



UNIVERSIDADE DO MINHO
ESCOLA DE ENGENHARIA

Agentes Inteligentes

TRABALHO PRÁTICO

Ana Marta Santos Ribeiro A82474
Jéssica Andreia Fernandes Lemos A82061
Tiago Dias de Sousa A81922

MIEI - 4º Ano - Sistemas Inteligentes

Braga, 8 de Dezembro de 2019

Conteúdo

Conteúdo	1
1 Introdução	3
2 Estado de Arte	5
3 Arquitetura	6
4 Protocolo de Comunicação	8
4.1 Comunicação entre agente secundário e central	8
4.2 Comunicação entre o agente central e os agentes participativos	8
4.2.1 Coordenação dos agentes	8
4.2.2 Combate ao incêndio	10
5 Diagramas UML	11
5.1 Diagramas de Atividade	11
5.1.1 Gerar incêndio	11
5.1.2 Coordenação de agentes	11
5.1.3 Combater incêndio	12
5.2 Diagrama de estados	14
6 Alterações aos Diagramas UML	15
6.1 Diagrama de Classes	15
6.2 Diagramas de Sequência	16
6.2.1 Coordenação de agentes	16
6.3 Diagramas de Atividade	17
6.3.1 Coordenação de agentes	17
6.3.2 Combate ao Incêndio	18
6.4 Diagrama de Estados	19
7 Implementação	20
7.1 Geração do Mapa	20
7.2 Armazenamento da Informação em Jess	21
7.3 Geração dos incêndios	23

7.4	Coordenação dos agentes	23
7.5	Combater incêndios	25
7.6	Interface	27
8	Demonstração	29
8.1	Mapa 2D	29
8.2	Estatísticas	30
9	Conclusões e Trabalho Futuro	31
	Bibliografia	32

1 Introdução

Este relatório surge no âmbito do trabalho prático da unidade curricular de Agentes Inteligentes que se encontra inserida no perfil de Sistemas Inteligentes. Neste trabalho pretende-se que seja elaborado um sistema multiagente recorrendo ao ambiente de desenvolvimento JADE (Java Agent Development Framework) bem como JESS no desenvolvimento dos agentes.

O problema apresentado consiste em desenvolver um sistema de combate a incêndios florestais. Neste sistema existe um agente central que é responsável por distribuir e coordenar todos os veículos de forma a extinguir os incêndios existentes do modo mais eficiente possível. Desta forma, podemos identificar quatro tipos de agentes, nomeadamente o central, o secundário, o participativo e o interface. O agente secundário é responsável por gerar incêndios em posições aleatórias, o central representa o quartel e os participativos correspondem a veículos autônomos e inteligentes que combatem o incêndio (drones, aeronaves e camiões). O agente interface é aquele com que o utilizador interage.

Estrutura do Relatório

Este relatório contém 4 secções correspondentes à primeira fase da elaboração deste projeto, que são uma contextualização do estado dos sistemas multiagente de resposta a catástrofes naturais existentes no mercado atual, com a apresentação de algumas das soluções. De seguida é explicada a arquitetura do sistema que vamos desenvolver, com a apresentação do diagrama de classes e descrição da função de todos os agentes que nele atuam. Depois é descrita a comunicação entre os principais agentes ativos no sistema com recurso a diagramas de sequência. Por fim, é feita a modelação, através de diagramas de atividade e de estados, dos comportamentos e decisões que os agentes principais tomam perante as situações com que se deparam no decorrer do funcionamento do sistema. As situações descritas são a coordenação de agentes por parte do agente central, combate de incêndios por parte dos agentes participativos, e gerar fogos por parte do agente secundário.

De seguida, surgem mais 4 novas secções referentes à fase final do projeto. Começamos com a apresentação das alterações necessárias realizar em diagramas apresentados anteriormente devido à implementação. Tendo em conta que um dos objetivos seria o desenvolvimento dos agentes com JESS, apresentamos como implementamos

o mesmo no nosso projeto. Passando depois para a implementação em que indicamos como desenvolvemos o projeto, de modo a demonstrarmos por fim o resultado da fase anteriormente referida.

Este relatório termina com a apresentação das nossas conclusões, na qual fazemos uma pequena reflexão do trabalho desenvolvido bem como sobre o trabalho futuro.

2 Estado de Arte

As catástrofes naturais fazem parte da dimensão trágica da história da civilização humana. Os desastres naturais bem como aqueles que são causados pelo homem criam desafios sem precedentes às operações de resposta, socorro e recuperação. Segundo a Organização das Nações Unidas, atingimos uma média de um desastre natural por semana, um ritmo alarmante e preocupante. Assim, temos apostado cada vez mais em sistemas que possam auxiliar na prevenção e diminuição dos danos causados. A abordagem dos sistemas multiagentes já provou ser uma solução eficaz para resolver os problemas de coordenação de situações de desastre devido à sua estrutura distribuída, o uso de princípios de agentes móveis inteligentes e ajuste natural para modelar a formação de equipas.

O *DrillSim* é uma framework de simulação e modelação multiagente para exercícios de resposta a emergências. Nesta, os agentes adquirem consciência do mundo ao seu redor, transformando os dados adquiridos em informações e tomando decisões com base nessas informações usando redes neurais recorrentes.

O *ALADDIN* é um sistema que visa modelar, projetar e construir sistemas descentralizados que podem reunir informações de várias fontes, de modo a tomar ações informadas. Para alcançar este objetivo, este considera diferentes aspetos, como fusão de dados, tomada de decisão e arquitetura de sistemas.

O *RoboCup Rescue* é um sistema de simulação e modelação multiagente. O principal objetivo do projeto é permitir que as equipas de resgate cooperem, apesar das limitações de deteção e comunicação. Este é baseado na pesquisa urbana e na simulação de resgate.

O *DrillSim* foi projetado para estudar métricas de tecnologias de informação em resposta a desastres, e o *ALADDIN* para estudar diferentes arquiteturas de agentes em operações de resposta. Enquanto isso, o *RoboCup Rescue* concentra-se nas operações de busca e salvamento urbano.

Estes são apenas alguns exemplos das várias propostas de sistemas multiagentes que seguem diferentes estratégias. O cenário escolhido para este projeto é um simulador de combate a incêndios florestais.

3 Arquitetura

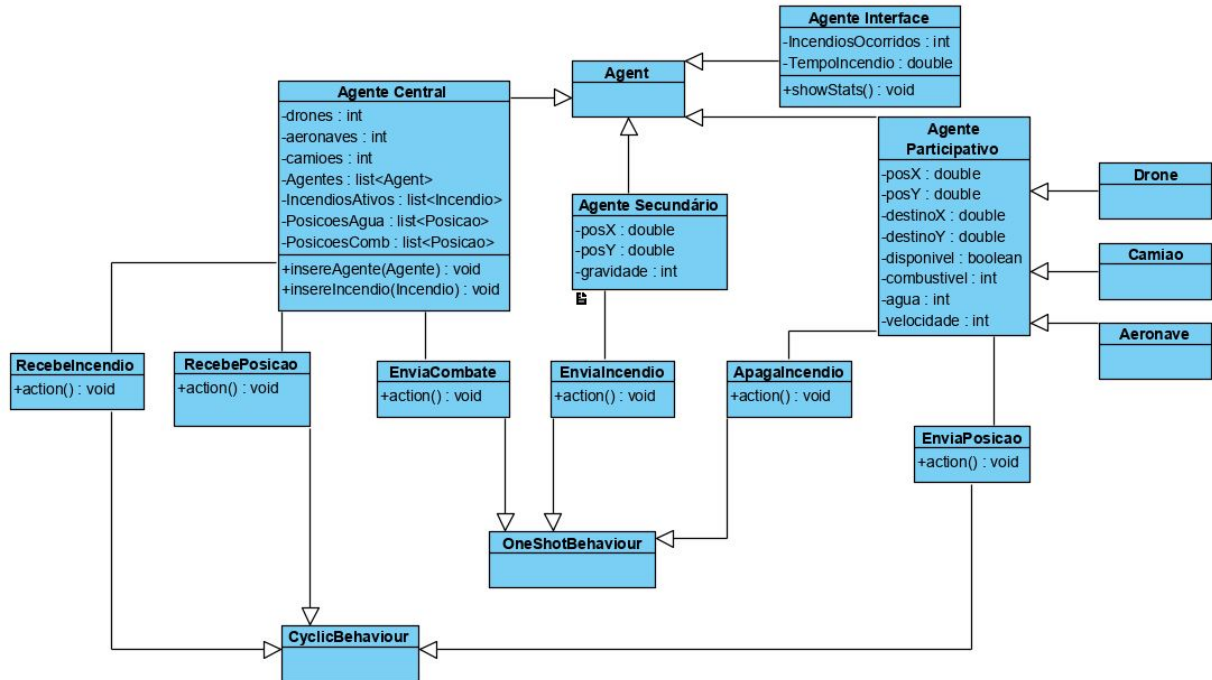


Figura 1. Diagrama de Classes que descreve a arquitetura do sistema

A estrutura do sistema, representada pelo diagrama de classes, é feita através de quatro tipos de agentes, que correspondem a quatro classes que são especificações da classe *Agent*, que é disponibilizada pelo JADE. Estas quatro classes são:

