

Trabalho Prático

Sistema de Partilha de Bicicletas

Ricardo Ferreira^[A82568] Hugo Faria^[A81283]
Rodolfo Silva^[A81716] Bruno Veloso^[A78352]

Universidade do Minho, Departamento de Informática



Resumo

O presente relatório tem como objetivo abordar a modelação de um sistema multi agente para simular um sistema de partilha de bicicletas. Inicialmente será feita a contextualização do problema e de seguida serão abordados os diagramas feitos para o modelação do sistema, evidenciando para cada um quais as razões das decisões tomadas.

Keywords: agente, sistema multi agente, classes, diagrama

Conteúdo

Trabalho Prático Sistema de Partilha de Bicicletas	1
<i>Ricardo Ferreira Hugo Faria Rodolfo Silva Bruno Veloso</i>	
1 Introdução	3
1.1 Estrutura do Relatório	3
1.2 Contextualização	3
2 Agentes	4
2.1 Agente Utilizador	4
2.2 Agente Estação	4
2.3 Agente Interface	5
3 Classes Adicionais e Behaviours	6
3.1 Coordenadas	6
3.2 Mapa	6
3.3 System	6
3.4 Ticker Behaviour	6
3.5 One-Shot Behaviour	7
3.6 Cyclic Behaviour	7
4 Diagramas	7
4.1 Diagrama de Classes	7
4.2 Diagrama de Comunicação	8
4.3 Diagrama de Atividade	10
4.4 Diagrama de Estados	11
4.5 Diagramas de Sequência	11
5 Conclusão	14

1 Introdução

Este relatório é referente à primeira fase do trabalho prático da unidade curricular de Agentes Inteligentes do perfil de Sistemas Inteligentes. Este projeto consiste em modelar e implementar um sistema multi agente (SMA) que irá lidar com a distribuição de bicicletas em certas zonas. Este sistema multi agente irá permitir às pessoas requisitarem bicicletas para percorrer distâncias curtas e tentará lidar com a distribuição das mesmas de forma a não existir um aglomeramento elevado de bicicletas no mesmo espaço.

Nesta primeira fase, o objetivo é efetuar uma modelação do sistema multi agente. Para isso será apresentada a sua arquitetura e os protocolos de interação entre agentes utilizando Agent UML.

1.1 Estrutura do Relatório

O relatório encontra-se estruturado da seguinte forma:

Inicialmente, na *secção 1* será feita uma breve introdução e contextualização do problema do caso em estudo, a *secção 2* apresenta os vários agentes a implementar e quais as suas características, de seguida, na *secção 3* serão apresentadas as classes adicionais e os diferentes *behaviours* presentes no diagrama de classes, a *secção 4* apresenta os diferentes diagramas utilizados para modelar o sistema, explicitando para cada um as decisões tomadas. Por fim o relatório termina com conclusões na *secção 5*, onde é também apresentada uma análise crítica aos resultados obtidos.

1.2 Contextualização

Nos dias que correm um dos meios de transporte mais utilizados pelas pessoas para deslocações curtas tem sido a bicicleta. Sendo maioritariamente utilizados nas cidades, uma vez que são amigas do ambiente e evitam o tráfego elevado quando comparado com outros meios de transporte. De forma a lidar com aumento acentuado de procura de bicicletas existe a necessidade de criação de um sistema possível de aluguer de bicicletas e respetiva orientação do utilizador no após aluguer e devolução das mesmas.

Para isto foi-nos proposto a implementação de um sistema multi agente que consiga resolver tal problema e assim facilitar as nossas vidas quotidianas, em que numa fase inicial correspondente a este relatório se passará à modelação do mesmo sistema.

Este sistema possui alguns requisitos básicos como:

- Utilizador alugar bicicleta;
- Estações de bicicletas possam indicar melhores opções de devolução utilizando incentivos, de forma a tentar obter uma melhor distribuição das bicicletas, evitando assim um aglomerado;
- Permitir ao utilizador aceitar ou recusar incentivos.

