (da mesma forma que nas salas públicas);
- Permitir que usuários saiam destas salas (da mesma forma que nas salas públicas);
- Permitir que as salas privadas sejam deletadas mediante a apresentação da mesma senha de acesso.

5 Observações

As seguintes observações deverão ser consideradas pelos alunos que forem fazer o trabalho:

- Este é um trabalho INDIVIDUAL;
- Código copiado da Internet ou do colega é considerado COLA. Trabalhos copiados levarão nota zero, podendo o aluno ser reprovado na disciplina!
  - Ou seja, **não copie o trabalho**!
- O trabalho deve ser **OBRIGATORIAMENTE** executável na plataforma GNU/Linux;
- É responsabilidade do aluno verificar se o código desenvolvido compila e executa corretamente na plataforma GNU/Linux;
- A linguagem de programação na qual o projeto será desenvolvido fica a critério do aluno;
- Será atribuida nota **NULA** aos códigos que não compilam!
- É responsabilidade do aluno se certificar que o arquivo, no formato ZIP, contendo todo projeto (código e relatório) foi corretamente armazenado no Moodle. Ou seja, após a submissão, faça o *download* do seu projeto e descompacte o arquivo para ter certeza que ele está correto;
- Códigos copiados serão considerados "cola"e todos os alunos envolvidos ganharão nota zero;
- É necessário incluir referências no trabalho escrito. Cite a origem de seus dados!
- O relatório deverá ser entregue **OBRIGATORIAMENTE** no formato de arquivo PDF;
- Dúvidas sobre o trabalho deverão ser tiradas **utilizando a plataforma virtual moodle**. Desta forma, dúvidas comuns poderão ser respondidos uma única vez para toda a turma.
- Será considerada como parte da avaliação a organização do código. Comentários coerentes e não óbvios devem ser incluídos.
- O prazo definido para submissão do trabalho é FIRME e já contempla possíveis indisponibilidades do Moodle. Não deixem para submeter o seu projeto na última hora. O sistema está configurado para vocês poderem fazer múltiplos *uploads*.
  - E-mails com o projeto anexado enviados ao professor ou à monitora serão automaticamente descartados.

Referências

[1] Lauren Goode. "**Messenger and WhatsApp process 60 billion messages a day, three times more than SMS**". [Online] Disponível em: "`https://www.theverge.com/2016/4/12/11415198/facebook-messenger-whatsapp-number-messages-vs-sms-f8-2016`". Acesso em: 28 de Agosto de 2018.

[2] Brown Woolley Caterpillar Holding Company. "**Talkomatic**". [Online] Disponível em: "`http://talko.cc/`". Acessado em: 28 de Agosto de 2018.