



Universidade do Minho  
Escola de Engenharia

# **Sistemas de Partilha de Bicicletas**

## **Agentes Inteligentes: Trabalho Prático**

Mestrado Integrado em Engenharia Informática

1º Semestre

2017-2018

Grupo 1

A74219 - Hugo Alves Carvalho  
A70676 - Marcos Morais Luís  
A74260 - Luís Miguel da Cunha Lima

26 de Novembro de 2017  
Braga

## Resumo

Este documento relata todo o estudo prévio e desenho da arquitetura do sistema a ser implementado na área de AGENTES E SISTEMAS MULTIAGENTE, realizado no âmbito da componente prática da unidade curricular de **Agentes Inteligentes**, pretendo colocar em prática os elementos estudados durante o decorrer do semestre letivo. Este documento serve para expor o tema e o problema em concreto, propondo a sua resolução com a identificação e modulação dos vários aspetos necessários para a implementação do sistema.

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>5</b>
<b>2</b>	<b>Contextualização do Problema</b>	<b>6</b>
<b>3</b>	<b>Agentes</b>	<b>7</b>
3.1	Agente Utilizador . . . . .	7
3.1.1	Características . . . . .	7
3.2	Agente Gestor de Estações . . . . .	8
3.2.1	Características . . . . .	8
3.3	Agente de Estação . . . . .	8
3.3.1	Características . . . . .	8
3.4	Agente Interface . . . . .	9
3.4.1	Características . . . . .	9
3.5	Agente de Informação Geral . . . . .	9
3.5.1	Características . . . . .	9
<b>4</b>	<b>Arquitetura do Sistema</b>	<b>10</b>
<b>5</b>	<b>Comunicação entre Agentes</b>	<b>11</b>
5.1	Agente Estação - Agente Gestor de Estações . . . . .	11
5.2	Agente Estação - Agente Interface . . . . .	12
5.3	Agente Informação Geral - Agente Utilizador . . . . .	12
5.4	Agente de Estação - Agente Utilizador . . . . .	12
<b>6</b>	<b>Proposta de Solução Inicial</b>	<b>14</b>
6.1	Propostas dos Agentes Estação . . . . .	14
6.2	Análise do Agente Gestor de Estações . . . . .	14
6.3	Ações dos Agentes Utilizadores . . . . .	14
<b>7</b>	<b>Gestão de Riscos/Possíveis Soluções</b>	<b>16</b>
<b>8</b>	<b>Implementação</b>	<b>17</b>
8.1	Decisões Tomadas/Estratégias Implementadas . . . . .	17
8.2	Agente Utilizador . . . . .	17
8.3	Agente Gestor de Estações . . . . .	20
8.4	Agente Estação . . . . .	20
8.5	Agente de Informação Geral . . . . .	22
8.6	Agente Interface . . . . .	22
<b>9</b>	<b>Interface</b>	<b>23</b>
<b>10</b>	<b>Resultados</b>	<b>25</b>
<b>11</b>	<b>Vulnerabilidades/Limitações do Sistema</b>	<b>26</b>
	<b>Soluções Implementadas</b>	<b>26</b>
<b>12</b>	<b>Conclusões e Trabalho Futuro</b>	<b>28</b>
<b>13</b>	<b>Referências Bibliográficas</b>	<b>29</b>

## Lista de Figuras

1	"Bike Sharing Takes Off" by Statista . . . . .	5
2	"Overview da Arquitetura	10
3	<i>ACL performatives</i> . . . . .	11
4	Comunicação Agente Estação - Agente Gestor de Estações . . . . .	11
5	Comunicação Agente Estação - Agente Interface . . . . .	12
6	Comunicação Agente Informação Geral - Agente Utilizador . . . . .	12
7	Comunicação 1 - Agente Utilizador - Agente Estação . . . . .	12
8	Comunicação 2 - Agente Utilizador - Agente Estação . . . . .	13
9	Comunicação 3 - Agente Utilizador - Agente Estação . . . . .	13
10	Comunicação 4 - Agente Utilizador - Agente Estação . . . . .	13
11	Modelo inicial de ofertas sugeridas pelos AE . . . . .	14
12	Modelo de Análise de Informação do Agente Gestor de Estações . . . . .	14
13	Modelo inicial de níveis de aceitação e parâmetros correspondentes . . . . .	15
14	Correlação entre Nível Médio Calculado e Desconto Aceitável . . . . .	15
15	Ilustração do posicionamento geográfico das estações para efeitos de teste . . . . .	17
16	Percurso com início na Estação1 e destino Estação2 . . . . .	19
17	Ilustração da proposta realizada pela Estação3 e aceitação por parte do Utilizador . . . . .	19
18	Ilustração da proposta realizada pela Estação3 e negação por parte do Utilizador . . . . .	19
19	Mensagens enviadas pelo Agente Gestor . . . . .	20
20	Gráfico Quantitativo - Necessário para a perceção de stock em cada uma das estações. . . . .	23
21	Gráfico Percentual - Utilizado para saber a percentagem de ocupação relativamente à capacidade da estação, crucial para a gestão de recursos. . . . .	23
22	Gráfico de Métricas - Possibilita saber onde se encontram os recursos de todo o sistema e a capacidade do mesmo. . . . .	23
23	Gráfico de Utilizadores - Possibilita saber quantos utilizadores estão no sistema ao longo do tempo. . . . .	24
24	Gráfico de Descontos - Necessário para saber o número de descontos aceites, ou seja, número de vezes que utilizador muda o seu destino. . . . .	24
25	Ilustração das áreas de propostas sobrepostas . . . . .	26
26	Ilustração da proposta aceite por parte da Estação onde o utilizador iniciou o percurso . . . . .	27

# 1 Introdução

A locomoção é a capacidade que muitos organismos têm de se movimentarem por seus próprios meios no habitat em que vivem. Os seres humanos para além de o conseguirem fazer devido a sua fisionomia conseguem também a partir de máquinas construídas por eles. Dos vários meios de locomoção que existem criados pelo Homem, a bicicleta começa a ser dos mais aliciantes para meios urbanos devido principalmente ao seu impacto ambiental ser nulo, mas também por ser um meio que necessita de pouco espaço para se mover. Cidades como Amesterdão, Copenhaga, Berlim, França e Tóquio cada vez mais interditam os centros da cidade a automóveis e motorizados, fazendo com que a bicicleta se torne mais aliciante e utilizada. Devido ao aumento da demanda por este meio de transporte começaram então a surgir os "Sistemas de Partilha de Bicicletas".

Um Sistema de Partilha de Bicicletas consiste em disponibilizar bicicletas em vários pontos espalhados num certo mapa e deixar utilizadores usufruir das mesmas em períodos de tempo relativamente pequenos podendo estes deslocar -se do ponto A até ao ponto B. Este sistema surgiu na Europa por volta de 1965, ganhando robustez e adeptos nos anos 2000's devido ao avanço tecnológico e a recente preocupação ambiental, atingindo assim em 2015 o numero de 1.000.000 de bicicletas utilizadas neste tipo de sistema.

