



Universidade do Minho

Mestrado Integrado em Engenharia Informática

# Agentes Inteligentes

Grupo 2

## Trabalho Prático

*José Diogo Xavier Monteiro, A83638*

*Benjamim Oliveira, PG42815*

*Gonçalo Borges, PG42643*

Agentes Inteligentes

4º Ano, 1º Semestre

Departamento de Informática

3 de janeiro de 2021

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>4</b>
<b>2</b>	<b>Agentes</b>	<b>5</b>
2.1	Agente Estação . . . . .	5
2.2	Agente Utilizador . . . . .	5
2.3	Agente Interface . . . . .	5
2.4	Agente Gestor . . . . .	6
<b>3</b>	<b>Interação dos Agentes</b>	<b>7</b>
3.1	Utilizador - Gestor . . . . .	7
3.1.1	Utilizador envia Posição Atual . . . . .	7
3.1.2	Gestor responde com Estação Próxima . . . . .	8
3.2	Utilizador - Estação . . . . .	8
3.2.1	Utilizador Requisita o Aluguer . . . . .	9
3.2.2	Estação Confirma o Pedido . . . . .	9
3.2.3	Utilizador informa a sua Presença . . . . .	10
3.2.4	Estação envia Proposta . . . . .	10
3.2.5	Utilizador responde à Proposta . . . . .	10
3.3	Estação - Interface . . . . .	10
<b>4</b>	<b>Arquitetura</b>	<b>12</b>
<b>5</b>	<b>Contexto Prático</b>	<b>14</b>
5.1	Diagrama de Classes . . . . .	14
5.2	Inicialização . . . . .	14
5.3	Agentes e respetivas funções . . . . .	14
5.3.1	Agente Interface . . . . .	15
5.3.2	Agente Gestor . . . . .	15
5.3.3	Agente Estação . . . . .	16
5.3.4	Agente Utilizador . . . . .	16

	3
6 Interações	18
7 Conclusão	21

# 1 Introdução

No âmbito da unidade curricular de Agentes Inteligentes, concebemos uma arquitetura distribuída baseada em agentes para um sistema de partilha de bicicletas.

Este sistema de partilha de bicicletas tem-se tornado bastante popular nomeadamente em metrópoles devido à facilidade em alugar e devolver, neste caso, uma bicicleta e devido a ser prático.

Um utilizador pode requisitar uma bicicleta e posteriormente efetuar a sua devolução. A requisição e devolução é feita em estações dedicadas a este fim. Por norma as estações estão distanciadas 100 metros e cada estação possui um número máximo fixo de bicicletas que pode armazenar.

Contudo este sistema sofre de um problema que visaremos em solucionar. Trata-se do re-equilíbrio de partilha de bicicletas, ou seja, estações que ficam sobrelotadas e consequentemente estações com falta de bicicletas. Afim de desenhar esta arquitetura, delegamos funções e ações distintas para os agentes apresentados ao longo do relatório.

## 2 Agentes

Neste ponto serão descritos os agentes no que toca a função e funcionamento dos mesmos. Utilizamos os três agentes sugeridos pela equipa docente (Agentes Estação, Utilizador e Interface), no entanto consideramos que seria benéfico ao desenvolvimento do nosso trabalho o acréscimo de um quarto agente, o agente gestor. Passamos então á descrição e apresentação dos mesmos.

### 2.1 Agente Estação

Este agente, como o próprio nome indica é referente á estação. Neste projeto representa o que pode ser considerado como um dos pontos cruciais no problema de equilíbrio que pretendemos simular.

Cada estação possui uma APE, ou **Área de Proximidade de uma Estação**. Um Agente Estação pode, no caso de ter falta de bicicletas na sua estação, tentar aliciar utilizadores em viagem a terminarem a mesma e deixarem a sua bicicleta na estação. Para isto o Agente Estação tem a possibilidade de fazer descontos sobre o preço da viagem, ajustando o desconto conforme a necessidade (ou falta dela) de aumentar o número de bicicletas na estação.

### 2.2 Agente Utilizador

Com o Agente Utilizador simulamos a atuação de uma pessoa que utilize um SPB (Sistema de Partilha de Bicicletas). No caso simulamos o comportamento que o mesmo teria usando um *smartphone*, fornecendo constantemente dados de localização ao Agente Gestor. A partir do momento que este estiver dentro da APE de uma estação, poderá aceitar descontos propostos por Agentes Estação em troca de terminar a viagem prematuramente, deixando a bicicleta antes do destino objetivo.

