



## Trabalho de Sistemas Operativos II



Trabalho realizado por :  
Miguel Azevedo nº36975  
Vasco Crespo nº37913

Professor:  
José Saias

## Introdução

O nosso trabalho tinha como objetivo representar informação de questionários e respectivas respostas, de forma a auxiliar uma empresa de estudos de mercado a reduzir o tempo de processamento de questionários.

Para isto utilizamos uma base de dados para guardar os diferentes questionários, que, a partir de operações efetuadas pelo utilizador serão modificados. Estas operações incluem a criação e modificação de qualquer questionário, a consulta das suas perguntas e das vezes que foram respondidos, a realização dos mesmos e ainda a média do valor respondido em cada questão.

## Desenvolvimento

Cada questionário pode constituir três, quatro ou cinco perguntas. Tendo em conta isto, decidimos criar cinco tabelas.

A primeira denominada questionários que contém um ID único de cada questionário, o nome do questionário respetivo e o número de perguntas que cada um tem. De seguida, as outras próximas tabelas denominadas questionario3, questionario4 e questionario5 contêm todas as perguntas dos questionários dependendo do número de perguntas dos mesmos. Por último temos uma tabela denominada addUnique que contém um id único para cada resposta efetuada a qualquer questionário e o id do questionário a que pertence essa resposta.

Para as diferentes operações ditas no enunciado, foram criados métodos para resolver cada questão:

- *int SaberN* (String id) - este método irá procurar na tabela questionários, o questionário com o ID dado como input e irá retornar o número de perguntas associada a esse ID.
- *int addUnique*(String id) – quando o cliente quer responder a uma questionário este método é invocado recebendo o id do questionário a ser respondido e adicionando este ID e um outro ID único (que representa a resposta realizada pelo cliente) à tabela *addUnique*.
- *String CriarQuest* (String id, String nome, int numero, String [] perguntas) – este método irá adicionar a informação dada (ID, nome e numero) à tabela questionários. O array *perguntas* irá conter as perguntas inseridas pelo utilizador e irá adicioná-las a tabela questionarioX em que X é o numero de perguntas do questionário juntamente com o *ID* de cada questionário. Posto isto, criará também uma tabela na Base de Dados com nome *ID* e contendo colunas para as respostas às perguntas do questionário. Caso o utilizador insira um *ID* já

existente na tabela questionários, o programa retornará uma *String* “*ID do questionário já existente!*”, senão irá retornar “*Questionario criado com sucesso*”.

- *ArrayList <String> ConsultQuest()* – este método tem como objetivo retornar todos os questionários inseridos na Base de Dados. Para isso irá aceder à tabela *questionários* e irá enviar para uma *arrayList* que irá ser retornada, os elementos da tabela (*ID, nome, numero de perguntas*).
- *String ApagaQuest (String id, int numero)* – este método tem como objetivo, apagar um questionário da Base de Dados. Para isso irá aceder à tabela *questionários*, apagar o questionário com o *ID* introduzido, apagar também a tabela com o nome *ID* e todas as linhas da tabela que tenham o nome *ID*. O questionário também é apagado da tabela *questionarioX* em que *X* é o número de perguntas.
- *ArrayList <String> Questions (String id)* – este método tem como objetivo retornar todas as perguntas do questionário *ID*. Para isso usamos o método *SaberN(String id)*, para acedermos à tabela com o numero de perguntas do questionário, onde iremos adicionar a uma *arrayList* que irá ser retornada.
- *String AnswerQuest (String id, int numero, int [] respostas)* – este método irá aceder à tabela com o número de perguntas respetivo e irá inserir as respostas contidas no array respostas.
- *int VezesQuest (String id)* – este método irá retornar o número de vezes que o questionário foi respondido. Para isso irá aceder à tabela criada do *questionario ID* e irá obter o número de vezes que o questionário foi respondido.
- *int [] Media (String id)* – este método irá aceder as respostas de cada pergunta relacionada ao *ID*, e irá fazer a soma das mesmas. Após isso, é chamado o método *VezesQuest(Id)* para ser efetuada a média.

## Conclusão

Ao longo deste trabalho, foram encontradas algumas dificuldades, tais como, a ligação ao Servidor e na utilização do java RMI. Mas ao trespassarmos estas dificuldades percebemos também a importância da utilização do java RMI, e da comunicação entre servidor-cliente e também da base de dados em si.