- O **Agente Central**, que representa o quartel que vai coordenar os agentes participativos para o combate de incêndios florestais. Para facilitar a coordenação, este agente central tem como atributos as informações dos agentes participativos, a quantidade de agentes de combate disponíveis de cada tipo, e informações sobre a posição e gravidade dos incêndios ativos.

A recepção das informações sobre os agentes de combate e incendiários vai ser feito pelas classes *RecebeIncendio* e *RecebePosicao*, em que a sua ação vai ser executada de forma cíclica pelo agente central. Para enviar um agente para combate tem a classe *EnviaCombate*.

- O **Agente Secundário** tem os atributos da posição do incêndio e a sua gravidade, que posteriormente serão enviados através da classe *EnviaIncendio* para o *AgenteCentral*.

- O **Agente Participativo**, que representa os agentes que combatem o incêndio contém como atributos a posição onde está e a posição para onde se deverá dirigir, ou seja, a localização do incêndio que irá combater. Para além disso, o seu estado de disponibilidade para que o agente central perceba quais os agentes que pode enviar para o combate de um incêndio, e ainda a quantidade de água e combustível que contém no momento, de modo a facilitar o processo de decisão por parte do quartel.

A posição onde o agente se encontra vai ser enviada de forma repetitiva num certo intervalo de tempo para o agente central através da classe *EnviaPosicao* de forma a acompanhar o seu movimento.

Este agente participativo pode ser representado em 3 tipos de agentes combatentes, podendo ser um drone, camião ou aeronave.

- Ao **Agente Interface** cabe mostrar ao utilizador algumas estatísticas sobre o funcionamento do sistema, contém os atributos do número de incêndios ocorridos e do tempo de incêndios ativos no sistema que são duas estatísticas que achamos relevantes no contexto do sistema de combate a incêndios. Estas estatísticas serão consultadas pelo método *showStats* que quando chamado as mostrará ao utilizador.

4 Protocolo de Comunicação

Para a implementação deste simulador de combate a incêndios florestais torna-se necessário termos em atenção a comunicação entre os vários agentes intervenientes.

Dado que é possível identificar a comunicação que deverá ser realizada entre os três diferentes tipos de agentes - participativos, secundário e central - optamos inicialmente por fazer uma divisão da interação entre estes, que serão demonstradas posteriormente de uma forma intuitiva através de Diagramas de Sequência. Surgem assim três diagramas, em que cada um representa uma fase do processo do combate a incêndios - geração, coordenação dos meios e combate.

Em cada um dos diagramas encontram-se ilustradas as mensagens trocadas entre os diferentes agentes, e para tal indicamos o tipo de performatives, ou seja, de mensagem que é enviada.

4.1 Comunicação entre agente secundário e central

Após o agente secundário gerar o incêndio este envia as informações do incêndio, ou seja, as coordenadas e a gravidade do mesmo, através de performatives *inform*. Sendo assim possível o agente central registar o incêndio.

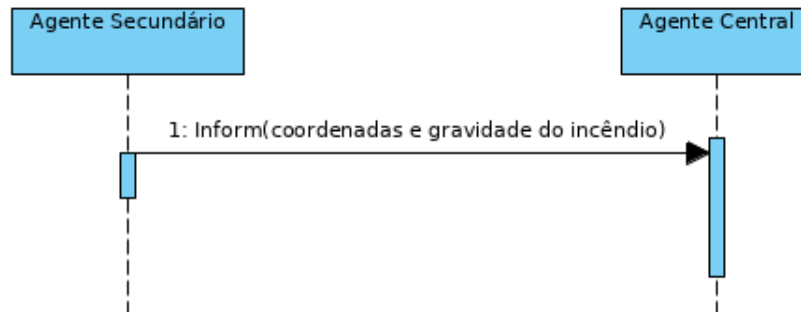


Figura 2. Diagrama de Sequência que descreve a geração do incêndio

4.2 Comunicação entre o agente central e os agentes participativos

4.2.1 Coordenação dos agentes

De forma a realizar uma correta coordenação e distribuição dos agentes participativos, o agente central necessita de ter conhecimento das características e da dispo-

nibilidade de cada um destes agentes. Deste modo é fundamental que haja uma boa comunicação entre ambos.

O agente participativo tem de informar constantemente o agente central do seu estado, isto é, a sua localização, o nível de combustível e o nível de água, para tal recorre a performatives *inform*. Caso haja alguma alteração na sua disponibilidade, ou seja, estar ocupado e passar a estar livre ou o contrário este deverá informar o agente central de tal modificação. Na eventualidade do agente participativo ser selecionado para combater um incêndio o agente central deverá enviar-lhe as coordenadas com a localização do mesmo. Sempre que um agente participativo extingue um incêndio informa o agente central.

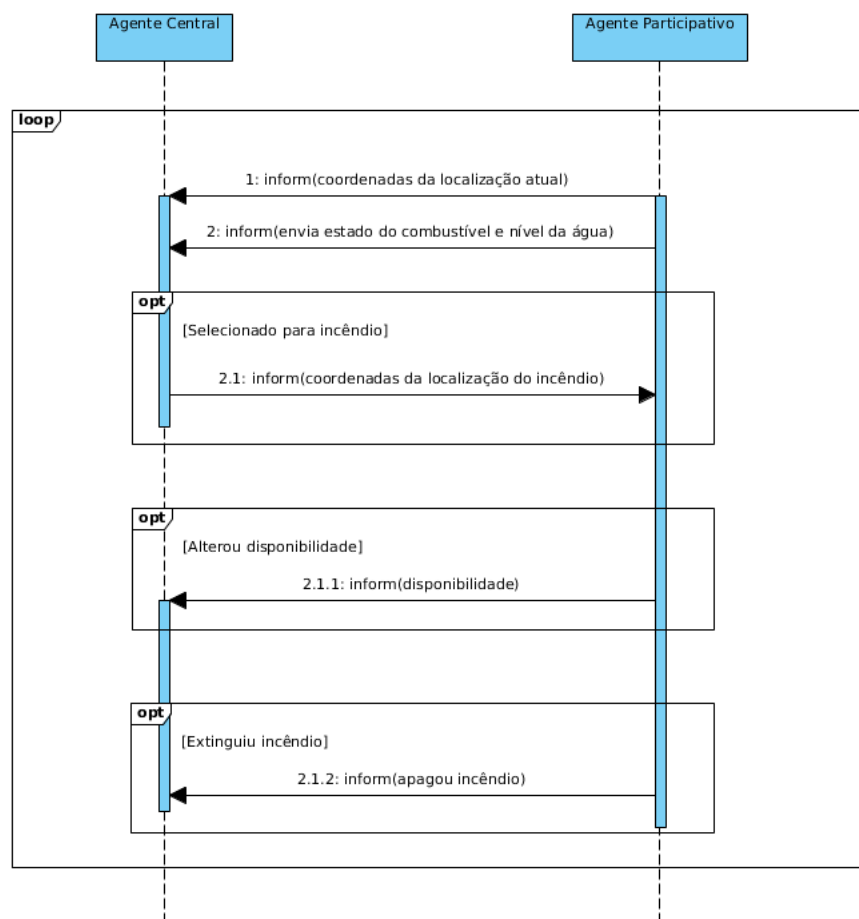


Figura 3. Diagrama de Sequência que descreve a comunicação entre o agente central e o participativo na coordenação dos agentes

4.2.2 Combate ao incêndio

O agente participativo recebe as coordenadas do incêndio, a partir daí envia constantemente a sua localização, nível de combustível e de água ao agente central, período no qual se está a dirigir para o incêndio. Quando o incêndio é apagado, o agente participativo informa o agente central do sucedido. Após o incêndio ser apagado, caso o agente participativo não precise de abastecer, informa o agente central de que está disponível para receber novas ordens. Caso contrário, só se informa disponível quando se verificarem as duas condições de água superior a 0 e combustível igual ou superior a metade do nível máximo de capacidade.