2 Agentes

Após uma contextualização do problema proposto, serão abordados quais os agentes que o sistema multi agente deve implementar, descrevendo de forma detalhada cada um deles e explicando o porquê de tais decisões. São estes:

- **Agente Utilizador;**
- **Agente Estação;**
- **Agente Interface.**

2.1 Agente Utilizador

Este agente representará um utilizador do sistema ou seja, uma pessoa que pretende alugar uma bicicleta.

Com o objetivo de facilitar a resolução do problema proposto, este utilizador sempre que é criado no sistema começará com bicicleta e numa posição de uma das estações. Este agente terá também de saber qual a posição atual e o seu percurso a percorrer e será o mesmo a ter que calcular a distância percorrida até ao momento do percurso. Para isto, este terá então que guardar informações como:

- **Posição inicial**, para saber de onde partiu;
- **Posição destino**, que representa a posição da estação destino do utilizador;
- **Posição atual**, para que seja possível calcular a distância percorrida do percurso;
- **Estação destino**, para indicar o identificador da estação onde este tem de entregar a bicicleta;
- **Distância percorrida**, pois é preciso ser o utilizador a calcular a distância que este já percorreu do seu percurso por forma a mais tarde avisar o agente interface sobre a sua chegada;
- **Estação próxima**, que representará a estação mais próxima do mesmo utilizador, podendo ser a estação de destino ou não.
- **Aceitou Incentivo**, variável indicativa se o utilizador já aceitou um incentivo no passado.

2.2 Agente Estação

Este agente representará uma estação, representada por um ponto em que é possível requisitar e devolver bicicletas alugadas e por uma área de controlo que representa a área próxima dessa estação.

Agente estação terá que ter características como:

- Este terá que saber todos os utilizadores que se encontram na área de proximidade do mesmo;
- Estará em constante comunicação com os utilizadores para saber a posição dos mesmos, atualizando caso necessário, a sua lista de utilizadores na sua área de controlo;

- Terá que saber todas as informações relativamente a bicicletas (disponíveis e capacidade total de armazenamento);
- Tentar prever a quantidade de utilizadores que num futuro próximo possam devolver as bicicletas na sua estação;
- Tentar distribuir da melhor forma as bicicletas pelas diferentes estações oferecendo incentivos conforme a lotação de cada estação levando o utilizador a entregar a bicicleta numa estação que esteja com menos bicicletas.

Para responder a estas características, terá que guardar informações como:

- **Posição da estação**, para saber as suas coordenadas;
- **Capacidade**, que representa a capacidade total de bicicletas da estação;
- **Bicicletas**, para indicar quantas bicicletas a estação tem disponíveis de momento;
- **Coordenadas de área de controlo**, em que todas as coordenadas da sua área de proximidade são conhecidas e permitindo assim calcular quais os utilizadores dentro da sua área próxima;
- **Utilizadores na área de controlo**, para saber a quantidade de utilizadores que se encontram na sua área de controlo, identificando os mesmos, e assim prever a quantidade de utilizadores que possam devolver futuramente a bicicleta na sua estação;
- **Ocupação das estações**, para o mesmo saber o quão lotadas as outras estações estão e conseguir assim calcular melhor um incentivo a dar ao utilizador, de forma a atrair o mesmo para uma estação com menor taxa de ocupação.

2.3 Agente Interface

O agente interface será quem comunica com os agentes estação, de forma a que este(Agente Interface) consiga saber a disponibilidade de cada estação em termos de bicicletas (capacidade total e bicicletas disponíveis) e consequentemente saber também as posições dos utilizadores. Com estas informações obtidas depois de uma comunicação entre agentes, o agente interface mostrará toda esta informação graficamente para que o utilizador consiga visualizar a informação no sistema multi agente.