### 2.3 Agente Interface

Este agente é utilizado com o intuito de nos permitir visualizar, ao longo do tempo de execução, o estado do sistema em termos de equilíbrio.

Para este ponto foi desenvolvida uma interface de visualização gráfica.

## **2.4 Agente Gestor**

Tendo em conta o alto número de mensagens que seriam trocadas entre os agentes Utilizador e Estação decidimos implementar um agente que servisse como intermediário. Assim, ao invés de existirem comunicações de todos os agentes utilizador com todos os agentes estação temos o Agente Gestor a receber localizações de todos os agente e a informar os mesmo quando estes se encontrarem dentro de alguma APE. Nesse caso o Utilizador irá então comunicar diretamente com a estação, ou vice-versa.

### 3 Interação dos Agentes

Esta secção do nosso trabalho sofreu bastantes alterações aquando da sua implementação. Como base, aqui, idealizamos algumas das interações entre agentes que presumimos serem ideais ao funcionamento do projeto, mas, tal como referido, pouco se manteve igual. O objetivo desta secção será que o leitor entenda os vários tipos de comunicação base que idealizamos, tal como a sua necessidade. Para tal, vamos tentar explicar todos de forma sucinta.

#### 3.1 Utilizador - Gestor

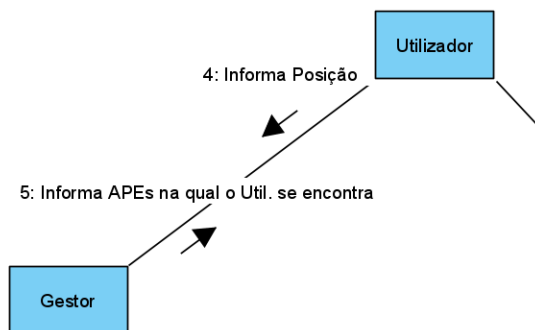


Figura 1: Comunicação entre Utilizador e Gestor

Tal como explicado anteriormente, o Gestor é um Agente cuja única função é servir de intermediário entre o utilizador e todas as estações pelas qual ele passar. Para esse efeito, este irá receber continuamente comunicação da nova localização de todos os Agentes Utilizadores que se encontrem em viagem:

##### 3.1.1 Utilizador envia Posição Atual

A partir do momento que o Agente Utilizador aluga uma bicicleta até ao momento em que a entrega, irá, tal como referido anteriormente, comunicar a sua posição ao Agente Gestor. O Gestor, por sua vez, utilizará a nova localização de todos os Utilizadores que com ele

comunicaram e verificar se encontram dentro de uma zona APE.

Performative: INFORMATIVE

### **3.1.2 Gestor responde com Estação Próxima**

Quando o Agente Gestor finalmente deteta que o Utilizador se encontra dentro da zona APE de uma ou mais estações, passa a enviar uma resposta ao Utilizador com a informação das mesmas Estações. Assim, este pode passar à fase de comunicação direta, em que vai conseguir receber as propostas vindas da Estação.

Performative: INFORMATIVE

## **3.2 Utilizador - Estação**

Todos estes tipos de comunicação entre Utilizador e Estação, ocorrerão em 2 situações diferentes. A primeira, será quando o Utilizador for gerado aleatoriamente numa das estações e terá de proceder a pedir o aluguer de uma bicicleta. O segundo, será quando o Utilizador que prossegue em viagem se encontra na zona APE da Estação.



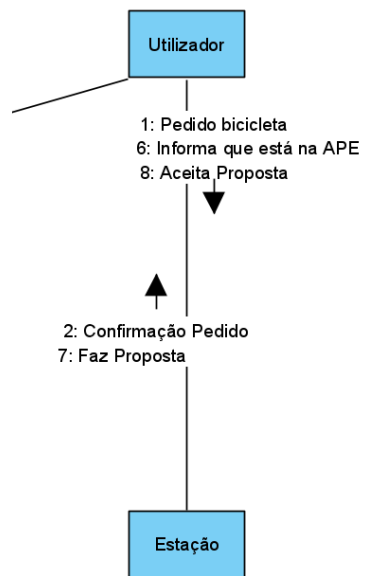


Figura 2: Comunicação entre Utilizador e Estação

### 3.2.1 Utilizador Requisita o Aluguer

Quando o Agente Utilizador é gerado com a localização de um Agente Estação aleatório terá de proceder a fazer-lhe um pedido do aluguer de uma bicicleta.