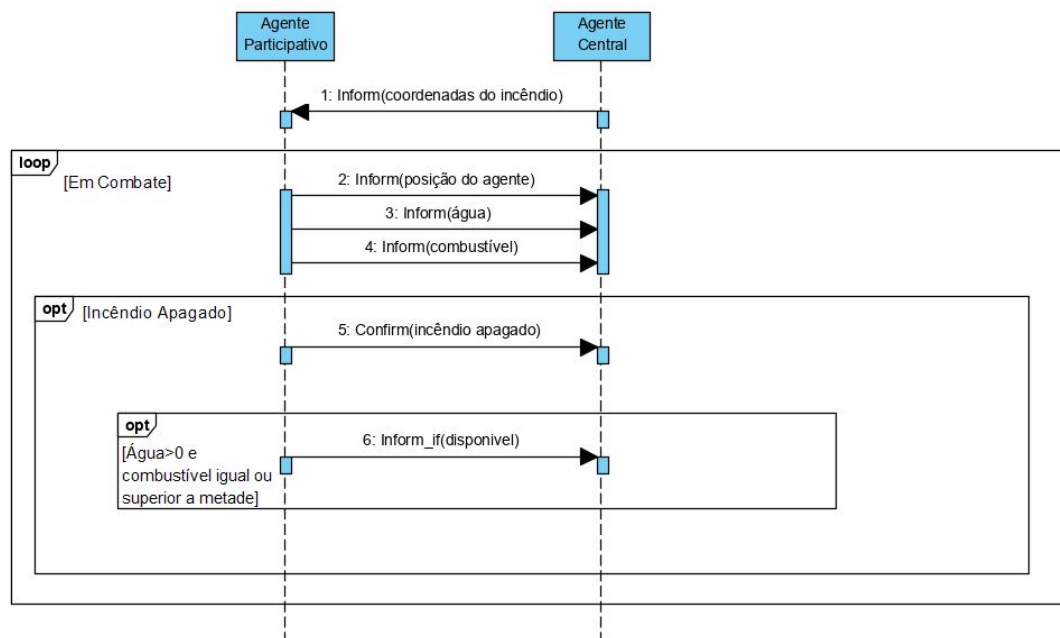


Figura 4. Diagrama de Sequência que descreve a comunicação entre o agente central e o participativo durante o combate ao incêndio

5 Diagramas UML

5.1 Diagramas de Atividade

Estes diagramas têm como objetivo descrever com rigor todo o processo interno dos agentes. Assim, tornam-se mais perceptíveis todas as decisões tomadas por cada um que provêm da troca de mensagens existentes entre os diferentes agentes.

5.1.1 Gerar incêndio

Inicialmente, o agente secundário seleciona uma posição aleatoriamente no mapa onde irá gerar o incêndio. Posteriormente, este irá informar o agente central sobre o incêndio, enviando as informações sobre o mesmo, tal como as coordenadas e a gravidade. Desta forma, torna-se possível o agente central registrar o incêndio.

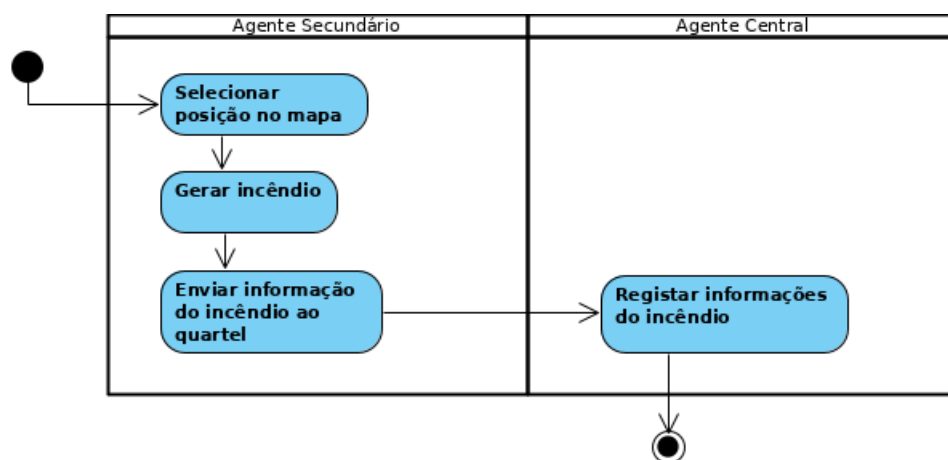


Figura 5. Diagrama de Atividade que descreve a geração do incêndio

5.1.2 Coordenação de agentes

O objetivo deste diagrama consiste em representar o processo de decisão do veículo a enviar para um incêndio. Como tal o processo inicia-se com o agente central a registrar as informações do incêndio, nomeadamente a localização e a gravidade do mesmo. De seguida, este verifica que agentes participativos tem disponíveis para combater o incêndio. De modo a selecionar a melhor opção, este consulta a posição dos agentes, bem como o seu nível de combustível e água. Tendo em conta a capacidade de cada um, a distância a que se encontram do incêndio e a velocidade a que se deslocam, é calculada

uma estimativa de quanto tempo demoram a chegar ao local do incêndio. Posteriormente é escolhido o veículo mais eficaz. Na eventualidade de ser selecionado um drone e o incêndio não ser considerado grave, ser perto ou não se encontrar ativo há muito tempo, vamos verificar o número de drones disponíveis no momento. Caso haja vários, enviamos o agente para o incêndio, caso contrário selecionamos outro agente, de forma a mantê-lo de reserva para um incêndio em que a sua utilização seja imprescindível. Uma vez tomada a decisão, o agente central coloca o agente participativo selecionado com indisponível e depois envia-lhe as coordenadas da localização do incêndio. Quando recebe a localização, o agente participativo regista as coordenadas.