3 Classes Adicionais e Behaviours

Passaremos a abordar as classes adicionais que achamos pertinentes para a modelação deste sistema multi agente, anotando algumas características/atributos das mesmas. Estas classes passarão desde:

- classes de auxílio à implementação do sistema;
- classes que representarão os *behaviours*, que definirão qual o comportamento dos agentes;

3.1 Coordenadas

Esta classe serve para representar qualquer coordenada num mapa 2D. Para tal tem atributos como:

- **Coordenada X**, representa a posição no eixo do x num mapa 2D;
- **Coordenada Y**, que representa a posição no eixo do y num mapa 2D.

3.2 Mapa

O mapa vai representar o estado a ser desenhado pelo agente interface. Com esse efeito, possui o seguinte atributo:

- **Posição Estações**, um map que possui as Coordenadas de todas as estações e a sua respetiva identificação.

3.3 System

Devido ao facto de este sistema servir como "uma simulação", é necessário que de alguma forma seja possível simular o mesmo. Nesse sentido, o grupo decidiu modelar uma classe chamada System, que tem como objetivo criar utilizadores de x em x tempo e assim conseguir testar o mesmo para verificar a sua viabilidade.

3.4 Ticker Behaviour

Utilizado nos seguintes comportamentos dos agentes:

- **Movimento do utilizador**, em que de x em x tempo este *behaviour* irá movimentar o utilizador;
- **Geração de utilizadores**, no qual a cada y segundos, este irá gerar utilizadores para o sistema;
- **Desenho da interface**, em que de x em x segundos desenha o mapa e as informações das estações.

3.5 One-Shot Behaviour

Vai ser aplicado para fazer os comportamentos que requerem mandar uma mensagem a outro agente. A resposta vai ser recebida por parte dos *cyclic behaviours* que os agentes vão apresentar.

Serão utilizados nos seguintes momentos:

- Utilizador pedir a identificação da estação que controla a área do seu destino;
- Utilizador informar a estação da sua área sobre a sua nova posição;
- Estação avisar o utilizador que necessita de entregar a bicicleta quando este chegar ao seu destino;
- Estação fornecer o incentivo um utilizador para este se deslocar para outra estação que contenha poucas bicicletas disponíveis.

3.6 Cyclic Behaviour

O *cyclic behaviour* vai ser maioritariamente utilizado nos agentes como forma de continuamente esperar por *informs* ou *requests* feitos por outros agentes. Estes serão utilizados:

- Na espera por resposta por parte de uma estação ou utilizador.

4 Diagramas

De seguida serão apresentados e explicados alguns diagramas que serão a base da implementação do projeto numa segunda fase. Estes diagramas servem como base para apresentar uma arquitetura do sistema no geral, bem como os protocolos de comunicação usados entre os agentes e para designar o fluxo do sistema.

Para isto, foram desenvolvidos os seguintes diagramas: diagrama de classe, comunicação, atividade, estados e sequência.

4.1 Diagrama de Classes

Nesta secção é apresentada a arquitetura do sistema através do diagrama de classes. Neste diagrama é possível verificar as diferentes classes a implementar e visualizar também as diferentes relações que terão umas com as outras de forma a que o sistema funcione como um todo.

