

Sistemas Operativos 2

Trabalho 2015/2016

29/04/16



UNIVERSIDADE
DE ÉVORA

Questionários



Professor:
José Saias

Realizado por:
João Calhau - 31621
José Pimenta - 31677

Índice

Introdução.....	Pag 3
Estrutura do Trabalho.....	Pag 4
Base de Dados.....	Pag 5
Funcionalidades.....	Pag 6
Admin.....	Pag 7
Criar.....	Pag 8
Responder.....	Pag 9
Listar.....	Pag 10
Ver.....	Pag 10
Media.....	Pag 11
Vezes.....	Pag 11
Apagar.....	Pag 12
Build da Aplicação.....	Pag 13
Conclusão.....	Pag 14

Introdução

No âmbito da unidade curricular de Sistemas Operativos II pretende-se, usando as metodologias de comunicação e soluções de Middleware abordadas nas aulas, desenvolver um sistema de processamento de questionários baseado no conceito de cliente-servidor. Esse sistema permite ao cliente a execução de um conjunto de operações sobre uma base de dados que contém questionários criados pelo mesmo (caso tenha privilégios de admin) ou criados por outro cliente. O conjunto de operações consiste então em criar questionários, consultar os que existem, apagar um questionário, obter as perguntas de um questionário, submeter respostas às questões de um questionário, consultar o nº de vezes que um questionário foi respondido e a média do valor respondido em cada questão. No nosso trabalho utilizamos a Invocação Remota de Métodos (javaRMI) que foi extremamente simples de utilizar, permitindo nos abstrair de aspectos de baixo nível e de comunicação.

Estrutura do Trabalho

O trabalho está organizado em 6 classes que em conjunto servem os mais variados propósitos, desde estabelecer ligação à base de dados, assinatura e implementação de métodos remotos até ao menu do cliente onde é feita toda a interação entre o cliente e o servidor, que graças à camada de middleware, é feita de maneira transparente, ou seja, o cliente não necessita saber onde estão localizados os dados.

As classes Pergunta e Questionario são as classes que foram criadas para representar objectos Pergunta e Questionario que ajuda na representação dos dados localmente no cliente quando necessários. Como estamos a utilizar javaRMI precisamos da interface remota OperacoesDB e a classe do objeto remoto OperacoesDBImpl que contém a implementação dos métodos de invocação remotada declarados na interface remota (com métodos para efectuar ações sobre os dados na base de dados). O MenuServer cria o servidor e faz a ligação à base de dados, e o MenuClient que tem o menu para o cliente, conecta ao servidor, e chama métodos declarados na classe OperacoesDB e tem o menu para o cliente poder escolher as várias opções disponíveis.

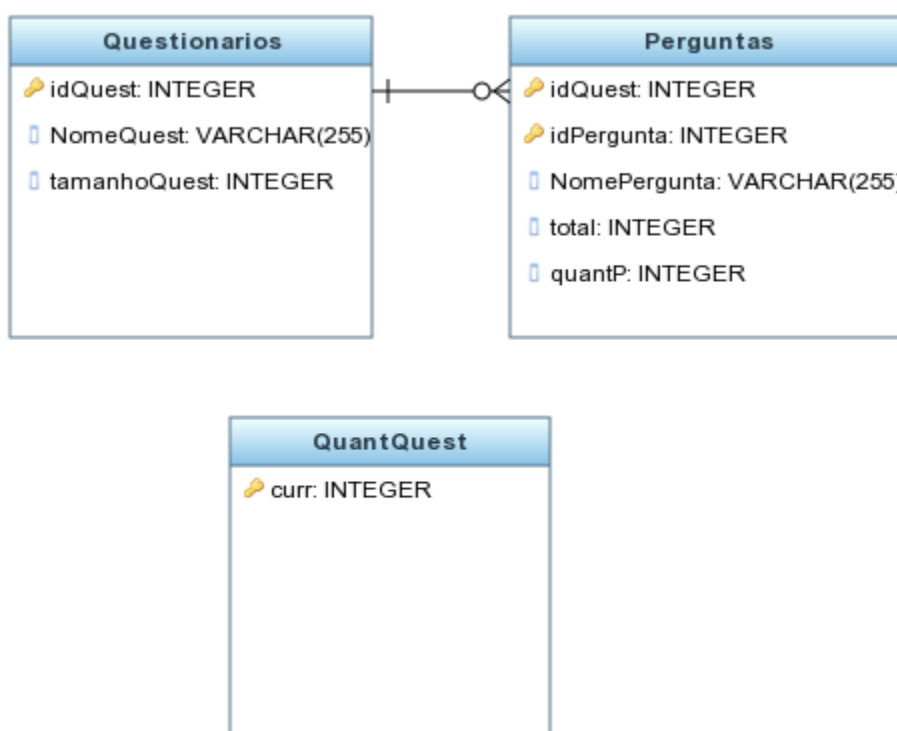
O trabalho tem ainda 2 shell scripts, 1 para o servidor e outro para o cliente, e ainda 1 ficheiro sql com as tabelas necessárias da base de dados para o trabalho poder funcionar.