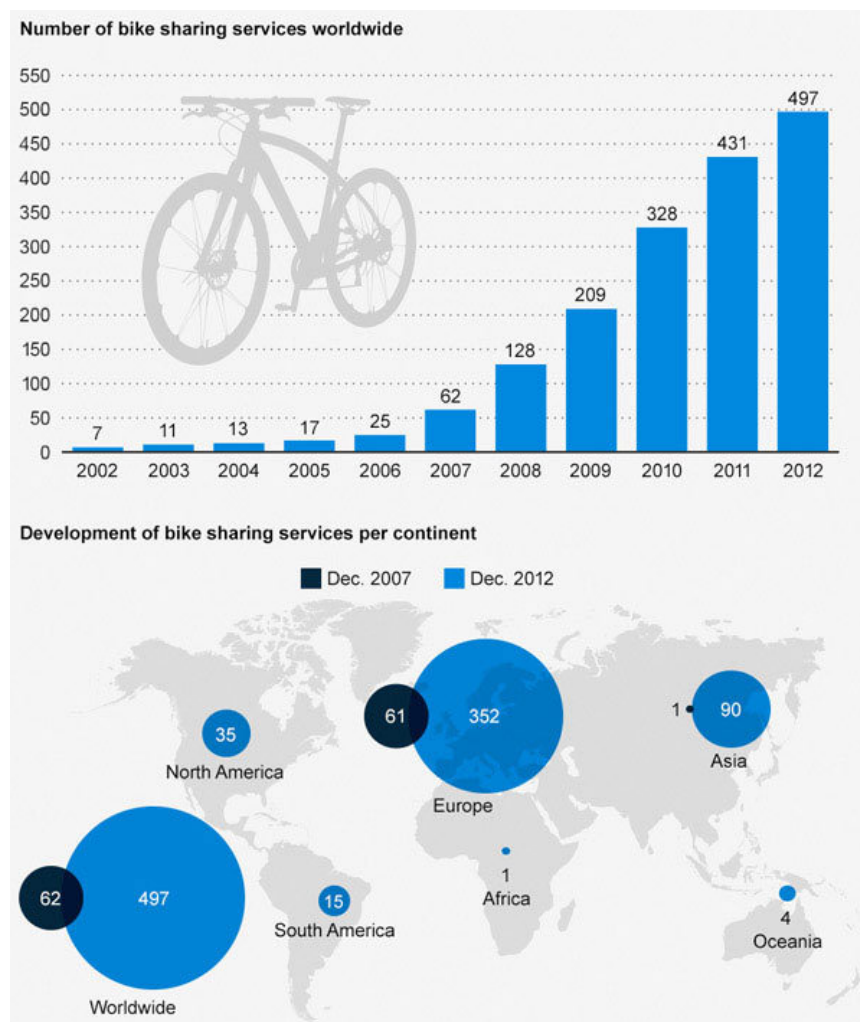


Figura 1: "Bike Sharing Takes Off" by Statista

## 2 Contextualização do Problema

Inicialmente foi feita uma procura para verificar os vários tipos de sistemas existentes e quais destes é que poderiam ser enquadrados na nossa solução. Chegou-se então à conclusão que existem no mínimo quatro tipos de sistemas de partilha de bicicletas: *Unregulated*, *Deposit*, *MemberShip* e *Dockless*. Destes quatro sistemas apenas o *Deposit* e o *MemberShip* se adequam ao ambiente no qual está inserido o problema pois ambos apresentam postos espalhados pela área onde são deixadas as bicicletas para uso do próximo utilizador.

Após o estudo dos dois tipos de sistemas conseguimos identificar as estratégias de cada um e diferenciá-los.

- ***Deposit*** - Sistema onde o conceito principal tem por base a utilização da bicicleta pagando o seu aluguer através de máquinas que se encontram nas estações ou através de métodos de pagamento por via eletrónica.
- ***MemberShip***- Sistema em que existe um plano de subscrições onde é fornecido um cartão ao utilizador, permitindo assim a este que usufrua das bicicletas.

Podemos então concluir que estes dois sistemas apresentam semelhanças, os dois têm pontos de paragem que iremos denominar como Estações onde é efetuado o pagamento/cartão de subscrição e utilizadores. A desvantagem encontrada em qualquer um dos dois sistemas é a gestão de stock de recursos neste caso das bicicletas.

Vamos assumir um exemplo real em que uma das empresas fornecedoras deste serviço tem uma Estação que iremos denominar "Estação A", localizada nos arredores da cidade em local de habitação de inquilinos locais, e uma segunda estação que terá o nome "Estação B" que se encontra localizada no centro histórico da cidade, local de grande afluência de turistas. Consideremos que a estação A é utilizada maioritariamente por inquilinos locais que pretendem chegar ao local B. Quando estes deixam lá as suas bicicletas, estas podem posteriormente serem transportadas para outro local que não a estação A, originando assim um problema de falta de transporte na localização da estação A. O inverso pode também acontecer, ou seja, pode uma estação apresentar um excesso de recursos para a sua capacidade de armazenamento e gestão de recursos.

Como tal, para a resolução deste problema descrito, é necessário a implementação de um sistema inteligente que através de incentivos aos utilizadores faça com que estes alterem as suas rotas, causando assim intencionalmente a gestão e equilíbrio de recursos pelas várias estações de determinadas áreas.

### 3 Agentes

Após a análise do problema e sua devida contextualização, podemos então apontar quais os tipos de agentes que terão que ser implementados para tornar o sistema abrangente, no sentido de cobrir o maior número de aspectos possíveis.

Temos então uma comunicação/interação feita entre as Estações e os utilizadores, o que leva à identificação dos dois primeiros tipos de agentes do nosso sistema, agente utilizador e agente estação. Por sua vez, esta comunicação terá que ser gerida e apresentada ao utilizador, através de um agente Interface.

Todas as estações podem sofrer alterações nas suas dimensões de stock, nos valores do desconto propostos ou no seu raio de proximidade, tornando-se assim fundamental a existência de um agente Gestor que consiga gerir todos os recursos das várias estações e as suas respetivas alterações.

O sistema englobará ainda um Agente de Informação Geral que providencia informações gerais, não disponíveis na interface, que podem ser consideradas na hora de decisão de aceitação de uma oferta por parte de um agente Utilizador.

Após esta pequena reflexão, podemos então numerar os diferentes agentes do sistema:

- Agente Utilizador
- Agente Gestor de Estações
- Agente de Estação
- Agente de Informação Geral
- Agente Interface

De seguida iremos descrever cada um dos agentes identificados, caracterizando-os com o objetivo de modelar corretamente o sistema. A criação de todos estes agentes será feita na *main classe* gerando assim os diferentes tipos de agentes.

Para além dos agentes anteriormente citados é necessário um suporte de interface para a visualização das informações referentes a este sistema, para isso foi criada uma camada interface em Swing.

#### 3.1 Agente Utilizador

O Agente Utilizador corresponde ao papel que o utilizador da bicicleta irá ter, podendo utilizar o seu smartphone ou smartwatch para comunicar com o sistema.

##### 3.1.1 Características

Relativamente às características do Agente Utilizador e que serão relevantes para o funcionamento do sistema, teremos:

- **Posição atual** - Posição que será atualizada constantemente de 1 em 1s, informando assim conhecimento da localização atual do utilizador. Será caracterizada por dois valores que representam os dois eixos de um modelo 2D.
  - Eixo x;
  - Eixo y;
- **Estação Destino** - Estação que o utilizador decide como seu destino final. Este parâmetro pode ser alterado ao longo da sua viagem, por exemplo quando aceita uma oferta de uma outra estação.
- **Posição atual** - Coordenadas da estação que o utilizador escolhe como seu destino final. Estes parâmetros são alterados ao longo da sua viagem sempre que o Agente Utilizador aceita uma oferta de uma outra estação.
  - Eixo x;

- Eixo y;
- **Condição Física** - Aspeto que irá influenciar a posterior decisão relativamente a uma oferta feita pelo sistema. Esta será caracterizada por um de três níveis disponíveis:
  - baixa, média e alta;
- **Idade** - A idade do indivíduo irá também ser um fator em ter em conta na hora da decisão perante uma oferta.