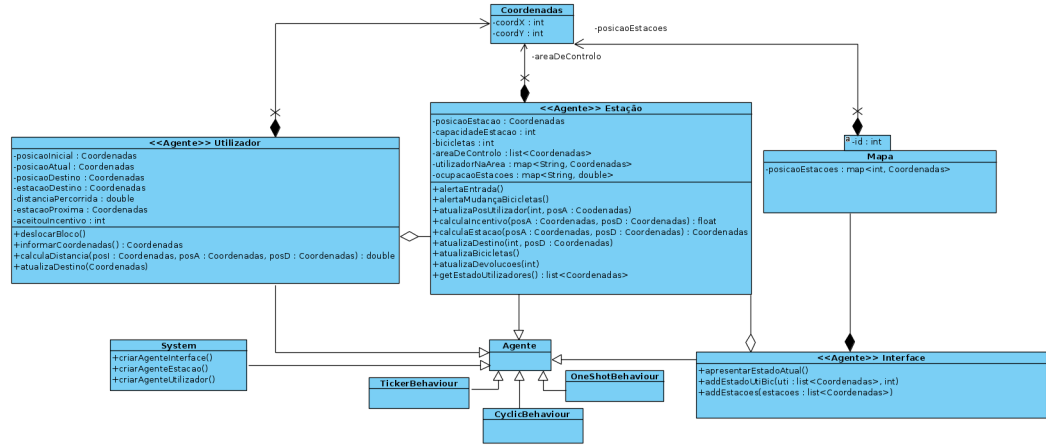


Figura 1. Arquitetura do sistema

Uma vez que as classes foram previamente explicadas na *secção 3*, passaremos agora a abordar as relações mais importantes existentes entre estas.

É de notar que o agente interface tem uma relação com a classe Mapa, isto deve-se ao facto de que, caso seja pertinente, o agente interface possa aceder a Mapa de forma a ser mais fácil obter informação sobre as posições de estações em vez de estar em constante comunicação para saber a sua posição visto que esta é fixa (Posição das Estações). Como era de esperar, existe uma relação entre o utilizador e a estação, sendo esta a principal comunicação que existirá neste sistema, uma vez que o utilizador precisa de fornecer a sua posição à estação, saber a lotação das mesmas, e a estação de fornecer incentivos ao utilizador. Por fim, é possível verificar ainda que também o agente interface tem relação com o agente estação, no qual servirá para a interface conhecer as lotações de cada estação e as posições de cada utilizador na sua área.

Sendo importante referir que algumas das relações identificadas anteriormente serão, na prática abstraídas por mecanismos de comunicação disponibilizados pelo JADE.

4.2 Diagrama de Comunicação

De forma a dinamizar o sistema, é necessário que os seus agentes comuniquem entre si, para isso é necessário que existam protocolos de comunicação estabelecidos, que serão as performatives(2), utilizados no auxílio da comunicação dos agentes. Na seguinte imagem, serão apresentadas os diferentes performatives usados entre cada par de agentes.

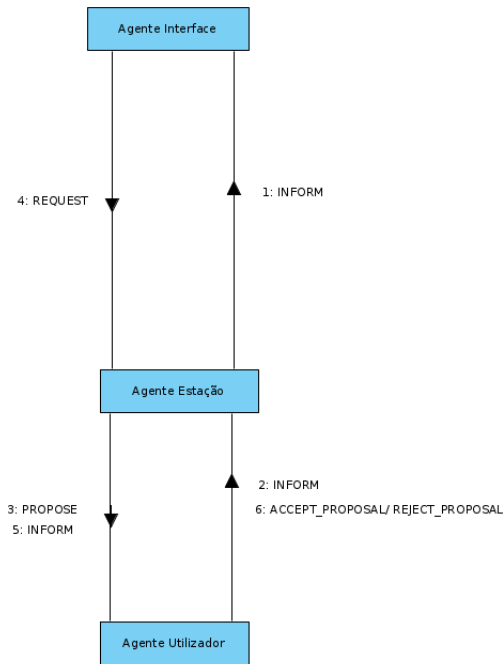


Figura 2. Comunicação entre agentes

Passaremos a explicar o porquê do uso de determinadas *performatives*.

No que diz respeito à comunicação entre Agente Estação e Agente Utilizador, existe um *inform* feito pelo utilizador no qual indica que se moveu e a respetiva distância. Já do lado do Agente Estação, este utiliza as *performatives propose* e *inform*. A *propose* é utilizada quando quer fornecer ao utilizador uma opção de escolha, que no caso deste sistema será o incentivo, e o *accept_proposal/reject_proposal* será quando o utilizador decidir escolher ou não o incentivo fazendo com que o Agente Estação informe o utilizador se mudou ou não a sua estação original ou a manteve usando para tal o *inform*.

