QuequeApp[1]

Projeto PCD 2015 ISCTE-IUL

Neste projeto pretende-se que seja desenvolvido um sistema de *chat* semelhante ao popular WhatsApp. O objetivo não involve uma aplicação móvel propriamente dita, mas sim uma simulação das mesmas numa aplicação *desktop*. Desta forma, deverá ser possível executar o sistema com uma aplicação servidor que centraliza a gestão do envio de mensagens e múltiplas aplicações clientes a enviar mensagens entre si, via servidor. As funcionalidades exigidas são naturalmente simplificadas em relação ao WhatsApp verdadeiro. O foco do trabalho é na aplicação de programação concorrente e distribuída, sendo que a parte gráfica das aplicações cliente deverá ser considerado um aspeto secundário.

Arquitetura

A solução a desenvolver deverá seguir uma arquitetura cliente-servidor. A Figura 1 ilustra a arquitetura do sistema tendo em conta os principais intervenientes e a funcionalidade elementar (envio e entrega de mensagens). Deverá existir um *servidor* com o qual diversas aplicações *cliente* comunicam para enviar as mensagens entre si, sendo que os clientes nunca comunicam diretamente. Ao ser enviada uma mensagem, o cliente interage com o servidor fornecendo o conteúdo da mensagem e o destinatário. O servidor mantém em memória uma fila de mensagens a entregar, sendo que estas podem não ser entregues instantaneamente no caso de o destinatário não estar *online*. Quanto um utilizador está *online* o servidor entrega as mensagens que lhe são destinadas, e notifica o utilizador remetente que a mensagem foi entregue. Quanto uma mensagem é entregue ao destinatário é apagada do servidor. Desta forma, uma mensagem numa determinada conversa pode ter dois estados possíveis - *enviada* ou *entregue*. Na aplicação cliente esta distinção deve ser clara para o utilizador (p.e. com recurso a ícones). Se for enviada uma mensagem para um utilizador que está *offline*, a aplicação cliente considera a mensagem apenas como *enviada*, e assim que o destinatário recebe a mesma passa a ser considerada *entregue*.

A Figura 1 ilustra um cenário de envio de mensagens entre um utilizador *A* e um utilizador *B*, cada um utilizando a uma instância da aplicação cliente.

[1] como quem diz, a aplicação do "*que é que se passa?*"

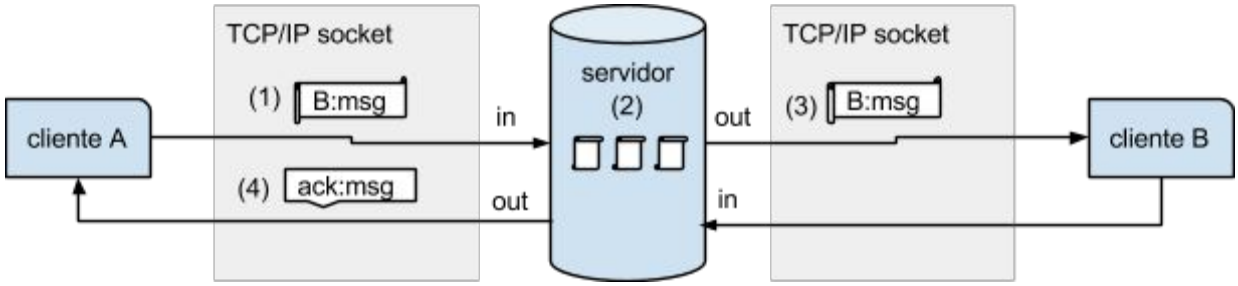

Figura 1. Arquitetura cliente-servidor. Cenário: (1) o utilizador A envia uma mensagem para o servidor, destinada ao utilizador B; (2) a mensagem enviada em guardada em memória no servidor; (3) a mensagem é entregue pelo servidor ao utilizador B; (4) o servidor envia ao utilizador A uma notificação da entrega da mensagem.

Servidor

O servidor deverá aceitar pedidos de conexão por parte de clientes, através de uma ligação TCP/IP num porto com um número bem definido (e do conhecimento dos clientes). Ao ser estabelecida uma ligação, os clientes enviam o nome de utilizador. Para efeitos de simplificação, não é exigido nenhum processo de autenticação dos utilizadores.

Ligações

O servidor deverá funcionar em *multi-threading*, tendo para cada ligação ativa duas *threads*, que designaremos por *in* e *out*:

- ***in*** para receção de mensagens dos clientes (*input stream*)
  Esta *thread* estará bloqueada à espera de mensagens do cliente. As mensagens que o servidor recebe deverão ser guardadas numa estrutura em memória. Quando um dado utilizador tem mensagens novas, caso este esteja *online*, a *thread out* associada à sua ligação deverá ser notificada, por forma a processar a entrega das mensagens.