## 3.2 Agente Gestor de Estações

Este agente tem o propósito de realizar o controlo, de forma periódica, a cada um dos Agentes Estação, tornando assim mais perceptível qualquer tipo de falhas e interagindo com este informando as decisões a tomar.

Por outro lado, este agente apresenta características essenciais para garantir corretamente o equilíbrio e rendimento do sistema: pode informar uma estação se esta precisa de aumentar/diminuir o stock das suas bicicletas, aumentar o seu raio de proximidade ou alterar o seu modelo de incentivos.

### 3.2.1 Características

- **Bicicletas Livres** - Número de bicicletas livres que o Agente Gestor de Estações tem em sua posse. Esta característica torna-se fundamental para garantir um melhor equilíbrio na distribuição de bicicletas pelas diversas estações.

## 3.3 Agente de Estação

Agente que será responsável por caracterizar uma estação, mantendo os seus dados devidamente atualizados. Este apresenta a capacidade de interagir com a Interface e posteriormente com Utilizadores. Por outro lado, esta mantém uma comunicação constante com o agente gestor de estações de modo a garantir o maior rendimento da sua estação.

### 3.3.1 Características

- **Capacidade Total** - Valor que representará o número máximo de bicicletas que pode supervisionar e guardar.
- **Stock Atual** - Valor que representará o número atual de bicicletas que estão na estação.
- **Nível de preenchimento da estação** - Após analisar a capacidade total e o stock atual podemos caracterizar o nível de preenchimento da mesma, variando entre três níveis:
  - baixo, médio e alto;
- **Área de Proximidade** - A estação irá comunicar com os utilizadores que se encontram numa certa área ao redor dela. Para isso é preciso caracterizar a área, ou seja, o seu tamanho. Para tal, assumindo que se trata de um radar circular, a única medida necessária para tal é o raio de uma circunferência, sendo mais uma vez utilizado a escala metros.
- **Localização** - Característica que representa a localização da estação no mapa. Será caracterizada no mesmo formato da posição atual do agente utilizador apresentado anteriormente.
  - Eixo x;
  - Eixo y;
- **Descontos Aceites** - Número total de descontos que já foram aceites pelos Agentes Utilizador, isto é, incentivos que fizeram com que este alterasse a sua rota tornando a estação como seu destino final.



- **Descontos Recusados** - Número total de descontos recusados pelos diversos Agentes Utilizador.

### 3.4 Agente Interface

A existência deste agente tem como objetivo a representação visual dos vários aspetos de cada uma das estações, de forma a auxiliar o utilizador na sua tomada de decisão.

#### 3.4.1 Características

- **Estações** - Informação sobre todas as estações que se encontram ativas no sistema.
- **Bicicletas em Circulação** - Número de bicicletas que se encontram em uso no momento atual.
- **Bicicletas Paradas** - Número total de bicicletas prontas para a utilização.
- **Bicicletas Paradas por Estação** - Número total de bicicletas prontas para utilização em cada uma das estações.

### 3.5 Agente de Informação Geral

Este agente providencia informações gerais que podem ser consideradas na hora de decisão de aceitação de uma oferta por parte do utilizador.

#### 3.5.1 Características

- **Estação do Ano** - A estação do ano pode influenciar o número de utilizadores, sendo então uma informação útil para as estações.
  - outono, inverno, primavera, verão;
- **Hora** - A hora do dia é um fator que pode influenciar a decisão do utilizador perante uma oferta
  - hh/mm
- **Nível de luminosidade** - Como a hora em diferentes países pode ter um nível diferente de luminosidade decidiu-se também caracterizar este aspeto
  - nula, média e alta;
- **Nível de humidade** - A humidade apresenta uma forte influencia no esforço físico, sendo assim fundamental considera-la nas características deste agente
  - baixo, média e alto;

## 4 Arquitetura do Sistema

De seguida apresenta-se um modelo que demonstra o modo como os agentes vão comunicar entre si, tendo em conta a existência de vários agentes estação e agentes utilizador. É de realçar que existem comunicações bilaterais entre AE e AU. O Agente Informação Geral apenas disponibiliza informação, não recebendo qualquer tipo de mensagens. O Agente Gestor de Estações receberá informações dos vários postos e, após um processo de filtragem, fornece dados à Interface que serão mostrados posteriormente aos Agentes Utilizadores.

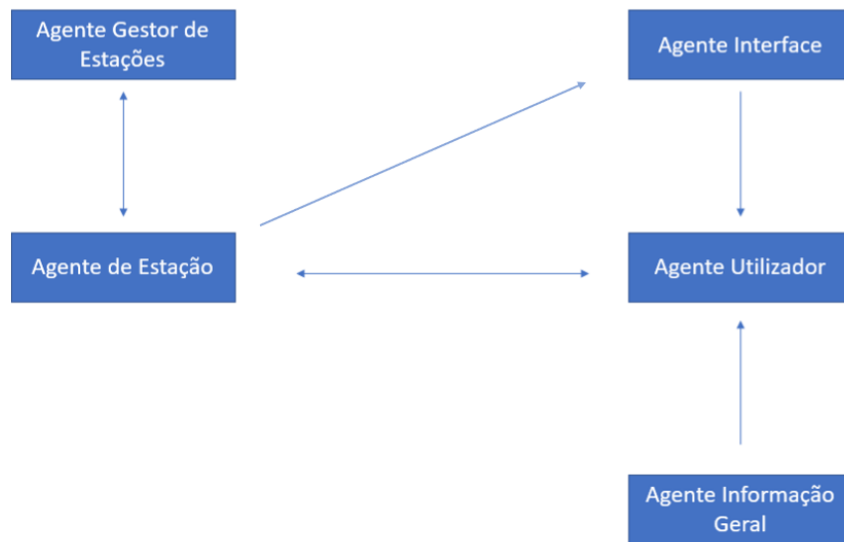


Figura 2: "Overview da Arquitetura"

Conseguimos assim ter uma visão geral do sistema e a sua arquitetura. Após esta conceção, o passo seguinte passa por identificar e interpretar os fatores que vão influenciar quer a proposta de aliciação por parte das Estações, como também os fatores quer internos e externos na hora da tomada de decisão por parte do Agente Utilizador.

## 5 Comunicação entre Agentes

Os vários agentes anteriormente classificados irão interagir entre eles ao longo tempo, torna-se então importante explorar as diferentes mensagens e respostas que cada um dos agentes poderá enviar. A Ontologia, modo de estruturação da comunicação, será dado por *Foundation for Intelligent Physical Agents* (FIPA). É também necessário definir a linguagem de comunicação entre os agentes pois não basta saber a ontologia para que haja comunicação. Deste modo, recorrendo a um conjunto de performatives Agents Communication Language (ACL), é possível clarificar os objetivos e as diferentes ações de comunicação entre agentes.

performative	passing info	requesting info	negotiation	performing actions	error handling
accept-proposal			x		
agree				x	
cancel		x		x	
cfp			x		
confirm	x				
disconfirm	x				
failure					x
inform	x				
inform-if	x				
inform-ref	x				
not-understood					x
propose			x		
query-if		x			
query-ref		x			
refuse				x	
reject-proposal			x		
request				x	
request-when				x	
request-whenever				x	
subscribe		x			

Figura 3: *ACL performatives*

Tendo em consideração os dados apresentados na figura 3, é agora necessário estabelecer uma conexão entre os diferentes tipos de *performatives* e as interações representadas no esquema da figura 2.

### 5.1 Agente Estação - Agente Gestor de Estações

Este tipo de comunicação representa apenas passagem de informação entre os agentes. Deste modo, o seu comportamento será implementado com as *performatives passing info*.