Quanto à comunicação entre Agente Interface e Agente Estação, o Agente Interface utiliza uma *performatives request* que servirá para quando o Agente Interface quiser saber do estado atual do mapa (utilizadores e informações relativamente a cada estação) e do lado do Agente Estação é utilizado o *inform* para fornecer os dados que o Agente Interface pretende saber.

Definidos assim os protocolos de comunicação, fica então assim definida toda a comunicação envolvente no sistema multi agente.

4.3 Diagrama de Atividade

Quanto ao fluxo que indica como todo o sistema deve fluir, foi utilizado para expressar este fluxo um diagrama de atividade.

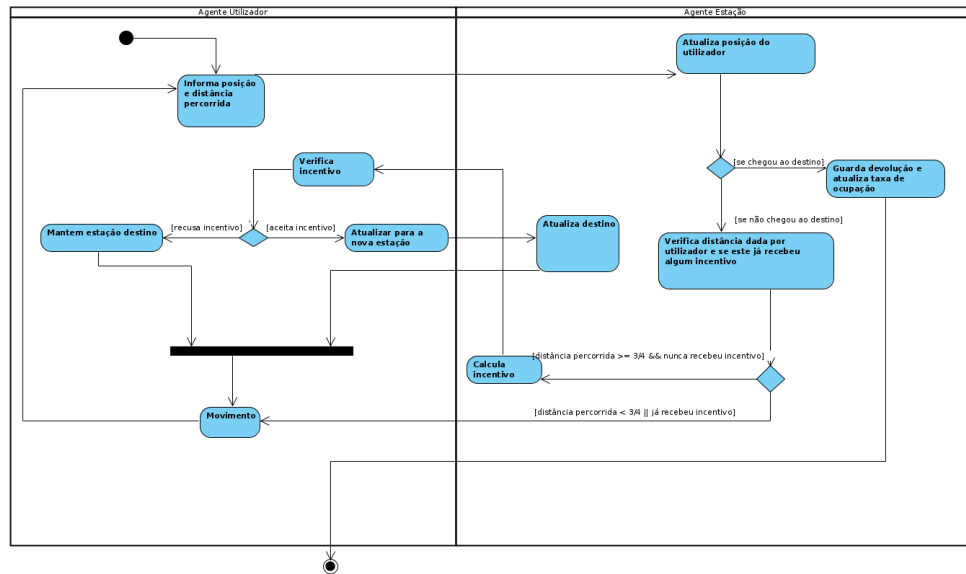


Figura 3. Fluxo do sistema

Neste diagrama é possível verificar de forma sucinta que é o utilizador o responsável pelo cálculo da distância total percorrida e da comunicação deste cálculo e respectiva posição atual à Estação mais próxima (da sua área). Esta verifica se este chegou ao destino ou não. Se chegou ao destino atualiza a sua lotação e guarda a devolução da bicicleta. Se não chegou ao destino, a estação verifica se a distância total percorrida já atingiu os 3/4 do caminho total proposto, se exceder esse valor esta calcula um incentivo que indica o utilizador para outra estação ou não. O incentivo é sempre calculado quando o utilizador já percorreu no mínimo 3/4 do caminho e quando este nunca recebeu incentivos, e este incentivo, caso não seja benéfico dar um incentivo ao utilizador, será 0. Caso se calcule um incentivo, o utilizador pode recusar o mesmo ou não. Se recusar, o destino dele mantém-se. Se aceitar, é atualizado o seu destino. Caso não se calcule incentivo, este utilizador pode continuar o seu caminho até ao destino.

Posto isto, este será todo o fluxo a ter em conta para a futura implementação deste sistema.

4.4 Diagrama de Estados

Quanto ao estado interno dos agentes utilizador, foi modelado um diagrama de estado para o mesmo. Neste diagrama pode-se verificar todos os estados que estes vão tomar, desde que saíram da estação inicial, desde ao estado após aceitação de incentivo ou não, até à devolução da bicicleta.