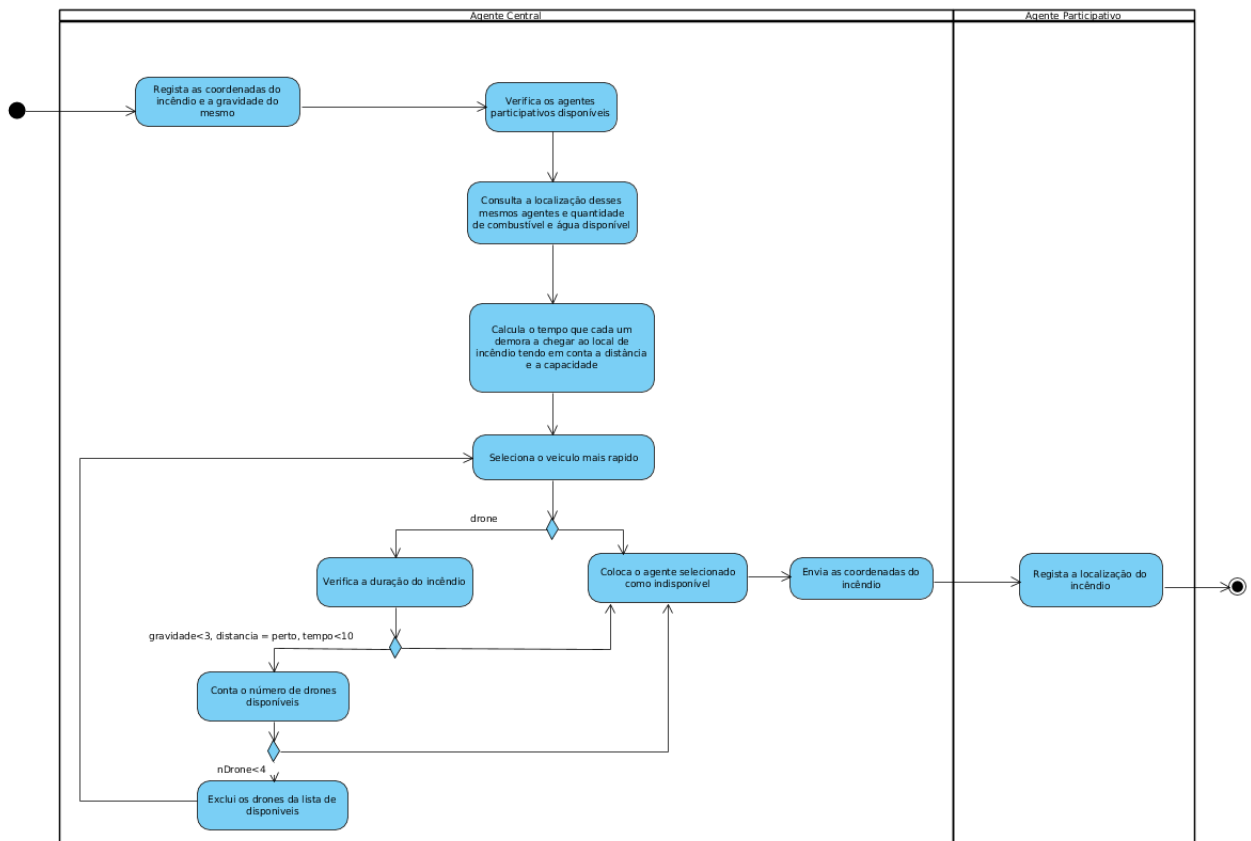


Figura 6. Diagrama de Atividade que descreve o modo como o agente central seleciona o agente participativo que deve combater o incêndio

5.1.3 Combater incêndio

O objetivo deste diagrama consiste em representar todo o processo de decisão do agente participativo no combate a um incêndio. O processo inicia-se com o agente

participativo a dirigir-se para a posição das coordenadas do incêndio que lhe foram enviadas pelo agente central (processo representado na secção anterior). Após chegar ao incêndio, o agente participativo apaga-o, o que leva a que o agente central elimine o incêndio da lista de incêndios ativos. Depois disto, passamos à verificação dos recursos do agente participativo, caso o mesmo tenha o nível da água superior a 0 e do combustível igual ou superior a metade da capacidade, torna-se disponível para receber novas ordens do agente central. Caso não tenha mais de metade do nível de algum desses recursos dirige-se para o posto de abastecimento do recurso que necessite, de modo a reabastecer novamente até à capacidade máxima, ficando após isto, disponível para ordens do agente central. Caso necessite dos dois recursos em simultâneo dirige-se inicialmente para o posto de combustível, e após abastecer, dirige-se para o posto de água onde também abastecerá, e só depois fica disponível para receber ordens do agente central.

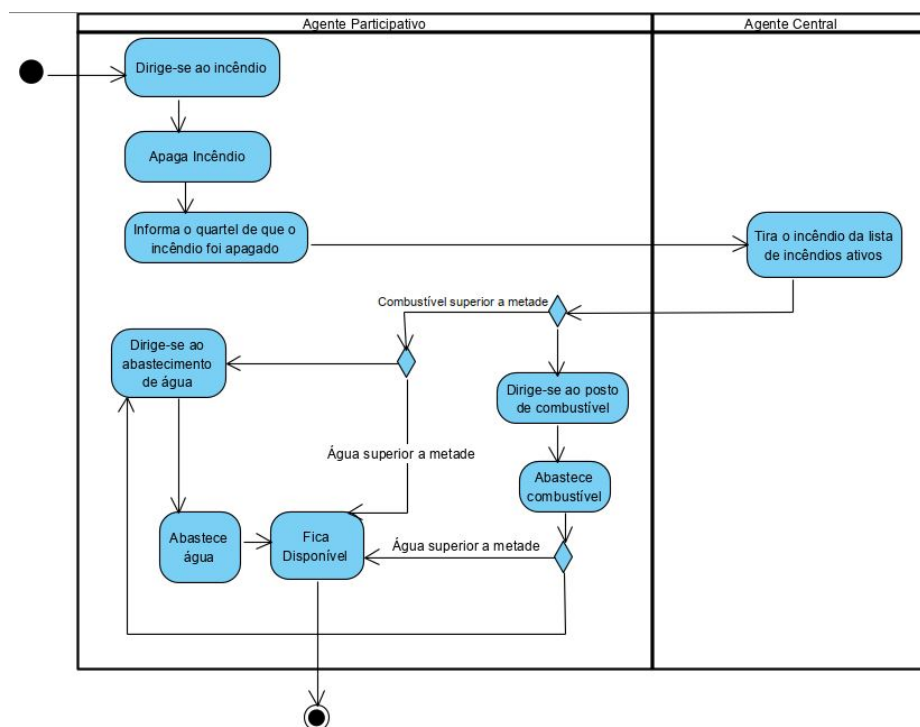


Figura 7. Diagrama de Atividade que descreve o modo de ação do agente participativo durante o combate ao incêndio

5.2 Diagrama de estados

A elaboração deste diagrama teve como intuito representar as diferentes transições e os possíveis estados de comunicação dos agentes participativos ao longo de todo o processo de combate a incêndios florestais. Neste processo estes agentes estarão sempre a partilhar as suas coordenadas com o agente central de modo a este estar a par das posições de todos os agentes, pelo que ao longo do seu ciclo de vida no sistema o agente participativo apresenta o estado *Partilhar Posição com AC* e paralelamente o estado *Partilhar nível de combustível e água* de modo ao agente central ter as informações necessárias para a tomada de decisões. Inicialmente, os agentes participativos estão no estado *Disponível* até que exista um incêndio e seja enviado um dos agentes participativos para combater o incêndio passando para o estado *Em combate*. Assim que o incêndio seja extinto, o agente, caso não tenha combustível ou água, irá passar para o estado *Em abastecimento*. Pelo que, de seguida o agente pode informar o quartel da sua disponibilidade exibindo assim dois estados paralelamente, o *Disponível* e o *Comunicar disponibilidade ao AC* de modo a indicar ao agente central que pode já pode ajudar no combate de outro incêndio.

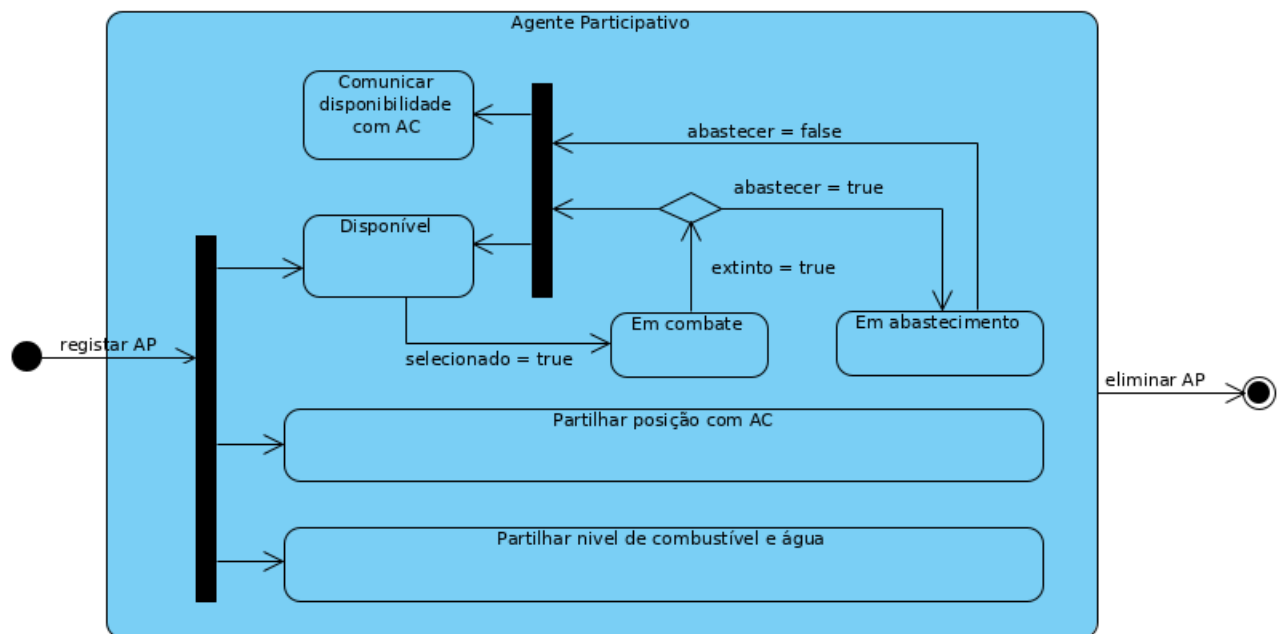


Figura 8. Diagrama de Estados dos Agentes Participativos

6 Alterações aos Diagramas UML

Durante a segunda fase do trabalho prático verificamos que alguns dos diagramas UML elaborados anteriormente tinha alguns pormenores que não se adequavam à solução implementada. Assim sendo, nesta secção iremos apresentar os que tiveram de sofrer pequenas alterações, bem como o motivo para tal, assim como se pode observar de seguida.

