Universidade do Minho
Escola de Engenharia
Licenciatura em Engenharia Informática

Unidade Curricular de Sistemas Distríbuidos

Ano Letivo de 2022/2023

Trabalho Prático

Filipa Gomes(A96556) Miguel Gomes(A93294) Tiago Carneiro(A93207)
Pedro Ferreira(A93282)

9 de janeiro de 2023

SD

Índice

1 Introdução

O objetivo pretendido é a implementação de uma plataforma de gestão de uma frota de trotinetas elétricas. A plataforma deve estar na forma de um sistema cliente-servidor em Java usando *sockets* e *threads*. O principal objetivo do serviço é permitir que os utilizadores reservem e estacionem trotinetas em diferentes locais. O mapa é uma matriz de posições N x N, onde N é um número inteiro positivo. As coordenadas geográficas dos locais são índices pares discretos. Para calcular a distância entre dois locais, é utilizada a norma *L1*, também conhecida como distância de *Manhattan*. Para garantir uma boa distribuição das trotinetas pelo mapa, existe um sistema de recompensas. Este sistema de recompensas, recompensa os utilizadores por levarem as trotinetas da posição A a B. A qualquer momento, há uma lista de recompensas disponível, atualizada ao longo do tempo. Cada recompensa é identificada por um par origem-destino, com um valor de recompensa associada à distribuição de trotinetas pelo mapa.

1.1 Estrutura do Relatório

O presente relatório é constituído por 6 capítulos:

- **Introdução:** Pequena introdução do trabalho realizado e alguma contextualização;
- **Requisitos:** Definição e descrição dos requisitos;
- **Funcionalidades:** Descrição e definição das funcionalidades do sistema;
- **Servidor:** Descrição do funcionamento do Servidor no Sistema;
- **Cliente:** Descrição do funcionamento do Cliente no Sistema;
- **Conclusões:** Conclusões finais da realização do projeto proposto

2 Requisitos

Tal como pedido no enunciado, na implementação deverão ser respeitados os seguintes requisitos:

- Cada cliente apenas pode estabelecer uma única conexão para o servidor, por onde pedidos, respostas e notificações devem passar;
- O protocolo de comunicação entre cliente e servidor deverá ser num formato binário, por código desenvolvido no trabalho, podendo recorrer apenas a Data[Input|Output]Stream.
- Para o serviço não ficar vulnerável a clientes lentos, não deverá ter *threads* do servidor a escrever em mais do que um *socket*, devendo as escritas ser feitas por *threads* associadas a esse socket.
- Para manter os clientes responsivos, o sistema que gera recompensas deve correr no servidor em *thread(s)* específica(s), e nunca em *threads* associadas a clientes

3 Funcionalidades

Este serviço deverá suportar a seguinte funcionalidade:

- Autenticação e registo de utilizador, dado o seu nome e palavra-passe. Sempre que um utilizador desejar interagir com o serviço deverá estabelecer uma conexão e ser autenticado pelo servidor.
- Listagem dos locais onde existem trotinetas livres, até uma distância fixa D (e.g., D = 2) de um determinado local.
- Listagem das recompensas com origem até uma distância fixa D (e.g., D = 2) de um determinado local, sendo este o destino.
- Reserva de uma trotineta livre, o mais perto possível de determinado local, limitado uma distância fixa D. O servidor deverá responder com o local e um código de reserva, ou código de insucesso, caso tal não seja possível.
- Estacionamento de uma trotineta dando o código de reserva e o local. O servidor deve informar o cliente do custo da viagem, em função do tempo passado desde a reserva e da distância percorrida. Caso a viagem corresponda a uma recompensa, é informado do valor da recompensa. A aplicabilidade da recompensa deve ser avaliada no estacionamento, de lista segundo a recompensas em vigor nesse momento.
- Um cliente pedir para ser notificado quando apareçam recompensas com origem a menos de uma distância fixa D de determinado local. As notificações poderão ser enviadas muito mais tarde, devendo, entretanto, o cliente poder prosseguir com outras operações. O cliente poderá cancelar os pedidos de notificação.

Funcionalidades implementadas pelo grupo como extra:

- Pick-up points: serve para uma fazer uma distribuição equilibrada de trotinetas pelo mapa.