Base de Dados

A base de dados é composta pelas tabelas Questionarios, Perguntas e QuantQuest. A tabela Questionarios guarda informação com um id único para o questionario em questão, o nome do questionario e a quantidade de questões que este tem.

A tabela Perguntas guarda informação apenas de 1 Pergunta, que é o id do Questionário em que se encontra, o id que representa a ordem pela qual a pergunta deve aparecer no questionário, a pergunta propriamente dita (string), total que contém a soma total das respostas (1 a 10) enviadas por quem respondeu, e quantP que corresponde á quantidade de pessoas que responderam (utilizado sobretudo no cálculo de média e vezes que 1 questionário foi respondido).

Em baixo, o modelo de entidade relacional da base de dados:



Funcionalidades

Tal como referido anteriormente, as funcionalidades do Sistema de Questionários é o seguinte:

Após a execução do MenuServer, aparecem estas mensagens ao utilizador, em que só há 1 comando possível depois, o “exit” para fechar o servidor:

```
Joao-PC/10.2.14.47
ligado á BD
Bound RMI object in registry
servidor pronto
(Para fechar o servidor, basta escrever 'exit')
```

Após a execução do MenuClient, aparecem estas mensagens ao utilizador:

```
Bem vindo a plataforma de Questionarios!
Encontra-se neste momento no Menu.
(Para saber os comandos escreva 'help')
```

E assim com a ajuda do comando ‘help’ o utilizador sabe quais os comandos possíveis para aceder a todas as funcionalidades do sistema:

```
help
Comandos possiveis:
'listar'
'responder'
'ver'
'media'
'vezes'
'admin'
'exit'
```

Admin

Começando por este comando, o acesso a funcionalidades admin permite utilizar comandos tais como 'criar' e 'apagar' questionários. Isto é feito assim deste modo para que não seja qualquer utilizador a utilizar o client a poder apagar ou criar questionários. Após a execução do comando 'admin' é então solicitada uma password e assim obter-se privilégios de administrador.

```
admin
Insira a password:
1234
Tem agora privilegios de Admin.
logout
Tem agora privilegios basicos de cliente.
```

Após a execução do comando 'admin' é assim acessível mais 2 comandos, o 'criar' e o 'apagar' questionários, pralém de poder fazer 'logout' da conta de admin.

```
Comandos possiveis:
'listar'
'reponder'
'ver'
'media'
'vezes'
'criar'
'apagar'
'logout'
'exit'
```

(NOTA: isto é o que acontece quando se tentam executar comandos 'apagar' e 'criar' quando não se encontra logado como admin, e assim para qualquer tipo de mensagem digitada que não seja comando)

```
apagar
Comando nao reconhecido...
criar
Comando nao reconhecido...
```