Performative: REQUEST

### 3.2.2 Estação Confirma o Pedido

Após receber o REQUEST de uma bicicleta por parte de um Agente Utilizador, o Agente Estação terá de verificar se tem bicicletas de vago. Caso existam, responderá ao Utilizador que pode proceder ao aluguer, caso não existam terá de adicionar o Utilizador a uma queue e informar que terá de esperar até que haja uma nova vaga.

Performative: ACCEPT\_PROPOSAL // REJECT\_PROPOSAL

### 3.2.3 Utilizador informa a sua Presença

Após o Agente Utilizador, que segue viagem, ser informado que se encontra dentro da zona APE de um Agente Estação, irá proceder à fase de comunicação direta. Basicamente, o Utilizador irá informar a estação que se encontra na sua zona, para que assim possa receber proposta de entrega e descontos.

Performative: INFORMATIVE

### 3.2.4 Estação envia Proposta

O Agente Estação, utilizando a informação sobre todos os Utilizadores existentes na sua APE e sobre o número de bicicletas livres que possui, irá calcular um possível valor de desconto que estará disposta a oferecer ao Utilizador de forma a que este disponibilize/entregue a sua bicicleta. Mediante a necessidade de bicicletas, este valor de desconto poderá aumentar ou diminuir.

Performative: REQUEST

### 3.2.5 Utilizador responde à Proposta

Uma vez recebido o desconto/proposta da estação o utilizador irá então decidir se aceita ou não o mesmo. Para tal deverá ter em conta a distância que falta percorrer vs o desconto a receber e assim tomar uma decisão, informando a estação da mesma.

Performative: ACCEPT\_PROPOSAL // REJECT\_PROPOSAL

## 3.3 Estação - Interface

A Interface é responsável por manter atualizado o número de bicicletas de cada estação. Sempre que uma viagem é iniciada ou terminada, o número de bicicletas é ajustado pela Interface de forma a manter um registo da capacidade de cada Estação.