Estes podem ser observados na figura seguinte:

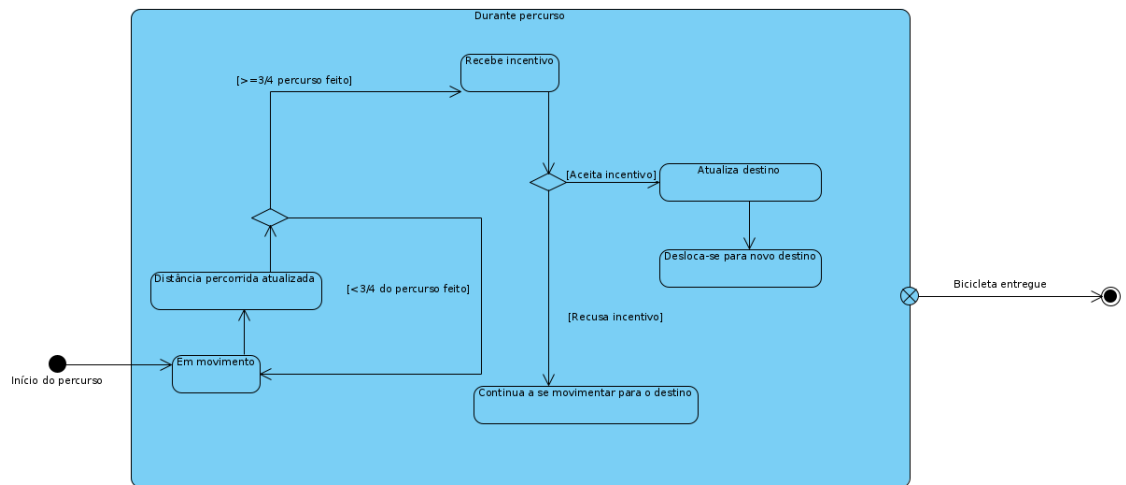


Figura 4. Estado interno dos agentes utilizador.

4.5 Diagramas de Sequência

No diagrama de sequência que se segue, é possível verificar todo o processo de obter informação até ao desenho da interface com tal informação. Para isso o agente interface pergunta à classe Mapa as posições das estações, ao qual esta lhe fornece tal informação. De seguida o agente interface pede a cada estação, os utilizadores que se encontram na área bem como o número de bicicletas disponíveis nas estações.

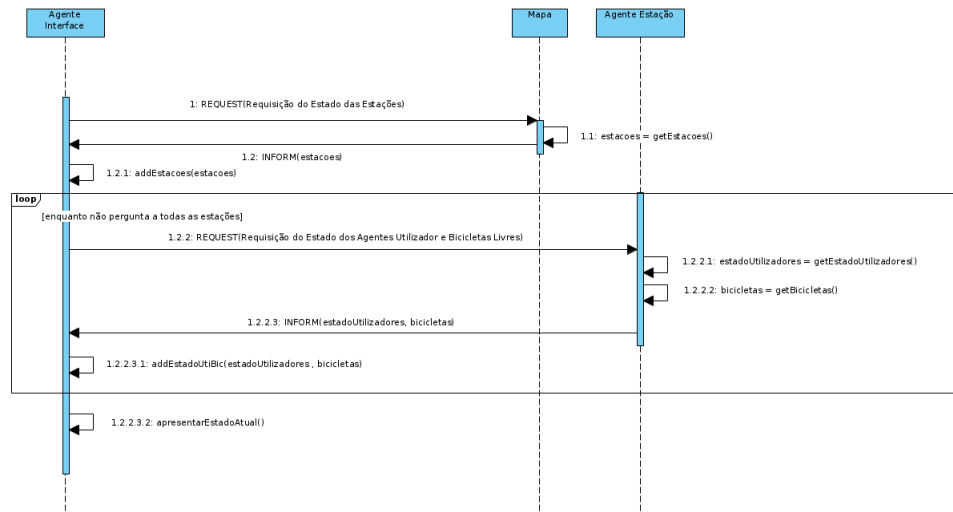


Figura 5. Diagrama de sequência que representa desenho da interface

No diagrama de sequência seguinte podemos observar toda a comunicação que existirá entre agente utilizador e agente estação. Esta comunicação já foi explicada anteriormente, no entanto é de notar que agora foram usadas *performatives* para explicar a mesma.