6.1 Diagrama de Classes

Neste diagrama foram alterados maioritariamente os tipos de behaviours, que tínhamos planeado de forma imprecisa na fase anterior. Foram também adicionados os behaviours de movimentação, e de receber mensagens por parte da interface.

A movimentação foi adicionada como forma de *TickerBehaviour*, de forma a que os agentes se movimentem um "passo" a cada intervalo de tempo definido.

O behaviour *GerarIncendio* passou a ser *TickerBehaviour*, para poder executar de 5 em 5 segundos a criação de um incêndio no mapa.

O behaviour *EnviaInfo* faz o envio da informação inicial dos agentes para o agente central. O behaviour *ReceberMensagens* foi criado no agente Interface como *CyclicBehaviour*, e foi adicionado com o objetivo de poder receber as informações dos agentes do sistema de modo a poder representar graficamente o seu estado e algumas estatísticas sobre o funcionamento.

Foi também adicionada a classe *Work*, do tipo *JPanel* que faz o desenho do mapa e dos agentes na Interface Gráfica. Neste agente temos também acesso ao mapa gerado inicialmente de modo a acedermos aos locais de abastecimento.

De modo ao agente central ter uma lista de todos os incêndios, criamos a classe *Incendio* que contém as informações do mesmo, como o seu estado, o tempo que se encontra ativo e a sua localização.

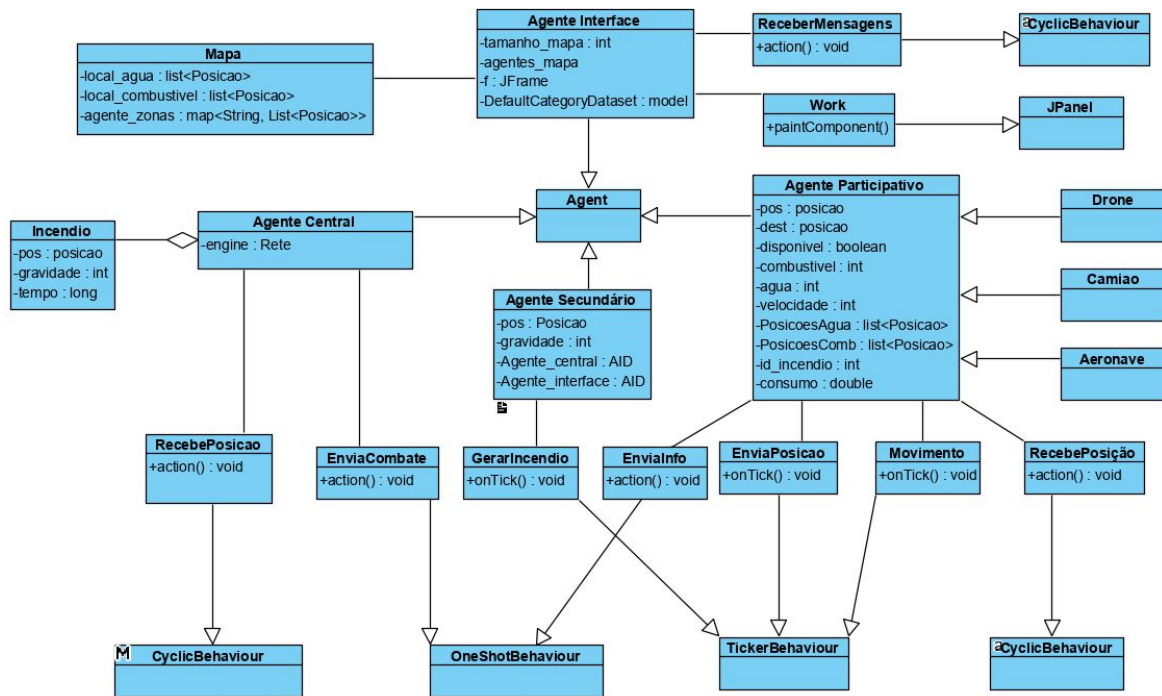


Figura 9. Diagrama de Classes

6.2 Diagramas de Sequência

6.2.1 Coordenação de agentes

Foi necessário alterar este diagrama, uma vez que na primeira fase tínhamos pensado em colocar os agentes participativos a enviar constantemente a sua posição e o nível de combustível e água. No entanto, verificamos que apenas fazia sentido enquanto este estivesse em movimento, ou seja, quando não está disponível. Assim sendo, os agentes participativos enviam as suas informações no início e depois informam o agente central sempre que há alteração.

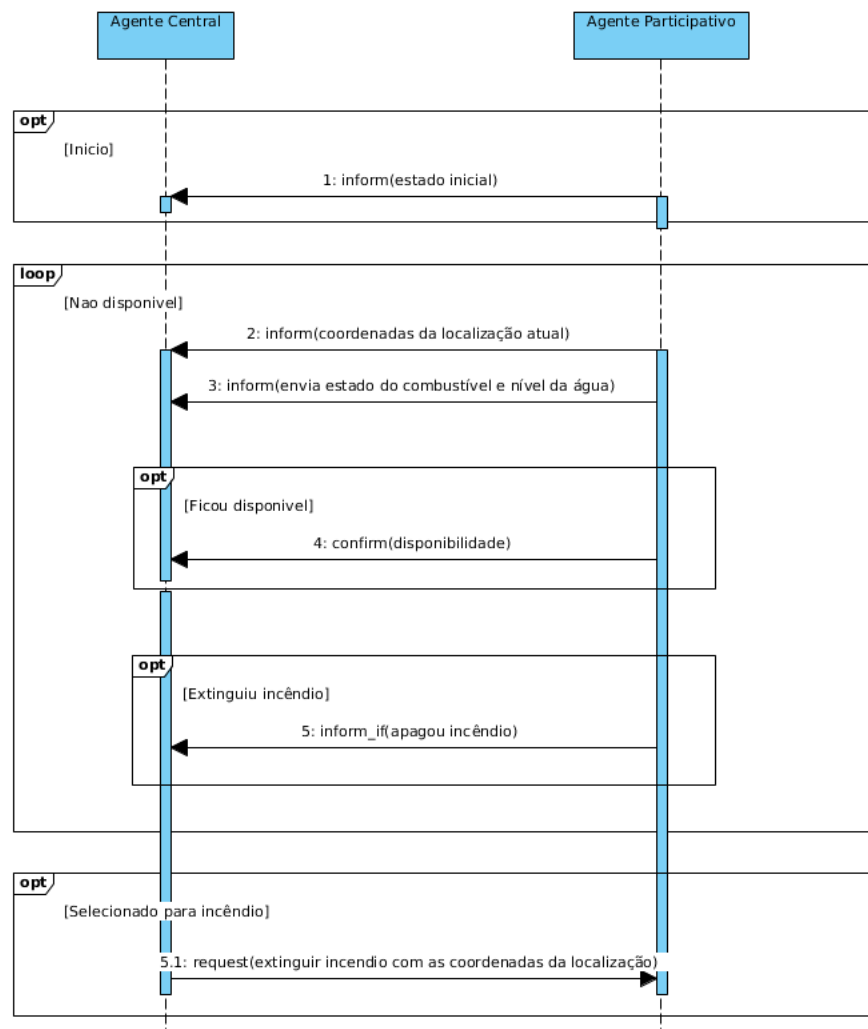


Figura 10. Diagrama de Sequência Coordenação de agentes

6.3 Diagramas de Atividade

6.3.1 Coordenação de agentes

Este diagrama pretende demonstrar o modo como que se realiza o processo de decisão do veículo a enviar para o incêndio. Neste, apenas alteramos o critério final de escolha, dado que na primeira fase só iríamos mandar o drone caso fosse grave ou houvessem vários disponíveis e agora nesta optamos por definir um conjunto de agentes de reserva que apenas vão quando o incêndio é de gravidade máxima, tal como se pode observar de seguida.