Criar

Após a execução deste comando (estando o admin logado), é pedido ao utilizador que insira o nome do questionário e se encontra-se satisfeito com o nome inserido. Se estiver satisfeito é pedido para inserir um valor entre 3 e 5, que é a quantidade de perguntas do questionário, seguido pelo pedido das ditas perguntas. No final é perguntado se pretende guardar o questionário ou não.

```
criar
Insira o nome do Questionario:
Questionario de Qualidade
Está satisfeito como nome? y/n
y
Nome aceite!
Insira valor int entre 3 e 5
3
Resposta Aceite
Insira a questão:
Grau de satisfacao de qualidade
Insira a questão:
Grau de satisfacao de entrega
Insira a questão:
Grau de satisfacao no geral
Quer guardar o Questionario? y/n
y
Questionario adicionado!
```

Tabela 'questionarios' da base de dados com questionário e tabela 'perguntas' com as perguntas inseridas através do comando 'criar':

```
l31621=> Select * from questionarios;
idquest |      nomequest      | tamanhoquest
-----+-----+-----
      1 | Questionario de Qualidade |          3
(1 row)
```

```
l31621=> select * from perguntas;
idquest | idpergunta |      nomepergunta      | total | quantp
-----+-----+-----+-----+-----
      1 |          1 | Grau de satisfacao de qualidade |      0 |      0
      1 |          2 | Grau de satisfacao de entrega   |      0 |      0
      1 |          3 | Grau de satisfacao no geral     |      0 |      0
(3 rows)
```


Responder

Após a execução deste comando (estando o admin logado, e sabendo os questionários disponíveis através do comando 'listar'), é pedido ao utilizador que insira o número do questionário, e acontece uma de três coisas:

Caso não haja questionários guardados:

```
responder
Nao existem questionarios guardados!
```

Caso queira responder e o questionário se encontre presente, é pedido que insira um valor entre 1 e 10 às perguntas do questionário:

```
responder
Qual o numero do questionario a que pretende responder?
Insira valor int entre 1 e 1
1
Resposta Aceite
Grau de satisfacao de qualidade
Insira valor int entre 1 e 10
10
Resposta Aceite
Grau de satisfacao de entrega
Insira valor int entre 1 e 10
10
Resposta Aceite
Grau de satisfacao no geral
Insira valor int entre 1 e 10
10
Resposta Aceite
```

Existe ainda o caso de tentar responder a um questionário que já esteve presente, mas que foi apagado, e nesse caso é enviada 1 mensagem para o cliente a dizer que o questionário já foi apagado (isto porque cada questionário tem 1 número que o identifica, e esse número nunca mais pode ser utilizado por outros questionários), e esta mensagem aplica-se a quase todos os comandos seguintes neste caso específico.

Tabela 'perguntas' após o questionário ter sido respondido, em que verificamos que 1 pessoa respondeu ao questionário até ao momento (quantP), e que o total de valores inserido é 10 em cada (valores inseridos por este utilizador, visto só 1 pessoa ter respondido até ao momento).

```
l31621=> select * from perguntas;
 idquest | idpergunta | nomepergunta | total | quantp
-----+-----+-----+-----+-----
      1 |          1 | Grau de satisfacao de qualidade |    10 |      1
      1 |          2 | Grau de satisfacao de entrega |    10 |      1
      1 |          3 | Grau de satisfacao no geral |    10 |      1
(3 rows)
```

Listar

Após a execução deste comando, uma de duas coisas pode acontecer, ou não existem neste momento questionários guardados (isto quando se corre 1 client pela primeira vez, ou porque todos os questionários que existiam foram apagados):

```
listar
Nao existem questionarios guardados!
```

ou então acontece o seguinte, em que lista os questionários disponíveis:

```
listar
1 - Questionario de Qualidade
```

Ver

Após a execução deste comando, uma de duas coisas pode acontecer, ou não existem neste momento questionários guardados (isto quando se corre 1 client pela primeira vez, ou porque todos os questionários que existiam foram apagados):

```
ver
Nao existem questionarios guardados!
```

ou então acontece o seguinte, em que verifica as perguntas do questionário escolhido:

```
ver
Qual o numero do questionario que pretende ver as questoes?
Insira valor int entre 1 e 1
1
Resposta Aceite
Grau de satisfacao de qualidade
Grau de satisfacao de entrega
Grau de satisfacao no geral
```