Performative: INFORMATIVE

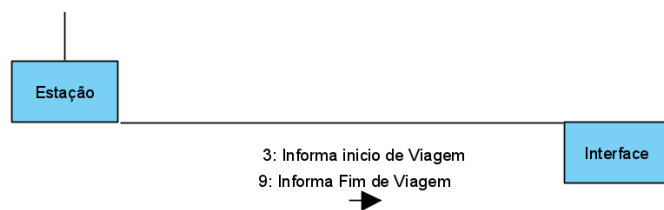


Figura 3: Comunicação entre Estação e Interface

## 4 Arquitetura

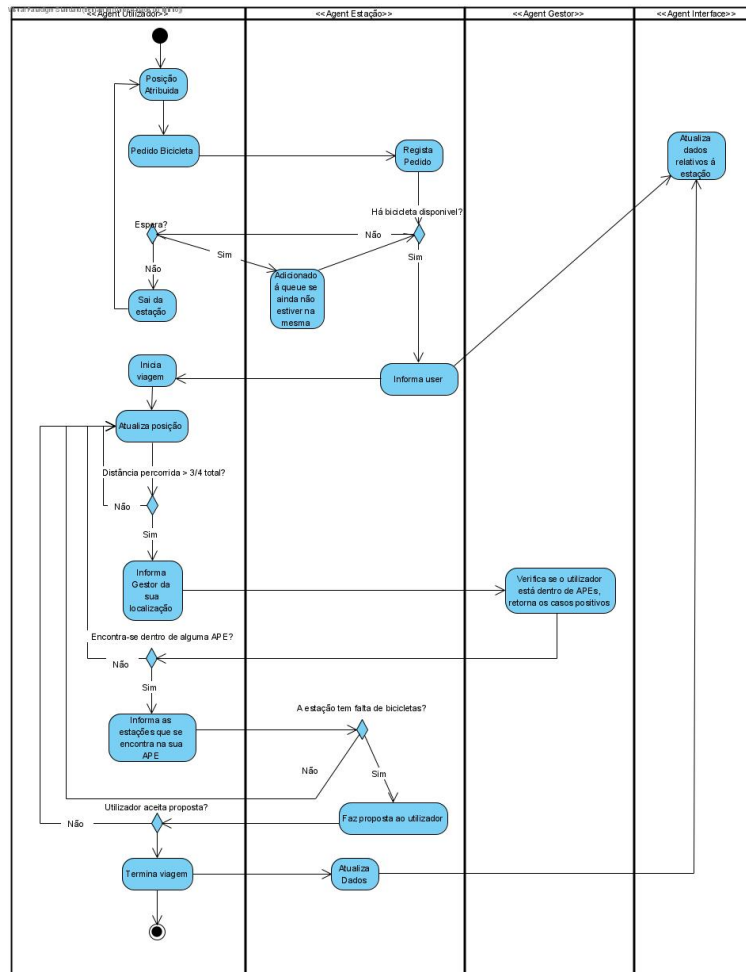


Figura 4: Diagrama de Atividades

No diagrama de atividades em cima apresentado, podemos analisar o processo End-To-End da requisição de uma bicicleta. Quando o utilizador requisita uma bicicleta, a sua posição é guardada e analisada. A Estação processa um preço final expectado e se o pedido for efetuado, este é registado.

É feita uma verificação de bicicletas disponíveis. Se houver disponibilidade de bicicletas, o utilizador é informado e este inicia a viagem, o número de bicicletas nessa estação é

atualizado na Interface. Caso não existam bicicletas, o utilizador é adicionado a uma fila de espera à qual poderá abandonar.

Depois de iniciar a viagem, quando o utilizador se encontra na APE de uma Estação, é enviada uma proposta ao utilizador para que este conclua a sua viagem nesta Estação.

Caso o Utilizador não se encontre numa APE, a sua posição é atualizada. Caso se encontre, as estações são informadas que se encontram na sua APE.

Caso a Estação em consideração tenha falta de bicicletas, é feita uma proposta ao Utilizador para devolver a bicicleta nessa estação. O Utilizador pode aceitar ou recusar. Em caso de recusa, a posição do utilizador volta a ser atualizada e o processo repete-se até encontrar uma estação disponível a receber a bicicleta. Caso aceite, a viagem é terminada, a Estação atualiza os dados de término da viagem e a Interface atualiza o número de bicicletas presentes nessa estação.

## 5 Contexto Prático

### 5.1 Diagrama de Classes

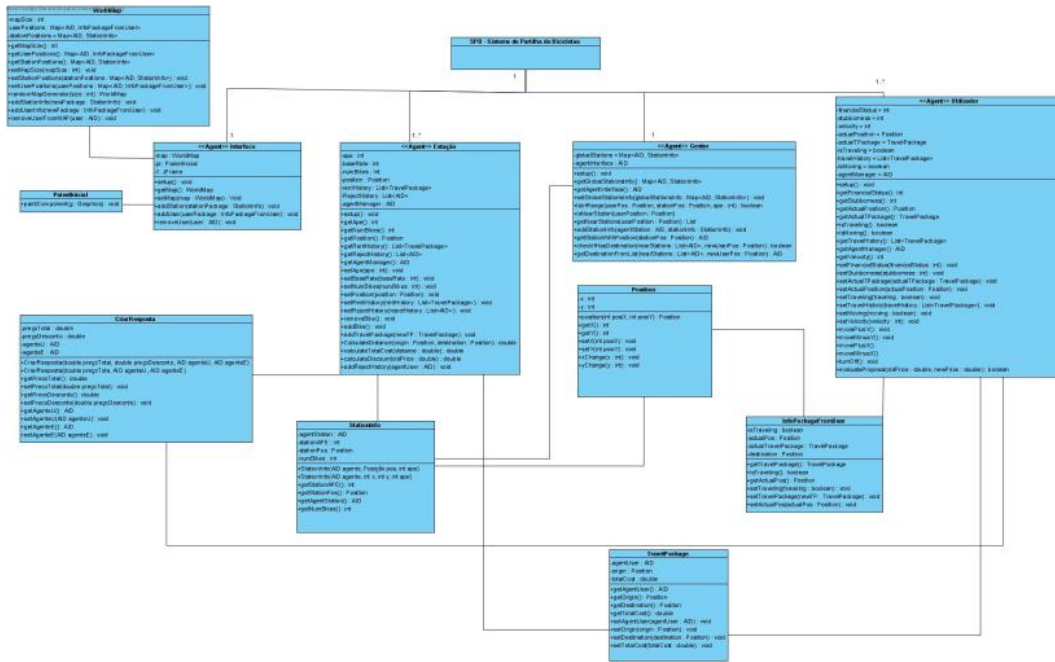


Figura 5: Diagrama de Classes

### 5.2 Inicialização

No ficheiro *Main Container* é iniciado o sistema e a interface JADE é iniciada. Através do ficheiro *DFFFunctions* os agentes são registados e controlados. No ficheiro *Main* é onde são criados os elementos do sistema. Primeiro são criadas posições aleatórias para posteriormente gerar-se as estações e os utilizadores consoante um número predefinido que permite ser alterado.