Sendo possível verificar que quando é fornecido um incentivo ao utilizador por parte da estação (usando a *performative PROPOSE*), este utilizador utilizará a *performative ACCEPT_PROPOSAL* caso queira aceitar o mesmo, ou *REJECT_PROPOSAL* caso o queira recusar. Existe também o uso da *performative INFORM* por parte da estação para dizer que o destino foi alterado e *INFORM* por parte do utilizador para dizer que chegou ao destino. De resto todo o diagrama flui da mesma forma que foi explicado no diagrama de actividades, por exemplo.

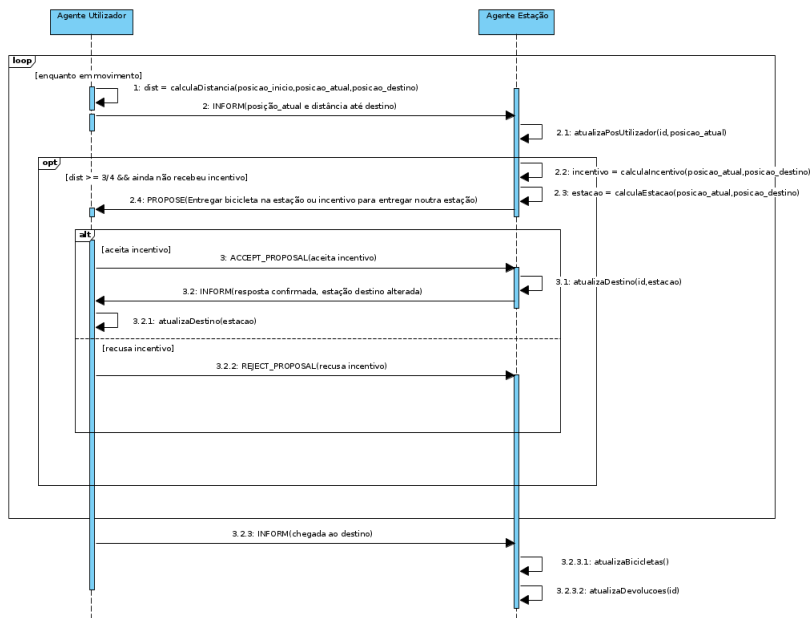


Figura 6. Diagrama de sequência que representa comunicação entre utilizador e estação

5 Conclusão

Nesta primeira fase do trabalho foi apresentada uma contextualização do nosso sistema multi agente que deve ser implementado, bem como a sua modelação.

Com o objetivo de esta modelação ser a arquitetura final do nosso sistema, foram identificados e explicados certos aspetos que irão ajudar na implementação do mesmo. Estes aspetos vão desde: explicação das classes a implementar, tendo sempre em conta os *behaviours* que indicarão o comportamento dos agentes, explicação dos diagramas que mostram tudo que deve ser implementado numa versão final e a comunicação entre agentes.

Este sistema, numa segunda fase do projeto, será implementado usando JADE em que na sua comunicação são usadas mensagens com a especificação FIPA(1). Também, num produto final, é de esperar que o mesmo sistema apresente uma interface com o qual o utilizador possa interagir e definir as suas intenções.

Bibliografia

- [1] FIPA website, <http://www.fipa.org/>, last visited 28 November 2020.
- [2] FIPA-Performatives, <http://jmvidal.cse.sc.edu/talks/agentcommunication/performatives.html?style=White>, last visited 28 November 2020.
- [3] JADE website, <https://jade.tilab.com/>, last visited 27 November 2020.
- [4] Bauer, B., Müller, J. P., & Odell, J. (2001). Agent UML: A formalism for specifying multiagent software systems. *International Journal of Software Engineering and Knowledge Engineering*, 11(3), 207–230