4 Servidor

4.1 Comunicação

4.1.1 Mensagem

Para facilitar a troca de informação entre Servidor-Cliente, foi implementado um protocolo de Mensagem em que consiste em dois campos:

- **Tag :** Identificador do Cliente no servidor;
- **Data :** Um par (Comando, Objeto) onde:
  - **Comando :** Identificador de comando a ser executado;
  - **Objeto :** Objeto necessário para a execução do comando;

4.1.2 Comandos

Os vários identificadores suportados pelo sistema:

- **OK :** Comando do Servidor para o Cliente como resposta a um pedido bem-sucedido;
- **ERROR :** Comando do Servidor para o Cliente como resposta a um pedido sem sucesso;
- **REGISTER :** Comando de pedido de Cliente para Servidor para registar um Utilizador;
- **LOGIN :** Comando de pedido de Cliente para Servidor para autenticar um Utilizador;
- **LIST_SCOOTERS :** Comando de pedido do Cliente, quando este pretende visualizar as trotinetas num raio D da sua posição;
- **RESERVE_SCOOTER :** Comando de pedido do Cliente, quando este pretende reservar uma trotineta num raio D da sua posição;
- **PARK_SCOOTER :** Comando de pedido do Cliente, quando este pretende estacionar a sua trotineta num determinado local;

- **KILL :** Comando de Cliente para Servidor, para encerrar o servidor, sendo apenas autorizado ao User "admin";
- **GET_PROFILE :** Comando de Cliente para Servidor, onde o Cliente pede informação sobre um certo objeto User;
- **SET_PROFILE :** Comando de Cliente para Servidor, onde o Cliente atualiza a informação sobre um certo objeto User, no Servidor;
- **GET_PICKUP_REWARDS :** Comando de Cliente para Servidor, caso o Utilizador esteja subscrito ao sistema de pontos, é lhe informado os valores das recompensas de levantamento de trotinetas num raio D da sua posição;
- **GET_DROP_REWARDS :** Comando de Cliente para Servidor, caso o Utilizador esteja subscrito ao sistema de pontos, é lhe informado os valores das recompensas de estacionamento de trotinetas num raio D do seu destino;
- **UPDATE_POSTITION :** Comando de Cliente para Servidor, onde o Cliente atualiza a posição de um certo objeto User, no Servidor;

4.2 Autenticação

Primeiramente foi implementado um sistema de autenticação de utilizador. Este processo começa no módulo Cliente, por um pedido das credenciais do utilizador. Estas são depois transformadas num objeto *default* User (que contem os campos de username e password apenas). O objeto User é enviado para o Servidor para a sua validação, ficando o cliente a aguardar uma resposta. No lado do Servidor, este recebe o pedido no formato explicado no subcapítulo anterior, onde verifica se o objeto é do tipo User. Caso seja, as credenciais são extraídas e validadas com os dados em memória, em situação de sucesso é enviado uma mensagem, com um objeto *boolean* com valor verdadeiro, ao Cliente. Caso não haja validação, é enviada para o Cliente uma mensagem com um objeto *Exception*. No lado do Cliente, este recebe a resposta enviada pelo Servidor, verifica se o objeto recebido é do tipo *boolean* e caso seja verdadeiro é depois pedido o resto dos dados do perfil do Utilizador. Caso a resposta não seja um objeto do tipo *boolean* verifica se é *Exception*, avisando o Utilizador conforme a *Exception* recebida.

5 Cliente

O Cliente é o módulo ativador das funcionalidades do Servidor, tendo também a capacidade de imprimir menus e respostas do Servidor no ecrã do Utilizador. O Utilizador utiliza uma TUI *Terminal User Interface* implementada no módulo Cliente. Nesta TUI foram implementados os seguintes menus:

- **Menu inicial:** Onde o Utilizador escolher se se quer se autenticar, registar ou sair do sistema;
- **Menu principal:** Onde o Utilizador escolhe se pretende ir para a secção de perfil, trotinetas ou caso seja Administrador se deseja encerrar o Servidor;
- **Secção de perfil:** Onde o Utilizador pode visualizar o seu perfil, adicionar fundos ou alterar a sua posição no mapa;
- **Secção trotinetas:** Onde o Utilizador pode visualizar as trotinetas num raio D relativo à sua posição no mapa, reservar uma trotineta num raio D, estacionar uma trotineta num destino ou subscrever (ou não) ao sistema de pontos.

6 Conclusões

Para concluir este trabalho no âmbito da Unidade Curricular de Sistemas Distribuídos, foi efetuada uma avaliação aos requisitos pedidos para a realização do projeto e concluiu-se que a implementação realizada pelo grupo cumpre os mesmos. Para este efeito foram utilizados os conceitos de Estado Partilhado com a utilização de *Lock*s e *ReentrantLock*s assegurando uma implementação robusta com a minimização de erros de colisão de acessos, *Thread*s para suportar várias conexões em simultâneo no mesmo Servidor e *Socket*s para a realizar a comunicação entre Cliente-Servidor. A Interface Utilizador foi efetuada de modo que a sua utilização seja intuitiva e de fácil compreensão.