Media

Após a execução deste comando, uma de duas coisas pode acontecer, ou não existem neste momento questionários guardados (isto quando se corre 1 client pela primeira vez, ou porque todos os questionários que existiam foram apagados):

```
media
Nao existem questionarios guardados!
```

ou então acontece o seguinte, em que verifica as perguntas do questionário escolhido:

```
media
Qual o numero do questionario que pretende ver a media das questoes?
Insira valor int entre 1 e 1
1
Resposta Aceite
Grau de satisfacao de qualidade - 10.0
Grau de satisfacao de entrega - 10.0
Grau de satisfacao no geral - 10.0
```

Vezez

Após a execução deste comando, uma de duas coisas pode acontecer, ou não existem neste momento questionários guardados (isto quando se corre 1 client pela primeira vez, ou porque todos os questionários que existiam foram apagados), tal como se mostrou anteriormente, ou então vê o número de vezes que um questionário foi respondido:

```
vezez
Qual o numero do questionario que pretende ver quantas vezes foi respondido?
Insira valor int entre 1 e 1
1
Resposta Aceite
0 Questionario Questionario de Qualidade foi respondido 1 vez(es)!
```

Apagar

Após a execução deste comando, uma de duas coisas pode acontecer, ou não existem neste momento questionários guardados (isto quando se corre 1 client pela primeira vez, ou porque todos os questionários que existiam foram apagados):

```
apagar
Nao existem questionarios guardados!
```

ou então acontece o seguinte, em que apaga um questionário:

```
apagar
Qual é o numero do questionario que quer apagar?
Insira valor int entre 1 e 1
1
Resposta Aceite
Quer mesmo remover o questionario Questionario de Qualidade? y/n
Insira y/n!
y
Questionario removido!
listar
Nao existem questionarios guardados!
```

Tabelas 'questionarios' e 'perguntas', após a remoção do questionário 1 que se encontrava anteriormente na base de dados:

```
131621=> Select * from questionarios;
 idquest | nomequest | tamanhoquest
-----+-----
(0 rows)

131621=> select * from perguntas;
 idquest | idpergunta | nomepergunta | total | quantp
-----+-----+-----
(0 rows)
```

Build da Aplicação

Para correr o serviço de questionários é necessário correr os SH que funcionam da seguinte maneira:

Para o servidor tem de se correr

```
bash Server.sh PORT IP_DATABASE DATABASE USER PWD
```

exemplo utilizado nos testes

```
bash Server.sh 9001 alunos.di.uevora.pt l31621 l31621 12345
```

Para o cliente tem de se correr

```
bash Client.sh IP PORT
```

exemplo utilizado nos testes

```
bash Client.sh localhost 9001
```

Conclusão

Em conclusão achamos que conseguimos cumprir os objetivos do trabalho com sucesso. Foram necessários vários testes exaustivos tanto na ligação entre servidor e cliente, testes estes que fizemos com servidor e cliente em máquinas diferentes na rede eduroam, com servidor numa máquina em rede eduroam e com o cliente na máquina alunos.di.uevora.pt, o único teste que não foi um sucesso foi o teste com o servidor na máquina alunos.di.uevora.pt e com o cliente na rede eduroam. O teste com servidor e cliente na máquina alunos.di.uevora.pt foi bem sucedido. Fizemos ainda testes no acesso à base de dados. No final o trabalho ficou mais ou menos a correr como idealizámos no início. Apenas tivemos algumas dificuldades a meio do trabalho, pois inicialmente criámos o trabalho apenas com cliente e servidor e a guardar os dados do lado do servidor em disco e só depois mudámos o trabalho para poder funcionar com base de dados, o que fez com que tivéssemos de alterar substancialmente partes do código de modo a poder funcionar com a base de dados (sobretudo devido à existência de ArrayList, entre outros). Tendo isto em conta, podemos assumir que o balanço final foi positivo e que o que aprendemos na realização deste trabalho foram coisas que podem vir a fazer falta num futuro, sendo assim bom este balanço final.