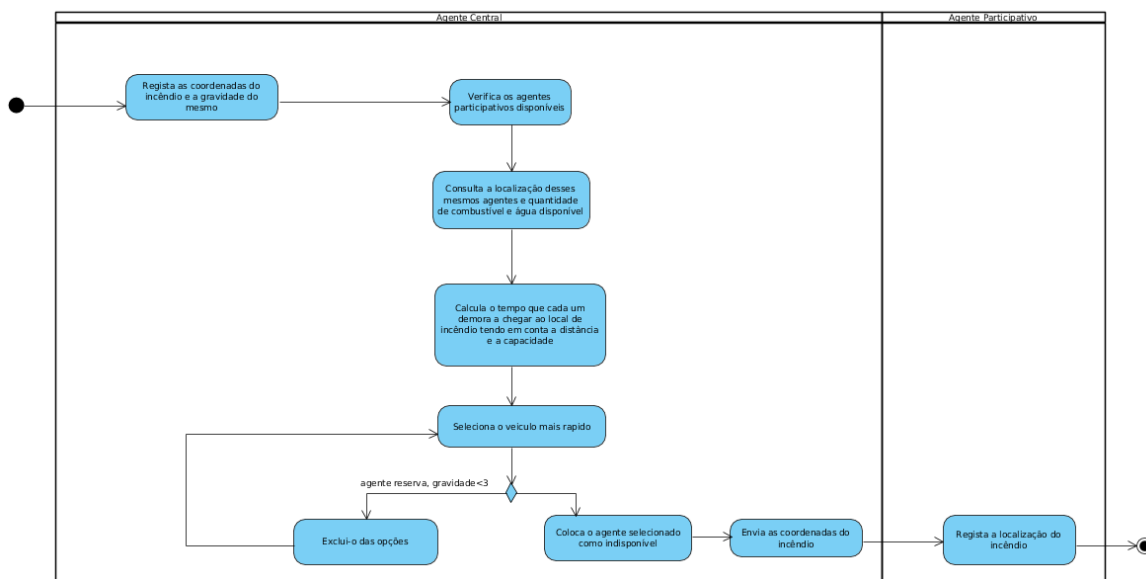


Figura 11. Diagrama de Coordenação de agentes

6.3.2 Combate ao Incêndio

Neste diagrama foi alterado o abastecimento de água, sendo que optamos por apenas enviar os agentes para o abastecimento quando ficarem sem água, esta medida permite poupar muito combustível que era gasto nas deslocações para postos de água.

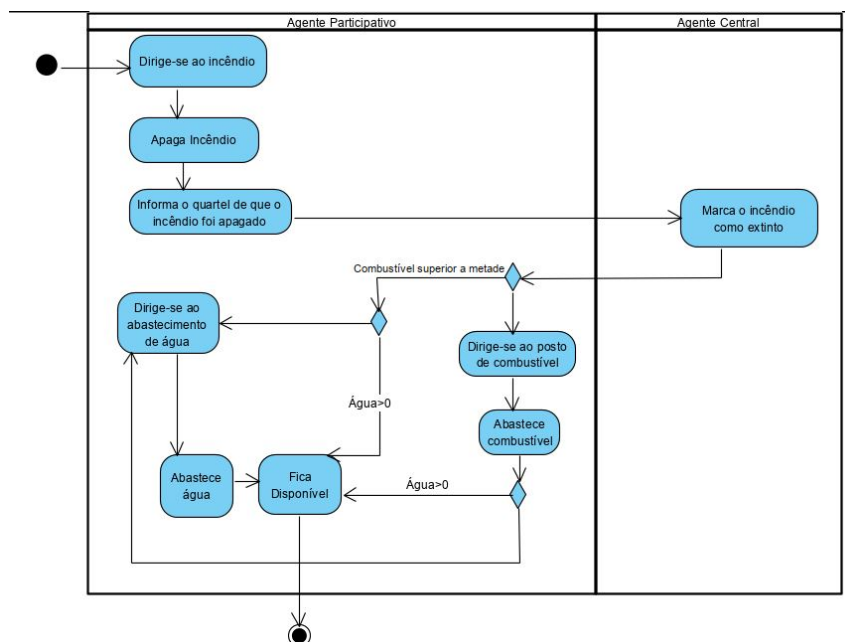


Figura 12. Diagrama de Combate

6.4 Diagrama de Estados

Devido a ser ter optado na implementação por inicialmente mandar as informações necessárias num *OneShotBehaviour* e depois mandar-se a informação que ia alterando do agente enquanto este estava indisponível como água, combustível e a sua posição num *TickerBehaviour* a parte inicial deste diagrama sofreu uma pequena alteração. Sendo que assim manda-se no início estas informações do agente e depois quando o agente fica indisponível é enviado paralelamente as suas informações.

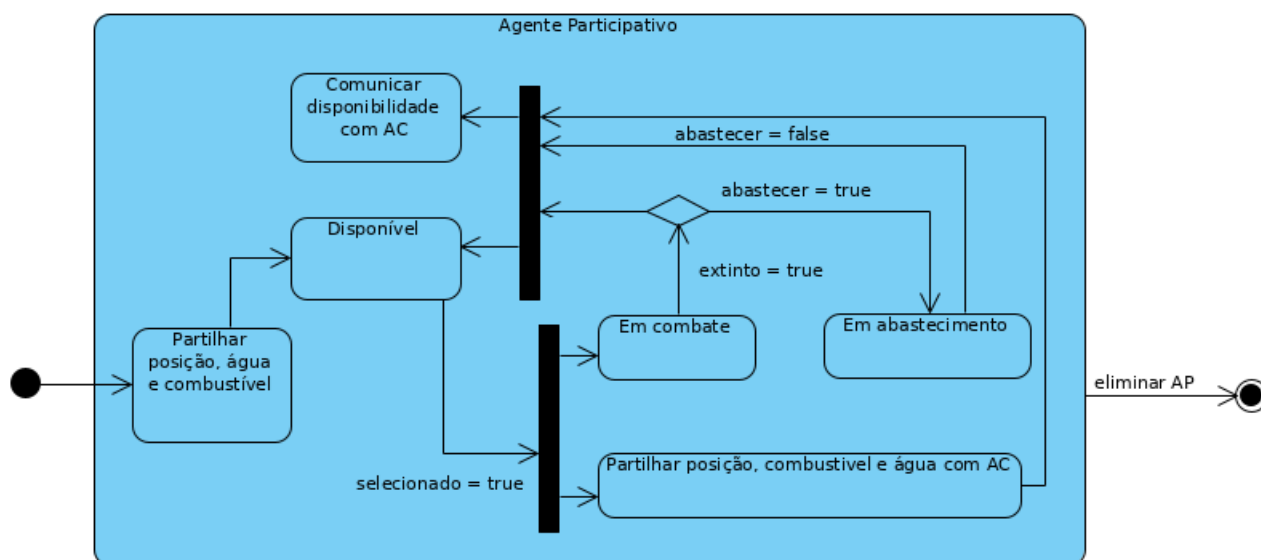


Figura 13. Diagrama de Estados dos Agentes Participativos

7 Implementação

7.1 Geração do Mapa

Inicialmente é gerado um mapa que contém todos os locais de abastecimento, quer de água quer de combustível. Para tal, decidimos dividir este mesmo mapa por zonas, de modo a gerar estes locais e posteriormente posicionar os agentes.