Em seguida apresentamos um esquema da comunicação entre estes dois agentes.

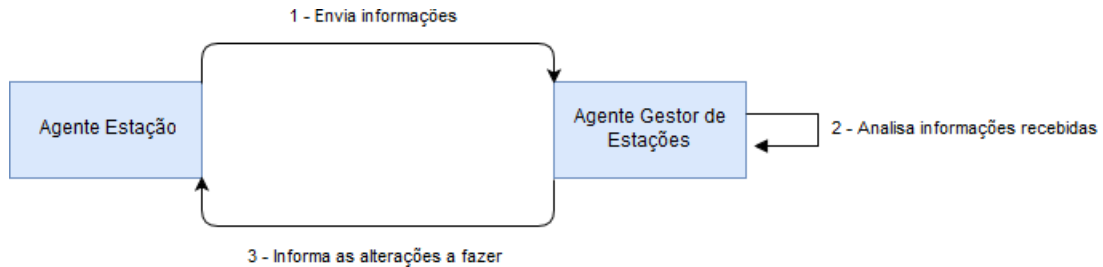


Figura 4: Comunicação Agente Estação - Agente Gestor de Estações

## 5.2 Agente Estação - Agente Interface

Seguindo o mesmo raciocínio do exemplo anterior, esta comunicação será implementada igualmente com as *performatives passing info*. Neste caso, existe apenas uma comunicação do Agente Estação para o Agente Interface, como podemos verificar no diagrama seguinte:

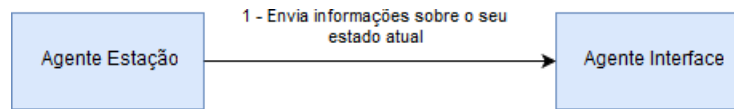


Figura 5: Comunicação Agente Estação - Agente Interface

## 5.3 Agente Informação Geral - Agente Utilizador

Este tipo de comunicação representa apenas passagem de informação entre os agentes. Deste modo, o seu comportamento será implementado com as *performatives passing info*.

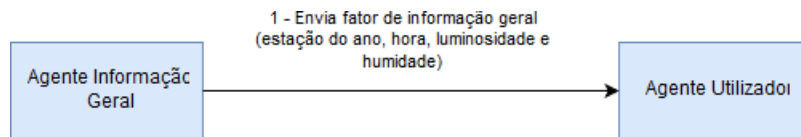


Figura 6: Comunicação Agente Informação Geral - Agente Utilizador

## 5.4 Agente de Estação - Agente Utilizador

Nesta comunicação existem várias condições que podem ser influenciar as mensagens que são enviadas entre os agentes.

Em primeiro lugar, são utilizadas as *performing actions* uma vez que existe passagem de informação entre o Agente Utilizador para o Agente Estação, informando-o das suas coordenadas no momento atual.

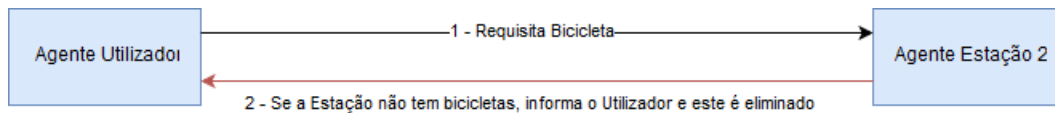


Figura 7: Comunicação 1 - Agente Utilizador - Agente Estação

Em seguida, já com uma bicicleta em sua posse, o agente Utilizador informa cada um dos agentes Estação das suas coordenadas atuais.

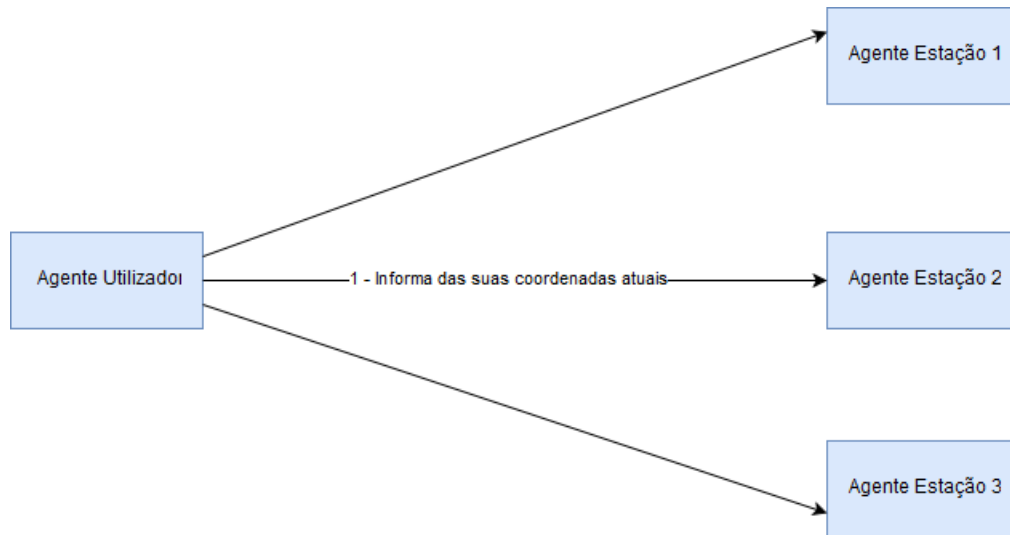


Figura 8: Comunicação 2 - Agente Utilizador - Agente Estação

Posteriormente, caso um agente Utilizador se encontre na área de proximidade de algum agente Estação, este pode efetuar uma oferta de desconto para o utilizador deixar lá a sua bicicleta. Neste caso, serão utilizadas as *performatives negotiation* uma vez que poderá existir um processo de negociação de descontos.

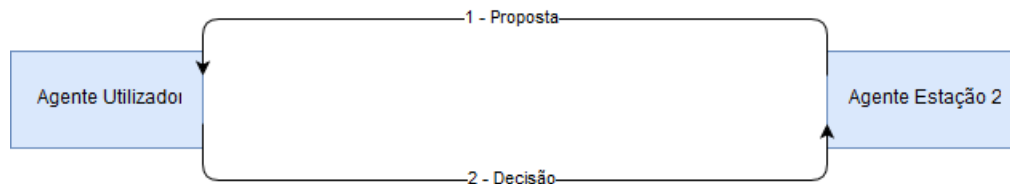


Figura 9: Comunicação 3 - Agente Utilizador - Agente Estação

Por último, sempre que um agente Utilizador atinge o seu destino final pretendido, informa a respetiva estação que vai deixar lá a sua bicicleta e é eliminado do sistema.

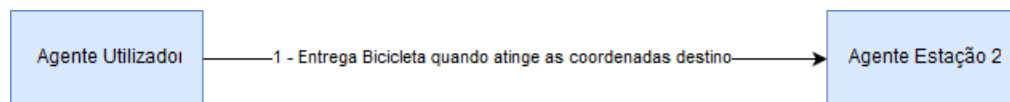


Figura 10: Comunicação 4 - Agente Utilizador - Agente Estação

## 6 Proposta de Solução Inicial

De modo a dar início a um processo de análise de resultados e possibilitar a obtenção de melhorias no sistema, é necessário dar um conjunto de valores iniciais para as interações dos diferentes agentes.

Em primeiro lugar, foi considerando que cada estação apresenta uma área de proximidade com um raio de **5000m**. Este valor poderá ser ajustado ao longo do tempo devido à interação entre o agente Gestor de Estações e o agente Estação.

