Relatório do Trabalho Prático de Sistemas Operativos II

José Serra (nº 33289) Carlos Valente (nº 33298)

27 Abril, 2019

1 Introdução

No âmbito da disciplina de Sistemas Operativos 2, foi nos proposto o desenvolvimento de um serviço de resposta a questionários, utilizando implementações de servidor e cliente.
Os dados destes questionários devem ser guardados do lado do servidor de forma persistente numa base de dados.

2 Compilação/Acesso

1. MakeFile;
2. svlocal.properties;

2.1 MakeFile

Criámos um ficheiro makefile que contém 4 operações:

- Make all:

Serve para compilar todas os ficheiros.java inseridos na pasta "src" do projeto, sejam eles do servidor ou do cliente.
Contém também dois comandos para copiar respetivamente os ficheiros .properties para a recém criada diretoria build/classes.

- make clean:

Remove a pasta build do projeto e todas as suas subpastas incluindo a pasta classes, que contém todos os ficheiro.class compilados através do comando make all.

- make Server:

executa o servidor.

- make Client:

executa o cliente.

2.2 svlocal.properties

Ficheiro com os parâmetros para ligar à base de dados que contêm as seguintes credenciais:

filename |	svlocal.properties(aceder a alunos)	svlocal.properties(aceder localhost)
port |	alunos.di.uevora.pt	localhost
bdname |	l33289	quest
user |	l33289	user1
pw |	sqldoze	umaPass

3 Classes

1. Cliente.java;
2. Server.java;
3. Questionario.java;
4. Pergunta.java;
5. QuestionarioPergunta.java;
6. QuestPergImpl.java;
7. Database.java;

3.1 Cliente.java

Classe main do projeto, que contém o menu e consequentemente redireciona o utilizador para todas as operações do serviço.

3.2 Server.java

Classe que contém as informações relativas ao servidor, registos, e declaração do objeto remoto.

3.3 Questionario

Classe que contém o construtor e todos os métodos relativos aos questionários tais como gets e sets.

3.4 Pergunta

Tal como a Classe anterior, contém o construtor e todos os métodos relativos ás Perguntas.

3.5 QuestionarioPergunta

Interface remota que contém os métodos da Classe QuestPergImpl.

3.6 QuestPergImpl

Classe que implementa os métodos da interface remota. Esses métodos por sua vez, retornam valores da Class Database. É também nesta Classe que é criado o objeto do tipo Database, de forma a fazer a ligação à base de dados.

3.7 Database

A classe database, é a nosso ver a mais importante e contém a implementação de todos os métodos que dizem respeito a Querys da Base de dados, ou seja, todos os métodos que envolvem statements e inserções/consultas.
Esses métodos/operações são os seguintes:

public boolean querySetup() :
-método que lê um ficheiro .sql e introduz todas as suas querys na base de dados, de maneira a criar uma nova, ou a repôr os valores da mesma.

public int queryExisteQuestionario(String id) :
-método que indica se o questionário que pretendemos criar já existe ou não .Efectuamos uma seleção de uma linha na tablela dos questionarios utilizando o "id" que pretendemos criar e se essa linha for nula é retornado um 0 caso contrário se esse questionário existir é retornado 1.

public ArrayList queryListarQuestionarios() :
-método que através de um resultSet, seleciona todos os 'id' da tabela questionários, e os retorna num arrayList.

public void queryCriarQuestionario() :
-método que recebe um id e um numero de perguntas como argumento, e faz uma inserção na BD criando assim um novo questionário.

public void queryApagarQuestionario() :
-método que recebe um id de um questionário e por sua vez faz DELETE de todos valores que da base de dados que dizem respeito a esse id.

public void queryAdicionarPergunta() :
-método que recebe um id, uma pergunta e um numero, que por sua vez INSERE respetivamente na base de dados, na table pergunta. Ou seja, adiciona perguntas a um questionário.

public void queryAdicionarResposta()
-método que recebe um id e uma resposta, que por sua vez INSERE respetivamente na base de dados, na table resposta. Ou seja, adiciona respostas ás perguntas de um questionário.

public void queryIncrementaVezesRespondido() :
-método que como o nome indica, através de um id que recebe como argumento, incrementa a o valor "vezesrespondido" da table questionários, que corresponde a todas as vezes que um questionário teve submissões de respostas ás suas perguntas.

public Questionario queryDevolveVezesRespondido() :
-método onde é criado um objeto do tipo questionário e que através de uma operação select na base de dados, e de um resultSet, devolve todas as informações referentes ao id que recebe como argumento na forma desse mesmo objeto do tipo Questionário. De forma a que no cliente seja possível imprimir o numero de vezes a que um questionário foi respondido.

public ArrayList queryDevolvePerguntas() :
-método que devolve todas as perguntas de um questionário cujo id recebe por argumento, através de uma consulta à base de dados.

public ArrayList queryDevolveRespostas() :
-método que devolve todas as respostas de um questionário cujo id recebe por argumento, através de uma consulta à base de dados e as insere num arrayList para que possam ser utilizadas em operações pelo cliente.

4 Conclusão

Consideramos que o trabalho se encontra em bom funcionamento tanto no Servidor como no Cliente.O cliente tem acesso ao objecto remoto sem exceções e consegue executar todas as suas operações. Da parte do servidor, a base de dados encontra-se funcional, bem como todas as operações que lhe dizem respeito. Conseguimos também a ligação sem problemas ao servidor alunos.di.uevora.pt. Ficaram por fazer apenas algumas verificações de user inputs no Client, de forma a evitar algumas exceções.