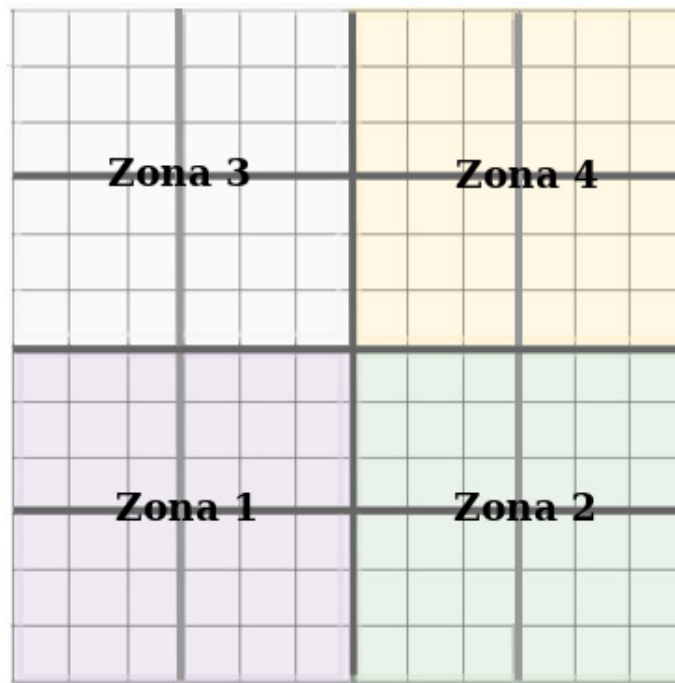


Figura 14. Divisão do mapa

A estratégia adoptada foi dividir o mapa em 4 regiões distintas, tal como podemos observar na imagem anterior, nas quais iremos posterior distribuir os vários agentes de cada tipo. Contudo para os locais de abastecimento decidimos dividir estas regiões em subregiões de modo a gerar os postos em zonas não muito afastadas uma vez que as dimensões do mapa pretendidas são bastante grandes. Assim, em cada subregião são gerados 2 postos de combustível e de água. Quanto à estratégia de distribuição dos agentes participativos optamos por gerar aleatoriamente dois drones e um camião por cada zona, as aeronaves sendo apenas duas foram distribuídas por duas zonas distintas. De modo a termos agentes prontos a combater em situações de incêndios graves optamos por colocar dois drones e um camião no centro do mapa, que estão prontos a atuar nestas emergências.

7.2 Armazenamento da Informação em Jess

Para o armazenamento de informação acerca dos Agentes Participativos foi definido um template em Jess.

```
(deftemplate combate
(slot x (type INTEGER))
(slot combustivel)
(slot y (type INTEGER))
(slot id (type INTEGER))
(slot agua)
(slot tipo)
(slot velocidade)
(slot consumo)
(slot disponivel))
```

Foi também criado um template agente, para quando são apenas enviadas as informações de combustível, água e posição, onde através duma rule, é inserida essa informação parcial no agente de combate correspondente que já se encontra na base de conhecimento.

```
(defrule updateCombate
?ag <- (agente)
?cb <- (combate {(id == ag.id)})
=>
(bind ?cb.x ?ag.x)
(bind ?cb.y ?ag.y)
(bind ?cb.agua ?ag.agua)
(bind ?cb.combustivel ?ag.combustivel)
(retract ?ag)
)
```

Para conseguirmos procurar agentes de combate na base de conhecimento definimos a seguinte query:

```
(defquery procuraAgenteCombate
(combate (x ?x)(y ?y)(id ?id)(agua ?a)(combustivel ?c)(disponivel ?d)(tipo ?t)
```

```
(velocidade ?v)(consumo ?co))  
)
```

Por fim, para utilizarmos estas rules criadas anteriormente no agente central inicializamos um engine no Agente central através de:

```
private Rete engine;  
this.engine= new Rete();  
try {  
    engine.batch("Apoio/ex2.clp");  
    engine.reset();  
} catch (JessException e) {  
    e.printStackTrace();  
}
```

Através disto, podemos inserir e retirar factos na base de conhecimento, como podemos ver no exemplo seguinte retirado do nosso Agente Central:

```
engine.executeCommand("(assert (agente (x "+c.getPos_x()+") (y "+c.getPos_y()+")  
(id "+local[1]+") (agua "+c.getAgua()+") (combustivel "+c.getCombustivel()+"))))");
```

Também podemos executar as queries de modo a obter as informações que definimos para pesquisa na base de conhecimento, como por exemplo:

```
QueryResult rs= engine.runQueryStar("procuraAgenteCombate",  
new ValueVector());
```

Por fim apresentamos de seguida uma demonstração dos factos do sistema durante a execução do programa.

```
engine.executeCommand("(facts)");
```

```

f-0 (MAIN::initial-fact)
f-1 (MAIN::combate (x 369) (combustivel 5.0) (y 294) (id 3) (agua 2) (tipo 1) (velocidade 4) (consumo 0.02) (disponivel true))
f-2 (MAIN::combate (x 233) (combustivel 5.0) (y 156) (id 5) (agua 2) (tipo 1) (velocidade 4) (consumo 0.02) (disponivel true))
f-3 (MAIN::combate (x 167) (combustivel 5.0) (y 233) (id 1) (agua 2) (tipo 1) (velocidade 4) (consumo 0.02) (disponivel true))
f-4 (MAIN::combate (x 379) (combustivel 5.0) (y 273) (id 6) (agua 2) (tipo 1) (velocidade 4) (consumo 0.02) (disponivel true))
f-5 (MAIN::combate (x 145) (combustivel 4.68) (y 250) (id 0) (agua 2) (tipo 1) (velocidade 4) (consumo 0.02) (disponivel false))
f-6 (MAIN::combate (x 33) (combustivel 5.0) (y 71) (id 4) (agua 2) (tipo 1) (velocidade 4) (consumo 0.02) (disponivel true))
f-7 (MAIN::combate (x 384) (combustivel 5.0) (y 374) (id 2) (agua 2) (tipo 1) (velocidade 4) (consumo 0.02) (disponivel true))
f-8 (MAIN::combate (x 272) (combustivel 4.64) (y 451) (id 7) (agua 2) (tipo 1) (velocidade 4) (consumo 0.02) (disponivel false))
f-9 (MAIN::combate (x 153) (combustivel 10.0) (y 95) (id 10) (agua 5) (tipo 2) (velocidade 2) (consumo 0.08) (disponivel true))
f-10 (MAIN::combate (x 389) (combustivel 20.0) (y 301) (id 13) (agua 10) (tipo 3) (velocidade 1) (consumo 0.05) (disponivel true))
f-11 (MAIN::combate (x 397) (combustivel 10.0) (y 310) (id 11) (agua 5) (tipo 2) (velocidade 2) (consumo 0.08) (disponivel true))
f-12 (MAIN::combate (x 250) (combustivel 5.0) (y 250) (id 8) (agua 2) (tipo 1) (velocidade 4) (consumo 0.02) (disponivel true))
f-13 (MAIN::combate (x 232) (combustivel 20.0) (y 110) (id 12) (agua 10) (tipo 3) (velocidade 1) (consumo 0.05) (disponivel true))
f-14 (MAIN::combate (x 250) (combustivel 5.0) (y 250) (id 9) (agua 2) (tipo 1) (velocidade 4) (consumo 0.02) (disponivel true))
f-15 (MAIN::combate (x 250) (combustivel 20.0) (y 250) (id 16) (agua 10) (tipo 3) (velocidade 1) (consumo 0.05) (disponivel true))
f-16 (MAIN::combate (x 116) (combustivel 20.0) (y 186) (id 14) (agua 10) (tipo 3) (velocidade 1) (consumo 0.05) (disponivel true))
f-17 (MAIN::combate (x 487) (combustivel 20.0) (y 320) (id 15) (agua 10) (tipo 3) (velocidade 1) (consumo 0.05) (disponivel true))

```

Figura 15. Factos da Base de Conhecimento

7.3 Geração dos incêndios

De forma a que os incêndios sejam gerados de forma aleatória foi necessário desenvolver um agente secundário responsável por esta tarefa. Assim sendo, este processo é tratado no comportamento *TickerBehaviour GerarIncendio*. Neste, o agente gera de cinco em cinco segundo posições aleatórias no mapa e a gravidade do incêndio e de seguida envia essa informação ao agente central e ao agente interface através de um mensagem com a *Performative Inform*. Perante a necessidade de enviar estas três informações de uma vez só, optamos por desenvolver uma classe *Incendio* que contém estas informações. Assim, na mensagem enviamos logo o incêndio, como se pode verificar de seguida.

```

ACLMessage msg= new ACLMessage(ACLMessage.INFORM);
msg.addReceiver(agente_central);
Incendio atual= new Incendio(gravidade,pos_x,pos_y);
try {
    msg.setContentObject(atual);
}catch (IOException e) {
    e.printStackTrace();
}
send(msg);

```

7.4 Coordenação dos agentes

O agente central faz a receção da informação dos agentes e dos incêndios, de modo a conseguir coordenar da melhor forma os movimentos dos agentes participativos. Esta

informação é recebida através de um *CyclicBehaviour RecebePosicao*. Este behaviour faz a receção de todas as mensagens que chegam ao agente central, através do método *receive*. Sendo depois verificado o tipo da mensagem e tratada a informação.

