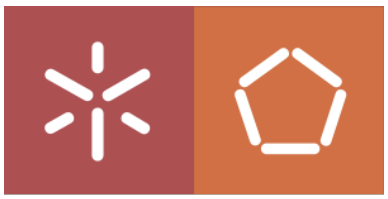

Universidade do Minho
Escola de Engenharia

Licenciatura em Engenharia de Telecomunicações e Informática

Sistemas Distribuídos

Gestão de Frota

Relatório de projeto

Camila Pinto - a91687

Eduarda Dinis - a95573

Barbara Fonseca - a97778

Conteúdo

1 Introdução

O presente relatório está a ser desenvolvido no âmbito da Unidade Curricular de Sistemas Distribuídos do 1º semestre do 3ºano do curso de Licenciatura em Engenharia Telecomunicações e Informática. O objetivo principal do trabalho prático é implementar uma plataforma de gestão de uma frota de trotinetes elétricas, sob a forma de um par cliente-servidor em Java, utilizando sockets e threads. Para a realização desse projeto foram utilizados vários conceitos aplicados nos guiões das aulas práticas e teóricas.

2 Servidor

O objetivo do servidor é responder aos pedidos do cliente. Para permiter um melhor tratamento dos pedidos de cada cliente, utilizamos o **worker** atráves das funcionalidades da classe **ServerWorker**.

3 Cliente

Na implementação do Cliente foi considerada a porta "12345". Implementou-se também o **multi-threaded** refletida também na classe **Demultiplexer** e, por consequência, na classe **TaggedConnection**. O Cliente efetua funcionalidades desde da autenticação até ao log-out do programa.

4 Conexão Servidor - Cliente

Para implementar a conexão servidor-cliente utilizou-se as classes **TaggedConnection** e a classe **Demultiplexer**.

4.1 TaggedConnection

A classe **TaggedConnection** tem a função de gerir as mensagens entre o cliente e o sevidor, pelo etiquetamento dessas mensagens . Para isso, é necessário diferidos metodos para o envio e a receção das mensagens do cliente e do servidor. Deste modo, as classes **Frame-**

Cliente e **FrameServidor** que armanezam as mensagens recebidas pelo servidor e cliente, respetivamente.

4.2 Demultiplexer

A classe **Demultiplexer** tem a função de agrupar todas as mensagens trocadas entre o servidor e o cliente e distribuí-las entre as threads que receberam i pedido.

5 Funcionalidades

As funcionalidades do Cliente são: efetuar o registo e a autenticação dos utilizadores (tag 0 e tag 1), pesquisar a trotinete mais próxima (tag 2), reservar uma trotinete (tag 3), estacionar uma trotinete (tag 4), e efetuar o log-out do programa (tag 5). Para cada funcionalidade irá ser enviado um pedido ao servidor de uma thread específica.

5.1 Autenticar e Registar Utilizador

Cliente - o utilizador escolhe se quer fazer a autenticação de um conta já existente ou se quer registar uma nova conta, e paraa ambos os métodos sã pedidos o nome de Utilizador e a palavra passe.

Servidor - se o utilizador escolher fazer um novo registo verifica-se se já existe, se não é guardado o nome e a palavra passe. Se o utilizador escolher autenticar-se verifica-se a correspondência do nome e da palavra passe. É enviado em ambos os casos uma mensagens de confirmação ou de erro para o cliente.

Cliente - o cliente recebe a mensagem do servidor, e se se confirmar a autenticação ou o registo, o utilizador tem acesso ás funcionalidades.

5.2 Pesquisar a trotinete mais próxima

Cliente - Pede as coordenadas da localização **x** e **y** ao utilizador.

Servidor - o servidor calcula a distância o utilizador com a lista das trotinetes, e envia ao cliente a trotinete mais próxima da localização do utilizador e o respetivo código de reserva.

Cliente - o cliente devolve ao utilizador a trotinete mais próxima.

5.3 Reservar trotinete

Cliente - o cliente solicita o código de reserva da trotinete que quer reservar para enviar ao servidor.

Servidor - o servidor procura a trotinete que contém o código de reserva que o utilizador adicionou. A seguir, verifica se existe a trotinete na lista de reservas, se exister na lista de reservas é enviada uma mensagem de erro para o cliente, se não é enviada uma mensagem de confirmação para o cliente, alterado o estado de reserva para ocupado e adicionada a reserva á lista com a hora de reserva.

Cliente - o cliente recebe a mensagem enviada pelo servidor.

5.4 Estacionar trotinete

Cliente - o cliente solicita o código de reserva da trotinete que quer estacionar e as coordenadas **x** e **y** da localização onde pretende estacionar.

Servidor - procura com o código de reserva fornecido o tempo de ínicio da reserva, e foi calculado o tempo final da reserva, a duração e a distância percorrida. Depois disso, calculou-se o custo da viagem, sendo 20 cêntimos por minuto a somar com 30 cêntimos por kilometro. A seguir, efetuou-se a alteração do estado para livre, a remoção da trotinete da lista das reservas e a alteração da a localização da trotinete para a localização do estacionamento. Por fim, é enviado para o cliente o as mensagens com o preço ou de erro.

Cliente - o cliente recebe as mensagens enviada pelo servidor.

6 Conclusão

Em conclusão, a realização deste projeto obrigou-nos a aprender e a pesquisar muito além das aulas, tanto teóricas como práticas. Conseguimos perceber com sucesso a conexão entre o servidor-cliente e como se faziam as ligações entre eles. A maior dificuldade foi a utilização de threads, a criação de processos concorrentes baralhou-nos um pouco e fez com que tivessemos muito mais trabalho na tentativa da correção de erros.