Por outro lado, é fundamental identificar um conjunto de regras de funcionamento para os diferentes agentes, mais concretamente para os agentes Estação e agentes Utilizadores. É importante decidir, ainda de que uma forma muito simples, quais as ofertas que serão feitas pelos AE e qual a que AU decide qual aceitar.

### 6.1 Propostas dos Agentes Estação

Como referido anteriormente, sempre que um agente Estação pretende efetuar uma oferta de modo a aliciar um utilizador a aceitar colocar lá a sua bicicleta, este deve ter em consideração a *distância do utilizador à estação* em questão e o *nível de utilização da estação*. Segue-se então uma tabela com os valores iniciais que o grupo achou adequados para uma versão inicial do sistema.

Utilizador	Estação	Desconto
<u>Distância à Estação</u>	<u>Nível de preenchimento</u>	
> 1000m	baixo	80%
< 1000m	baixo	70%
> 1000m	médio	15%
-	médio	0%
-	alto	0%

Figura 11: Modelo inicial de ofertas sugeridas pelos AE

### 6.2 Análise do Agente Gestor de Estações

O agente Gestor de Estações deve apresentar a capacidade de receber mensagens com dados referentes a cada um dos agentes Estação, analisa-los e informar o que este deve alterar nas suas características.

Assim, foi necessário elaborar um modelo com as várias respostas que o agente Gestor de Estações pode enviar face às informações recebidas:

Fator de Análise	Valor	Decisão do Agente Gestor
Nível de Preenchimento	1 (baixo)	Aumentar o Stock em 2 unidades
Nível de Preenchimento	2 (médio)	Aumentar o Stock em 1 unidade
Nível de Preenchimento	3 (alto)	Decrementar o Stock em 2 unidades
Descontos Recusados > 2* Descontos Aceites	true	Aumentar o valor de Desconto
Descontos Recusados < 2* Descontos Aceites	true	Diminuir o valor do Desconto
2* Total Incentivos < Capacidade Máxima	true	Aumentar o Raio de Proximidade da Estação em 500m

Figura 12: Modelo de Análise de Informação do Agente Gestor de Estações

### 6.3 Ações dos Agentes Utilizadores

Perante as ofertas dos diferentes AE's, o agente Utilizador tem que revelar a capacidade de escolher da melhor forma, de acordo com as suas características, informações gerais e analisando a oferta recebida. Deste modo, os fatores de decisão de aceitação ou recusa de uma determinada oferta são: *distância à estação, condição física, idade, estação do ano, hora, luminosidade, humidade e valor do desconto*.

Devido a um elevado número de atributos(8) , o número de diferentes casos seria de 40320, sendo algo de uma extensa execução e difícil apresentação. Posto isto, o grupo decidiu definir pontuações a cada um dos atributos dando assim um nível de 1 a 3.

		Desfavorável	Aceitavel	Favorável
		Nível 1	Nível 2	Nível 3
Utilizador	<u>Distância À Estação</u>	>2000	1000<2000	<1000
	<u>Condição Física</u>	baixa	média	alta
	<u>Idade</u>	>50	31-49	18-30
Informação Geral	<u>Estação do Ano</u>	Inverno	Primavera/Outono	Verão
	<u>Hora</u>	22h-6h	6h-14h	14h-22h
	<u>Luminosidade</u>	nula	média	alta
	<u>Humidade</u>	alta	média	media

Figura 13: Modelo inicial de níveis de aceitação e parâmetros correspondentes

A partir de cada nível atribuído procede-se à média dando então um valor entre 1 e 3 podendo este ter casas decimais. De seguida expõem-se a correlação entre a pontuação final e a percentagem de desconto.

Nível Médio	Desconto
<u>1.7</u>	<u>&gt;75%</u>
1.7-2.4	50%-74%
2.4-3	10%-49

Figura 14: Correlação entre Nível Médio Calculado e Desconto Aceitável

Com esta estratégia implementada mantém-se a diversidade de casos e simultaneamente todo o algoritmo é simplificado, continuando com a coerência e equilíbrio entre atributos.

## 7 Gestão de Riscos/Possíveis Soluções

Após analisado e interpretado o funcionamento do sistema a ser implementado, procedeu-se a um processo de identificação de possíveis falhas que este pode apresentar. Como o objetivo primordial do sistema é garantir o equilíbrio do uso de bicicletas nas diferentes estações, torna-se importante encontrar um conjunto de soluções que visam corrigir os problemas mais frequentes:

- Excesso/Escassez de bicicletas numa determinada estação
  - Através da comunicação periódica entre o agente Gestor de Estações e os diferentes agentes Estação, este problema pode ser corrigido colocando/retirando bicicletas na estação em questão.
- Número reduzido de alugueres numa estação
  - Este tipo de falha pode acontecer por dois motivos: a oferta não é suficientemente aliciante ou poucos utilizadores estão a aceder à área de proximidade da estação em causa. O problema pode ser corrigido informando os agentes Estação para alterarem as suas ofertas ou aumentarem o raio da sua área de proximidade, de modo a atingir um maior número de utilizadores.
- Número elevado de ofertas aceites
  - Esta situação pode levar não só à sobrelocação de uma dada estação, como também à obtenção de um baixo rendimento financeiro para o gestor. Assim, sempre que se verificar um número elevado de ofertas aceites, estes valores devem ser otimizados, de modo a atingir um equilíbrio entre o número de bicicletas alugadas e o lucro obtido.
- Número elevado de ofertas recusadas
  - Neste caso o agente estação deve ser informado para melhorar as condições que está oferecer aos utilizadores.



## 8 Implementação

Na fase de desenvolvimento, o grupo decidiu que a implementação dos diferentes agentes iria ser feita através de JADE. JADE é uma plataforma implementada em Java, utilizada para o desenvolvimento de agentes inteligentes. Um sistema Jade suporta a coordenação entre diversos agentes FIPA e providencia uma comunicação FIPA-ACL, facilitando assim o processo de comunicação e providenciando um sistema de detenção.

### 8.1 Decisões Tomadas/Estratégias Implementadas

Houve a necessidade prévia de decidir as localizações e quantidade de agentes Estação estáticos que iriam existir para conseguirmos testar todo o sistema. Como tal, foram definidas 5 estações ao longo de uma área geográfica, simulando criando assim um pequeno "ecossistema".

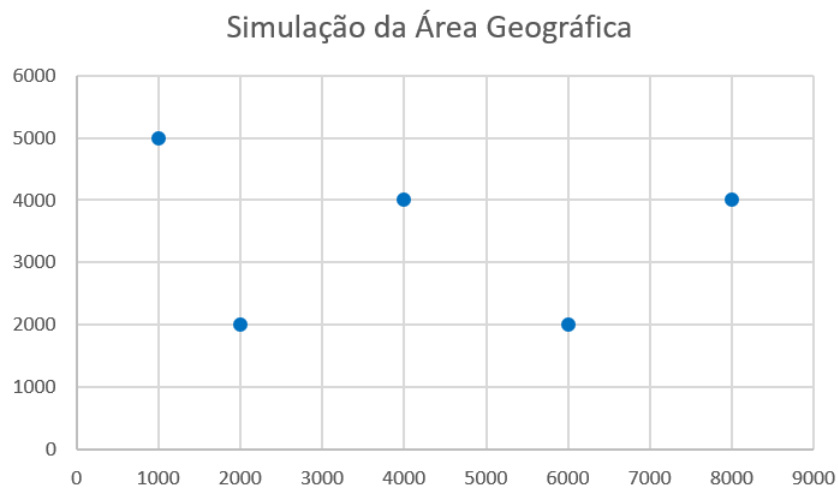


Figura 15: Ilustração do posicionamento geográfico das estações para efeitos de teste

### 8.2 Agente Utilizador

Na implementação do Agente Utilizador houve uma série de necessidades fundamentais para o conseguir tornar mais realista e imprevisível. Deste modo, no setup, todas as suas características que posteriormente iram afetar a sua decisão relativamente a uma proposta, foram definidas aleatoriamente.