### 5.3 Agentes e respetivas funções

Cada Agente possui o seu ficheiro respetivo para melhor divisão e organização.

### 5.3.1 Agente Interface

O Agente Interface é responsável pela criação da interface gráfica. No mapa criado, pinta as posições de cada estação e utilizador, atualiza a posição dos utilizadores consoante informação que recebe através do comportamento associado. Distingue quem está a viajar e quem se encontra parado.

As estações têm cores diferentes consoante o número de bicicletas nelas presentes. Uma estação terá coloração preta caso não possua nenhuma bicicleta, coloração vermelha caso tenha entre 1 a 4 bicicletas, coloração verde caso tenha mais do que 5 bicicletas e coloração amarela caso tenha mais do que 20 bicicletas. O Utilizador terá cor rosa quando não se encontra a viajar e cor azul quando se encontra em viagem.

### 5.3.2 Agente Gestor

O Agente Gestor é responsável pelos cálculos de distâncias, isto é, verificar se um utilizador está dentro de alguma APE de qualquer estação ou de alguma estação específica, fornecer todas as estações que estão ao alcance do utilizador e verificar se alguma das estações próximas é o destino do utilizador. Esta gestão é possível e realizada devido ao comportamento( "*Behaviour*") associado *ReceiveInfoM*. Aqui é feito um tratamento das mensagens enviadas pelos utilizadores e pelas estações.

- Caso o Agente seja um utilizador e o performative da mensagem seja do tipo "INFORM" é imprimido o AID do agente que recebeu a mensagem do utilizador e após uma recolha de informações, é possível verificar e informar se se encontra em viagem, caso não se encontre a viajar significa que o utilizador acabou de ser criado e as suas informações são enviadas para o agente Interface.
- Verifica também se o utilizador está ao alcance de alguma estação, para que seja imprimido se alguma das estações é o seu destino. Caso o utilizador não esteja no alcance de nenhuma APE de uma estação, o sistema imprime essa informação.
- Caso a mensagem seja enviada por uma estação e o performative seja do tipo "INFORM", é imprimido o AID do agente Gestor a informar que recebeu uma mensagem da dada estação. É realizada uma recolha de informações da estação e são enviadas para o agente Interface.

### 5.3.3 Agente Estação

O Agente Estação é responsável pelo cálculo de desconto de preço, cálculo de preço total e ainda por registar e guardar um histórico de aluguer de bicicletas. Num dos seus comportamentos (*"Behaviours"* associados, é criado um pacote com informações da estação (APE e Posição), a mensagem com o performative "INFORM", é colocado o pacote na mensagem e posteriormente a mensagem é enviada. No outro comportamento, é feito um tratamento de mensagens recebidas.

- Caso seja uma mensagem de um Agente Gestor a enviar uma mensagem com performative do tipo "INFORM", a estação recebe um pacote do utilizador que entrou na sua APE. É feita uma proposta ao utilizador para que o utilizador acabe o percurso nessa estação. Caso não seja a sua estação de destino, é aplicado um desconto ao preço total.
- Caso o Utilizador envie uma mensagem com o performative do tipo "PROPOSE", ou seja, é feito um pedido para uma viagem, se a estação tiver bicicletas disponíveis, o pedido é aceite caso não tenha bicicletas é enviada uma mensagem "REJECT\_PROPOSAL" para informar o Utilizador que não é possível iniciar viagem.
- Caso o Utilizador envie uma mensagem "ACCEPT\_PROPOSAL", ou seja, aceitou deixar a bicicleta numa dada estação, é adicionada uma bicicleta ao número de bicicletas disponíveis nessa estação e esta informação é enviada para o Agente Gestor.
- Por último, se o Utilizador rejeitar a proposta, este é adicionado a um histórico para que não sejam enviadas mais propostas.