- ***out*** para entrega de mensagens e notificações aos clientes (*output stream*)
  Esta *thread* estará em espera, tornando-se ativa no momento em que existem mensagens novas ou notificações de entrega de mensagens relativas ao utilizador da ligação.

Execução e consola de monitorização

O servidor deverá poder ser lançado num processo Java normal, por exemplo executando:

```
java QuequeServer
```

A partir do momento que o servidor é lançado está disponível para receber ligações de clientes. Porém, deverá também ser possível interagir com o servidor na consola, de forma a monitorizar o seu funcionamento. A consola do servidor será útil também para efeitos de *debugging*. Deverão estar disponíveis comandos para ser possível:

- listar os utilizadores que se encontram ligados
- listar as mensagens que se encontram em memória para serem entregues
- apresentar informação relativa ao periodo de tempo em que o servidor se encontra *online*, e ao número total de mensagens entregues
- desligar o servidor (terminando o processo)

Quando o servidor é desligado, considere que as mensagens que tinha em memória serão perdidas, e neste caso os clientes que as enviaram nunca serão notificados da sua entrega.

Cliente

As aplicações cliente deverão consistir numa interface gráfica minimalista com duas vistas:

- **Contactos**. Nesta vista deverão ser listados os contactos. Deve mostrar o nome, e quantas mensagens novas por ler. Deverá ser possível:
    - adicionar um novo contacto dado o seu nome de utilizador, o qual passará a constar na lista de contactos. Se um dado utilizador receber uma mensagem de outro utilizador que não está na lista de contactos, este último deve ser automaticamente adicionado à lista.
    - criar um grupo de contactos com base nos contactos individuais existentes, para o qual se podem enviar mensagens; os grupos não são geridos pelo servidor e não é exigido que se implemente um controlo de acesso aos mesmos.

- **Chat.** Nesta vista serão apresentadas as conversas do utilizador com outro utilizador ou grupo. As mensagens da conversa que foram enviadas para o outro utilizador devem ser distinguidas em termos do seu estado (enviada ou entregue), excepto no caso de se tratar de um grupo. Uma mensagem poderá conter uma imagem escolhida pelo utilizador. Quando a aplicação cliente recebe uma mensagem, a mesma deverá ser adicionada à conversa respetiva. Esta vista deverá também ser utilizada para as mensagens de um grupo.

A aplicação deverá ser lançada através de um processo Java, por exemplo da seguinte forma (onde `confdir` é um diretório que contém os dados do utilizador):

```
java QuequeApp confdir
```

A aplicação cliente deverá poder funcionar *offline* (i.e. sem ter uma ligação ativa com o servidor), não sendo desta forma possível enviar mensagens. Caso não seja possível

estabelecer uma ligação ao servidor, o cliente deverá periodicamente (com um intervalo de tempo de poucos segundos) tentar estabelecer a ligação. A partir do momento que o cliente consegue uma ligação, o utilizador passará a poder enviar mensagens.

As conversas de um utilizador deverão ser guardadas no sistema de ficheiros, de forma a que ao iniciar um cliente todas a conversas prévias serão carregadas na aplicação, incluindo o estado das mensagens. O formato de gravação em ficheiro das conversas é livre, mas é exigido que este formato seja textual (por oposição a serialização de objetos). As conversas deverão ser gravadas para os ficheiros à medida que as mesmas evoluem.

Fases, Avaliação, e Entrega

São propostas as seguintes fases de desenvolvimento como metas para a avaliação, e de forma a que seja mais fácil abordar o problema:

Fase 1: Desenvolver uma versão básica da interface gráfica dos clientes, sendo o envio de mensagens simulado de alguma forma (sem comunicar com o servidor).

Fase 2: Desenvolver o servidor, suportando a apenas a entrega de mensagens sem notificações de entrega de mensagens.

Fase 3: Enriquecer a comunicação cliente-servidor de forma a suportar notificação de entrega de mensagens.

Fase 4: Evoluir a solução para suportar envio de mensagens para grupos de utilizadores.

As notas do projeto serão atribuídas de acordo com a realização das fases propostas:

A: completar todas as fases
B: completar as fases (1, 2, 3) ou (1, 2, 4)
C: competar as fases (1, 2)
D: não são cumpridos os requisitos mínimos (reprovação à UC)

Grupos: Cada grupo de trabalho é composto por dois alunos, preferencialmente da mesma turma prática.

Entrega: O projeto desenvolvido deve ser entregue sob a forma de um projeto arquivado usando a funcionalidade de *Export/Archive File* do Eclipse. As entregas serão feitas exclusivamente de forma presencial nas aulas práticas da semana de 7/12/2015, sendo que os grupos deverão comparecer na aula prática onde estão inscritos.