Seguidamente, foram adicionados diversos *Behaviours* que permitem representar o comportamento necessário para o funcionamento deste agente.

[BEHAVIOURS]

```
/* Pesquisa as Estações que existem no sistema */  
this.addBehaviour(new searchEstacoes());
```

```
/* recebe informação geral && propostas */  
this.addBehaviour(new receiveMsgs());
```

```
/* muda de posicao e informa as estações de 1 em 1s */  
this.addBehaviour(new novaPos(this, 200));  
this.addBehaviour(new informPos(this, 200));
```

A pesquisa das Estações é necessária quer para o utilizador ter conhecimento de quem informar, quer para saber onde estas se localizam. Como todas estações são criadas inicialmente, este comportamento só será executado uma vez, não havendo assim alterações ao longo do ciclo de vida do sistema.

O *receiveMsgs()* terá três propósitos:

- Em primeiro lugar, irá receber mensagens periódicas, do tipo *performative* **INFORM**, contendo informação geral relativamente as condições climáticas e outros fatores que influenciam a sua decisão.
- Em segundo lugar, utilizando o mesmo tipo de *performative*, pode receber uma mensagem da estação inicial, informando que não existe bicicleta em stock para ele utilizar, eliminado em seguida o utilizador.
- Em terceiro lugar, através de uma mensagem do tipo *performative* **CFP**, o utilizador estação pode enviar uma proposta ao utilizador para este mudar a sua rota, à qual este responde se aceita ou não. A aceitação ou negação de uma proposta, será respondida também utilizando mensagem ACL: caso seja aceite, será utilizado o tipo **ACCEPT PROPOSAL**; caso seja negada, será do tipo **REFUSE**.

[Excerto da tomada de decisão sobre a proposta]

```
if (decisao.equals("SIM")) {
    atualiza_destino(msg.getSender().getLocalName(),
        Integer.parseInt(tokens[1]),
        Integer.parseInt(tokens[2]));
    ACLMessage msg1 = new ACLMessage(ACLMessage.ACCEPT_PROPOSAL);
    System.out.println("ACEITEI A PROPOSTA -> " + factor_total);
    msg1.addReceiver(msg.getSender());
    msg1.setContent(decisao);
    send(msg1);
    flag = 1;
}

if (decisao.equals("NAO")) {
    ACLMessage msg1 = new ACLMessage(ACLMessage.REFUSE);
    System.out.println("RECUSEI A PROPOSTA ->" + factor_total);
    msg1.addReceiver(msg.getSender());
    msg1.setContent(decisao);
    send(msg1);
}
```

De seguida, decidiu-se que os utilizadores iriam iniciar o seu ciclo de vida numa das estações existentes, tendo como destino final inicial uma outra estação. Para efeitos de teste, um utilizador percorre a distância primeiramente em diagonal até encontrar um dos eixos da estação final, realizando o resto do percurso em modo retilíneo.



Figura 16: Percurso com início na Estação1 e destino Estação2

Ao longo do percurso, o utilizador pode receber propostas de descontos e, caso aceite, deve alterar a sua rota e destino final. Tal proposta é enviada apenas se o utilizador se entrar numa área circular à volta de uma estação (raio de alcance da Estação para envio de propostas).

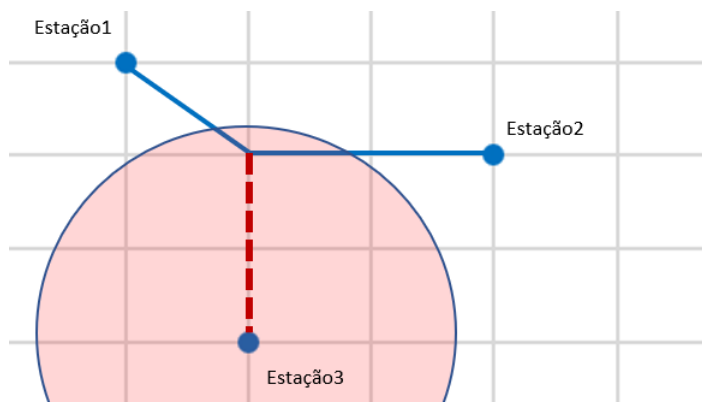


Figura 17: Ilustração da proposta realizada pela Estação3 e aceitação por parte do Utilizador

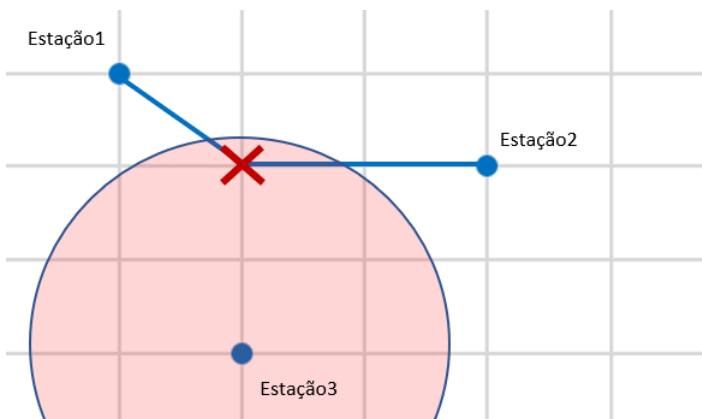


Figura 18: Ilustração da proposta realizada pela Estação3 e negação por parte do Utilizador

Para complementar todas estas decisões, houve também a necessidade de especificar o período de tempo que o utilizador necessita para se deslocar de uma posição atual para a seguinte. Foi então definido que o utilizador se movimenta de 200 em 200 milissegundos.

### 8.3 Agente Gestor de Estações

Como já referido anteriormente, o propósito da implementação deste agente passa por conseguir realizar um controlo periódico a cada uma das estações de modo a melhorar o rendimento destas.

Deste modo, o agente encontra-se constantemente à espera de mensagens dos Agentes Estação para poder analisar as informações recebidas e assim informa-los das alterações que devem fazer.

Para implementar esta situação em JADE foi necessário recorrer a um *CyclicBehaviour*:

```
/* Recebe mensagens dos Agentes Estação e analisa-as */  
this.addBehaviour(new analisarEstacao());
```

Este *behaviour* fica constantemente à espera de mensagens com o seguinte formato:

```
ACL Performative: INFORM  
"nivelPreenchimento;capacidadeTotal;descontosAceites;descontosRecusados"
```

Após receber a mensagem, o agente invoca a função *analisar* que recebe como argumento os quatro valores enviados na mensagem. Esta função atua de acordo com os dados apresentados na tabela 12, devolvendo uma *string* com a informação que deve passar ao agente Estação.

Apresenta-se então uma tabela com os códigos de mensagens que este pode enviar de resposta. É importante salientar que podem ser usados um ou mais códigos em cada mensagem, separados por ponto e vírgula.

Mensagem	Significado
"+2B;"	Aumentar o Stock em 2 unidades
"+1B;"	Aumentar o Stock em 1 unidade
"-2B;"	Decrementar o Stock em 2 unidades
"-1B;"	Decrementar o Stock em 1 unidade
"+D;"	Aumentar 5% ao valor de Desconto
"-D;"	Diminuir 5% o valor do Desconto
"+R;"	Aumentar o Raio de Proximidade da Estação em 500m

Figura 19: Mensagens enviadas pelo Agente Gestor

Por último, a mensagem é enviada para a estação e o gestor atualiza o valor do número de bicicletas livres que tem stock.

