

SCC 204 – Programação Orientada a Objetos

Trabalho 3

Este trabalho é o último referente ao desenvolvimento do Comunicador Instantâneo.

As funcionalidades que importam!!! Não será utilizada correção automática!!!

A modelagem poderá ser alterada pelos alunos conforme sua necessidade. Terão, portanto, liberdade para tomar decisões.

Ele será dividido em duas principais partes:

1) Servidor – Implemente um servidor utilizando Sockets para receber conexões de clientes. Este servidor será responsável por receber uma mensagem de um cliente e encaminhar para seu destino. O destino pode ser outro usuário ou um conjunto (grupo) de usuários;

2) Cliente – O cliente deve ser implementado a fim de autenticar-se no servidor e encaminhar mensagens para o servidor. A cada mensagem que o cliente envia ao servidor, ele informa qual é o usuário destino ou grupo destino da mensagem. Quando for um grupo, deve-se encaminhar a mensagem para todos os usuários associados a aquele grupo. Dois tipos de mensagem devem ser suportados:

2.1) Texto – uma mensagem texto qualquer

2.2) Arquivo – transferência de um arquivo para o destinatário (um usuário ou um grupo)

O cliente poderá enviar os seguintes comandos para o servidor:

- **creategroup** nomegrupo – o cliente irá encaminhar o comando **creategroup** para o servidor seguido de um nome de grupo, o qual deverá ser criado;
- **dropgroup** nomegrupo – o cliente irá encaminhar o comando **dropgroup** para o servidor seguido de um nome de grupo, o qual deverá ser removido, incluindo seus relacionamentos;
- **adduser2group** nomegrupo emailusuario – o cliente irá encaminhar o comando **adduser2group** para que o servidor adicione o usuário cujo e-mail é definido em emailusuario no grupo nomegrupo. Somente usuários podem ser adicionados em grupos, o que significa que um grupo nunca poderá ser adicionado dentro de outro. Verifique isso!
- **removeuserfromgroup** nomegrupo emailusuario – o cliente irá encaminhar o comando **adduser2group** para que o servidor remova aquele usuário do grupo em questão;
- **authenticate** emailusuario senhausuario – o cliente irá encaminhar o comando o servidor, o qual deve verificar se o usuário existe em seu banco de dados e autenticá-lo. Não deixe o cliente enviar nenhum comando adicional, enquanto o usuário não for autenticado;
- **createuser** emailusuario senhausuario nomeusuario – o cliente enviará este comando o servidor, o qual ficará responsável por criar um usuário em seu banco de dados. Caso o usuário já exista, avise que não poderá ser criado;

- **sendMessage** destino texto – Envia uma mensagem para o servidor, o qual será responsável por encaminhá-la ao destino, seja esse destino outro usuário ou um grupo. No caso de ser um grupo, encaminhe para cada usuário individual nele contido;
- **transferfile** destino nome do arquivo – encaminha o arquivo para o servidor, o qual será responsável por reencaminhá-lo ao destino tal como uma mensagem. Pode ser qualquer arquivo (PDF, Imagem, DOC, etc.);

Lembrete:

- Ao criar um usuário, crie um grupo para ele. Todos seus relacionamentos serão, por padrão, adicionados neste grupo. Isso significa que se o usuário não der um nome de grupo para adicionar um contato, você assume que o grupo padrão é este;
- O servidor deverá utilizar multithreading (threads tal como visto em aula) para que seja capaz de atender múltiplos clientes simultaneamente;
- Deve-se utilizar classes ou funções que manipulem conexões via rede de computadores
- Deve-se utilizar um banco de dados GNU/GPL (<http://www.gnu.org/licenses/gpl.html>) no trabalho tal como PostgreSQL ou MySQL.

Detalhes

- A correção NÃO será feita de maneira automática. Veja abaixo como seu trabalho deverá ser submetido para que isso ocorra;
- É obrigatória SOMENTE a implementação em Java, pois há detalhes adicionais sobre Redes de Computadores que precisam ser compreendidos para utilizar C++ (na verdade os comandos da linguagem C que permitem conexão via rede). No entanto, o aluno pode utilizar o pacote ACDK (<http://acdk.sourceforge.net/>) para criar seu programa em C++. Esse pacote oferece as mesmas classes de Java, contudo implementadas em C++;
- A implementação em Java (obrigatória) valerá nota 10. Caso o aluno também faça a implementação em C++, 10 pontos extra serão eventualmente adicionados, os quais influenciarão diretamente na média da disciplina;
- O cliente pode ser em linha de comando ou usando interface gráfica.

Leituras para adiar os tópicos necessários para o Trabalho

- Documentação do PostgreSQL: <http://www.postgresql.org/docs/9.2/interactive/index.html>
- Documentação do MySQL: <http://dev.mysql.com/doc/refman/5.6/en/index.html>
- Como manipular Bancos de Dados usando Java:
<http://docs.oracle.com/javase/tutorial/jdbc/index.html>
- Programando para Redes de Computadores em Java:
<http://docs.oracle.com/javase/tutorial/networking/index.html>
- Sugiro não começar por C++. Faça primeiramente em Java!!! Depois basta usar o ACDK para implementar em C++.

Para submeter o trabalho

- 1) Acesse o site: <https://ssp.icmc.usp.br>

- 2) Cadastre-se na opção “Cadastro”
- 3) Clique na opção “Login” e entre no sistema
- 4) Clique em “Matricular em disciplina” e selecione “SCC0204 - Programação Orientada a Objetos (Ano: 2013/Semestre: 1- Turma: B)” e matricule-se
- 5) Clique em “Login” novamente e, em seguida, na opção “Listar disciplinas matriculadas”
- 6) Clique na opção “Submeter exercícios” da disciplina
- 7) Escolha o exercício e, em seguida, defina a linguagem de programação como “Zip”
- 8) Selecione seu arquivo zipado
- 9) Submeta seu programa