### 5.3.4 Agente Utilizador

Para o Agente Utilizador foram criadas 2 variáveis para criar uma simulação mais perto da realidade, o grau de riqueza e a sua teimosia que variam entre valor 1 e 10. Quanto maior o valor do grau de riqueza e teimosia, existe uma menor probabilidade de os descontos serem aceites. O Utilizador recebe mensagens do Gestor ou de uma Estação. Na sua criação, é recebida uma mensagem do Gestor a indicar que o Utilizador foi criado para que posteriormente seja feita a requisição de uma bicicleta.



- Quando é recebida uma mensagem com performative do tipo "PROPOSE" de uma Estação, o Utilizador pára de se mover enquanto não responder à proposta. Se a proposta for proveniente da estação de destino, a proposta é aceite caso não seja, a proposta é avaliada. Se o utilizador rejeitar a proposta, a estação é informada e este retoma o seu movimento.
- Quando é recebida uma mensagem com o performative do tipo "ACCEPT\_PROPOSAL" de uma estação, o utilizador é informado que pode começar a sua viagem.
- Caso a estação envie uma mensagem com performative do tipo "REJECT\_PROPOSAL" significa que o utilizador não pode requisitar nenhuma bicicleta e, assim como acontece quando o utilizador termina uma viagem, este é desconectado.