### 8.4 Agente Estação

Para a implementação do Agente Estação, em primeiro lugar foi necessário registar cada um deles no DFService para que posteriormente possam ser encontrados pelos diversos utilizadores. Para efeitos de teste, foi definido que estes teriam um stock inicial de 10 bicicletas e uma capacidade máxima de 12.

Em seguida, implementou-se o conjunto de *behaviours* que permitem modelar todo o comportamento deste agente.

```
/* De 1 em 1s, envia os seus dados para o Agente Gestor de Estações */  
this.addBehaviour(new enviarStatusGestor(this, 1000));  
  
/* Recebe mensagens (do utilizador ou do gestor) e atualiza os seus dados */  
this.addBehaviour(new atualizaDados());  
  
/* de 200ms em 200ms, envia os seus dados para o Agente Interface */  
this.addBehaviour(new TickerBehaviour(this, 200) {
```

```

protected void onTick() {
    myAgent.addBehaviour(new enviarStatusInterface());
}
});

```

Os *behaviours* enviarStatusGestor e enviarStatusInterface servem simplesmente para enviar uma mensagem periódica, do tipo INFORM, contendo os dados relativos ao agente Estação.

Por outro lado, o *behaviour* atualizaDados recebe mensagens dos agentes Utilizador e do agente Gestor de Estações. Seguidamente, através de pseudo-código. Apresenta-se uma modelação do algoritmo que descreve comportamento deste *behaviour*, exemplificando as diversas ações do agente.

```

ACLMMessage msg = receive();

if (msg != null) {
    if (msg do tipo INFORM) {
        if(msg enviada pelo utilizador) {
            distancia = verificaDistancia();

            if (distancia < raio proximidade) {
                desconto = calculaProposta();
                enviaProposta(desconto);
            }

        }

        if (msg enviada pelo gestor) {
            analisaMsg();
            alteraDados();
        }
    }

    if (msg do tipo REFUSE) {
        /* utilizador recusou proposta */
        adicionaUtilizadorRecusado();
        descontosRecusados++;
    }

    if (msg do tipo ACCEPT) {
        /* utilizador aceita proposta */
        descontosAceites++;
    }
}

```

## 8.5 Agente de Informação Geral

Este agente tem como objetivo a simulação de vários aspetos do ambiente envolvente, de forma a consciencializar os utilizadores. Para cumprir os objetivos, foi necessário implementar comportamentos que variem ciclicamente os valores da estação do ano, horas, luminosidade e humidade.

Por outro lado, é ainda necessário informar todos os agentes do tipo Utilizador das variações dos valores dos parâmetros. Deste modo, torna-se essencial pesquisar os agentes utilizadores existentes no sistema e direcionar as mensagens para os mesmos.

[BEHAVIOURS]

```
/* Procura e informa os utilizadores das condições do ambiente */
this.addBehaviour(new informaUtilizadores(this, 1000));

/* Atualiza estação do Ano de 10 em 10 min */
this.addBehaviour(new atualizaEstacaoDoAno(this, 1800000));

/* Atualiza hora, luminosidade e humidade de 1 em 1 segundo */
this.addBehaviour(new atualizaHora(this, 1000));
this.addBehaviour(new atualizaLuminosidade(this, 1000));
this.addBehaviour(new atualizaHumidade(this, 1000));
```

Foi relevante a definição do comportamento paralelo, na altura de enviar as mensagens aos Agentes utilizador, de forma a garantir a coerência dos dados enviados aos vários utilizadores. Uma vez que estes atributos variam em gamas de tempo diferentes, o grupo definiu uma escala proporcional para a alteração destes mesmos, permitindo assim testar a aplicação.

## 8.6 Agente Interface

Houve a necessidade de implementar um agente intermédio entre os agentes estações e a camada Interface, conseguindo assim aglomerar os dados do sistema no seu todo e posterior visualização. Para tal, as ações necessárias são: perceção das estações existentes, receção das mensagens com informação por parte das mesmas e a posterior divulgação para a camada Interface.

[BEHAVIOURS]

```
/* Procura e guarda todas as estações registadas */
this.addBehaviour(new searchEstacao());

/* serve para receber todas as mensagens do agente
Estação, contendo todos os dados importantes.*/
this.addBehaviour(new EstacaoRequest());

/* envia e atualiza os dados da Interface de 0,2 em 0,2 segundos */
this.addBehaviour(new TickerBehaviour(this, 200) {
protected void onTick() {
myAgent.addBehaviour(new showInterface());
}
});
```

## 9 Interface

Para a visualização dos dados do sistema, foi criada uma interface em que se consegue ver todos os diferentes parâmetros e suas devidas oscilações.

Em seguida apresentam-se os diferentes gráficos que a interface mostra ao Utilizador, acompanhados da sua respetiva descrição.

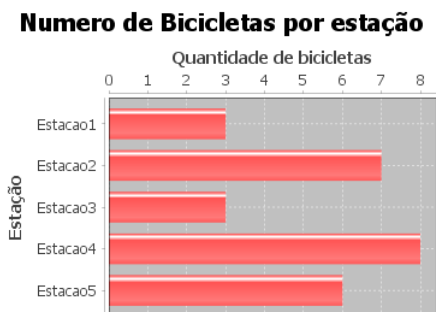


Figura 20: Gráfico Quantitativo - Necessário para a perceção de stock em cada uma das estações.

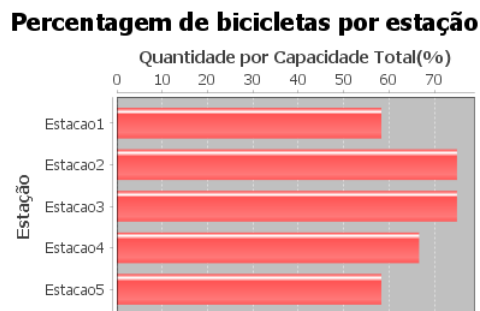


Figura 21: Gráfico Percentual - Utilizado para saber a percentagem de ocupação relativamente à capacidade da estação, crucial para a gestão de recursos.

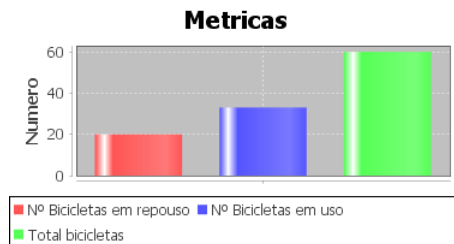


Figura 22: Gráfico de Métricas - Possibilita saber onde se encontram os recursos de todo o sistema e a capacidade do mesmo.



Figura 23: Gráfico de Utilizadores - Possibilita saber quantos utilizadores estão no sistema ao longo do tempo.

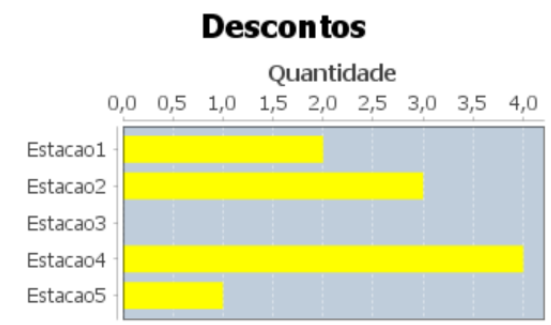


Figura 24: Gráfico de Descontos - Necessário para saber o número de descontos aceites, ou seja, número de vezes que utilizador muda o seu destino.