## 6 Interações

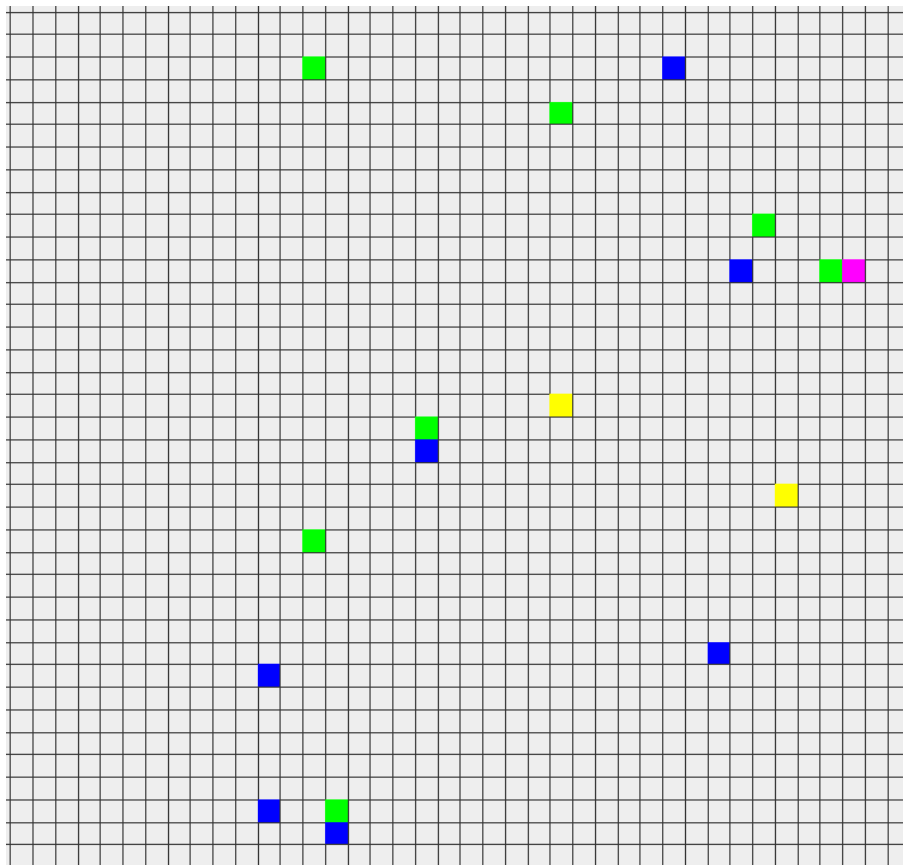


Figura 6: Basic World Map

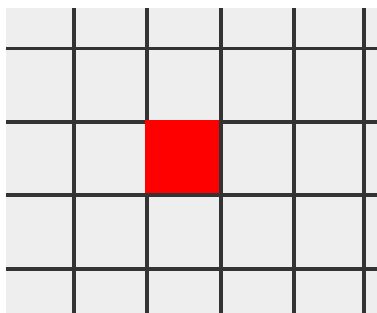


Figura 7: Station quase vazia



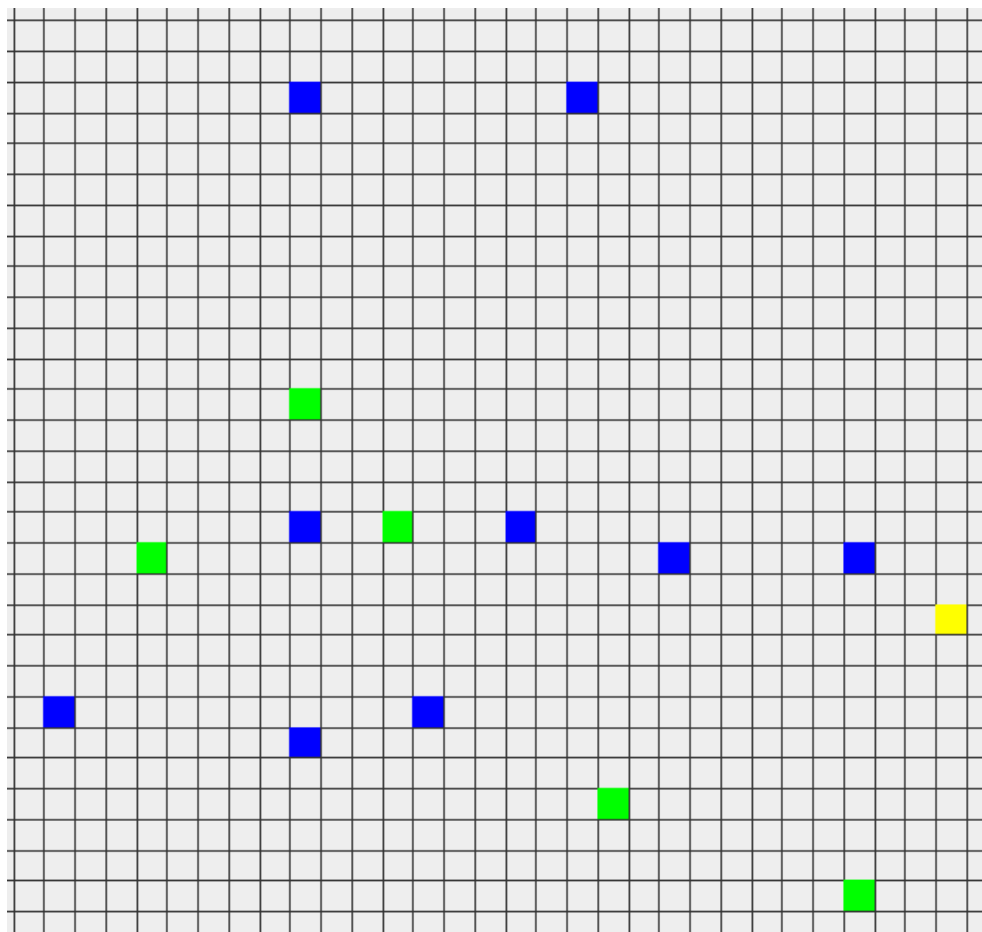


Figura 12: Basic World Map 2

## 7 Conclusão

Tendo em conta os pontos anteriores, a nossa arquitetura considera 4 agentes. Cada ação, consoante a natureza da mesma, despoletará uma reação dos agentes.

A nossa arquitetura considera o processo End-to-End da requisição de uma bicicleta.

De forma a evitar o desequilíbrio no número de bicicletas presentes em cada estação, é considerada a posição do utilizador e são propostas alterações no local de devolução da bicicleta conforme a lotação das estações. Num trabalho futuro e posterior a este, visando o melhoramento do mesmo, poderíamos implementar uma lista de espera de requisições caso uma estação não contivesse bicicletas disponíveis. Esta lista de espera poderia calcular o tempo de espera estimado, consoante o destino de outras viagens em progresso. Um outro melhoramento seria conectar o sistema a uma base de dados e registar um histórico de viagens, stock de bicicletas e das suas características (caso se implementassem bicicletas com características diferentes e preços variáveis). Com o histórico de viagens armazenado, seria mais viável e facilitado fornecer descontos e incentivos a utilizadores que beneficiaram o sistema, isto é, aceitaram propostas de outras estações diferentes do destino inicial permitindo o equilíbrio do sistema.