Para além destes gráficos, a interface disponibiliza um botão para cada Estação existente, de maneira a poder ser visível características individuais de cada uma delas..



## 10 Resultados

Após a conclusão do desenvolvimento do sistema, este mesmo foi testado com várias amostras de diferentes tamanhos. De forma a dificultar/facilitar a tarefa do sistema, foram feitos vários testes alterando os seguintes valores:

- Quantidade de utilizadores em simultâneo no sistema;
- Stock das estações relativamente à capacidade das mesmas;
- Raio de Proximidade das estações;
- Número de bicicletas livres na posse do Agente Gestor de Estações;
- Recursos iniciais de cada estação;
- Alterações das ofertas dos Agentes Estação;

Posto isto, foi então verificado que o sistema consegue manter o seu equilíbrio para todos os diferentes cenários, evitando ocupações acima dos 70% ou menores que 30% em cada estação. Por outro lado, o número de utilizadores que foram influenciados e alteraram as suas rotas ronda sempre 60% do total da população utilizada nas experiências.

É também importante referir que a percentagem de propostas aceites nem sempre se dividem proporcionalmente por todas as estações existentes, existindo algumas onde a percentagem de êxito é superior devido à sua localização face ao aglomerado de utilizadores.

## 11 Vulnerabilidades/Limitações do Sistema

### Soluções Implementadas

Após a implementação do sistema e da realização dos diversos testes, foram encontradas algumas possíveis vulnerabilidades no nosso sistema que teriam de ser solucionadas.

Um utilizador, já no decorrer do seu percurso, pode aceder a áreas de propostas de várias estações mas, após aceitar uma proposta, é obrigado a dirigir-se para essa estação, não podendo receber qualquer outra proposta. Com esta restrição impedimos que um utilizador fique infinitamente a deslocar-se entre as várias estações.

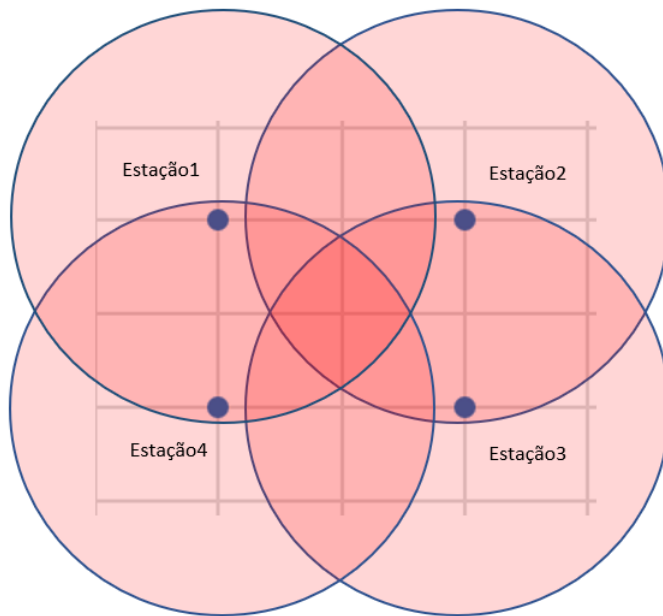


Figura 25: Ilustração das áreas de propostas sobrepostas

Como podemos observar, um utilizador que tivesse como plano inicial fazer o percurso desde a Estação4 para a Estação2 iria entrar numa área de múltiplas propostas feitas por diferentes estações. Assumindo que o seu fator de decisão seria recetivo a todas as estações, o agente estaria constantemente a mudar de direção, ficando infinitamente no sistema.

Por outro lado, se houver uma grande percentagem de utilizadores que saem de uma estação e, posteriormente, através de propostas, decidam acabar na mesma estação que iniciaram o percurso, nunca serão transferidos por meio de propostas para outras estações.

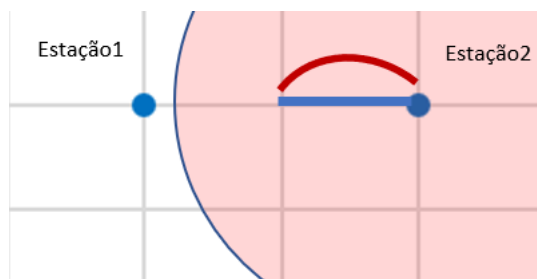


Figura 26: Ilustração da proposta aceite por parte da Estação onde o utilizador iniciou o percurso

Para além das situações exemplificadas, conseguimos ainda identificar problemas quando as estações se encontram nas extremidades do mapa. Estas estações têm muita dificuldade a aliciar utilizadores a mudarem a sua rota. Mesmo após melhorar as percentagens de desconto e o raio de captação de utilizadores, o seu nível de propostas aceites continua baixo em comparação às propostas recusadas.

## 12 Conclusões e Trabalho Futuro

A realização deste trabalho envolveu todo o processo de caracterização necessário ao desenvolvimento de um sistema multiagentes para a resolução de um problema de Sistemas de Partilha de Bicicletas (SPB) e a sua implementação. Em primeiro lugar foi feita uma contextualização do problema que permitiu compreender melhor os domínios concretos em que o problema está inserido, bem como dos vários atores que constituem o sistema. Por outro lado, esta especificação permitiu-nos concluir que o modelo a ser implementado corresponde a um sistema multiagentes aberto e colaborativo.

Em seguida foram definidos os conceitos chave do sistema e elaborado um estado da arte sobre os agentes e suas aplicações, abordando as suas diferentes propriedades e características.

Identificados todos os agentes essenciais para a aplicação, o passo seguinte passou por desenhar um esquema que demonstrasse a arquitetura do sistema proposto. Deste modo, foi possível assimilar as diferentes interações entre os agentes, auxiliando também na elaboração dos protocolos de comunicação, onde foi estabelecida uma conexão entre as diferentes comunicações e as *performatives* da norma FIPA/ACL.

O grupo elaborou uma solução com valores iniciais de modo a permitir interpretar uma primeira reação do sistema e assim compreender os pontos onde este poderia ser melhorado no futuro. Com base em vários testes realizados, foram identificadas várias lacunas que podem acontecer frequentemente neste tipo de sistemas. Assim, foi necessário desenvolver um conjunto de soluções para a resolução destes problemas, garantindo a correta execução do sistema.

Com toda a estratégia delineada, passou-se então à implementação dos agentes identificados para a aplicação. Todas as comunicações e interações anteriormente identificadas, foram modeladas representando os diversos comportamentos de cada agente.

Por último foram analisados os resultados obtidos, sendo estes alcançados com sucesso uma vez que todo o equilíbrio do sistema foi garantido.

O grupo sente que o objetivo de consolidar os conceitos aprendidos ao longo do semestre foram cumpridos com êxito. Futuramente, o grupo pretende estender este trabalho a outras ferramentas também lecionadas, como JESS ou JADEX. A decisão da utilização do Jade como ferramenta para a realização deste trabalho deve-se ao facto desta ser uma ferramenta bastante completa, bem documentada, onde o grupo se pode apoiar para o desenvolvimento deste trabalho prático. O FIPA-ACL também facilitou a execução do trabalho pois permite a realização de comunicações desde de um nível de complexidade baixo até a um alto nível de complexidade.

## 13 Referências Bibliográficas

1. Bellifemine, F. L., Caire, G., Greenwood, D. (2007). *Developing multi-agent systems with JADE (Vol.7)*. John Wiley Sons.
2. Contardo, C., Morency, C., Rousseau, L.-M. (2012). *Balancing a dynamic public bike-sharing system (Vol. 4)*. Cirrelt Montreal.