- Caso seja uma mensagem com *Performative Inform* cujo conteúdo é um objeto *Agente* então é guardada a informação do agente, caso não exista, ou atualizada, caso o agente já exista na base de conhecimento de Jess.
- Para o caso de ser uma mensagem com *Performative Inform* e o conteúdo seja um objecto *incêndio* é guardado na *List incendios*, e é chamado o método *EnviaCombate*, para se enviar algum agente participativo para combater esse incêndio.
- Se a mensagem for de *Performative Confirm* então foi recebida a confirmação da extinção de um dado incêndio que se encontra identificado no conteúdo, pelo que é possível alterar o estado desse incêndio para extinto e também calcular o tempo que este se encontrou ativo. É também enviada uma mensagem para o agente interface a informar que este incêndio já se encontra extinto.
- No caso de a *Performative Inform_if* então é feita alteração da disponibilidade de um agente, pelo que é alterada a informação do mesmo na base de conhecimento.

Tendo as informações dos agentes participativos bem como as dos incêndios, o agente central pode fazer a coordenação dos agentes. Assim, sempre que este recebe a informação de um novo incêndio ou então algum agente participativo fica disponível este verifica se pode mandar algum agente para combater algum incêndio, através do *OneShotBehaviour EnviaCombate*. Neste são percorridos todos os incêndios que ainda não estão extintos ou em combate de modo a verificar se algum agente participativo poderá ir combater-lo. Decidimos percorrer todos estes uma vez, que existe a possibilidade de nenhum dos agentes poder ir combater um dado incêndio num dado momento. O agente central envia para combate de um incêndio o agente mais rápido que tiver capacidades para o extinguir pelo que para verificarmos se algum agente poderá combater um dado incêndio, começamos por verificar quais estão disponíveis e que tem capacidade para o extinguir, ou seja, se tem combustível suficiente para chegar ao incêndio. Deste modo, de seguida obtemos o agente que demoraria menos tempo a extinguir o incêndio. É de salientar que o grupo definiu como estratégia estabelecer alguns agentes

como de reserva (dois drones e um caminhão) que apenas serão usados quando a gravidade do incêndio for máxima pelo que se torna necessário verificar qual a gravidade dos incêndios.

[Excerto que obtém o agente disponível mais rápido]

```
//percorrer todos os agentes
    if (l.isDisponibilidade() == true && consegueChegar(l, x, y)) {
        double dist = Math.sqrt(Math.pow((l.getPos().getX() - x), 2) +
            Math.pow((l.getPos().getY() - y), 2));
        double tempo = (dist / l.getVelocidade());
        if (min > tempo && gravidade < 3 && !agente_reserva(l)) {
            //guardado o agente
            min = tempo;
        } else if (min > tempo && gravidade == 3) {
            //guardado o agente
            min = tempo;
        }
    }
```

[Excerto que verifica a capacidade de chegar a um incêndio]

```
boolean fim = false;
double dist = Math.sqrt(Math.pow((xagent - x), 2) +
    Math.pow((yagent - y), 2));
if (dist * agent.getConsumo() < agent.getCombustivel())
    fim = true;
```

7.5 Combater incêndios

O agente participativo envia informação ao agente central de modo a permitir a melhor coordenação possível dos agentes por parte do mesmo, pelo que inicialmente os agentes participativos enviam toda a sua informação relevante através do *OneShotBehaviour EnviaInfo* com uma mensagem de *Performative Inform* e depois através do *TickerBehaviour EnviaPosicao* enviam a posição bem como combustível e água atual

enquanto se encontram em movimento, como é exposto de seguida. Para o envio destas informações, uma vez que eram várias optamos por criar a classe *Agente* de modo a criarmos um objecto com as informações pretendidas.

[Excerto mensagem a enviar durante o movimento]

```
ACLMMessage msg = new ACLMessage(ACLMMessage.INFORM);
Agente a = new Agente(
    combustivel_atual,
    agua_atual,
    pos.getX(),
    pos.getY()
);
try {
    msg.setContentObject(a);
    msg.addReceiver(agente_central);
    send(msg);
} catch (IOException e) {
    e.printStackTrace();
}
```

O processo de combate aos incêndios começa com a seleção e envio para o combate de um dado incêndio a um agente participativo. Como tal, o agente participativo vai receber os pedidos para combater incêndios que são processados pelo *CyclicBehaviour RecebePosição* que apenas recebe mensagens com as coordenadas do incêndio que o agente deverá combater, tal como podemos no seguinte excerto de código.

```
String[] coords = mensagem.getContent().split(";");
dest = new Posicao(Integer.parseInt(coords[1]), Integer.parseInt(coords[2]));
disponivel = false;
id_incendio = Integer.parseInt(coords[0]);
```

Assim, é alterado o destino do agente e sua disponibilidade, uma vez que este vai agora combater o incêndio, que tem um dado identificador que faz parte do conteúdo da mensagem. Tal deve-se ao facto de após o agente se derigir para o incêndio e o extinguir este deverá confirmar o agente central sobre tal:

```
ACLMessage msg = new ACLMessage(ACLMessage.CONFIRM);  
...  
msg.addReceiver(central);  
msg.setContent(""+id_incendio);  
send(msg);
```

Para a extinção do incêndio o agente participativo movimenta-se recorrendo ao *TickerBehaviour Movimento* no qual é calculado todo o movimento do agente, que tem em consideração a sua posição atual e o seu destino bem como a sua velocidade. Tal deve-se ao facto de o movimento do agente ser possível na oblíqua também. Para além disso, à medida que o agente se vai movimentando consome combustível.

[Excerto do cálculo da próxima posição e do combustível atual]

```
int next_x = (int) (velocidade * (dest.getX() - pos.getX()) / dist) + 1;  
int next_y = (int) (velocidade * (dest.getY() - pos.getY()) / dist) + 1;  
pos.setX(pos.getX() + cenas_x);  
pos.setY(pos.getY() + cenas_y);  
int dist_per = (int) Math.sqrt(Math.pow(cenas_x, 2) + Math.pow(cenas_y, 2));  
combustivel_atual -= dist_per * consumo;
```

O agente movimenta-se em várias situações: para chegar ao incêndio, para abastecer água e abastecer combustível. Tendo em conta a estratégia definida previamente o agente verifica se precisa de abastecer água e/ou combustível. Caso precise, então é verificado o posto de abastecimento mais próximo, sendo esse o novo destino do agente. Este agente ainda envia este movimento para o *agente interface*.

Estando extinto o incêndio e não sendo necessário abastecer o agente encontra-se disponível pelo que envia uma mensagem com Performative Inform_if ao agente central sobre o seu estado.

7.6 Interface

O agente interface recebe várias informações por parte de todos os outros agentes presentes no sistema de modo a se apresentar um mapa com todos os incêndios ativos, os postos de abastecimento e ainda os agentes participativos. Como tal, através de um *CyclicBehaviour* verificamos os diferentes casos:

- Caso a mensagem tenha *Performative Inform* e o seu conteúdo seja um *Agente* então podemos armazenar no *HashMap agentes_mapa* as coordenadas atuais e anteriores do agente, uma vez que assim é possível atualizar as posições no mapa.
- Se a mensagem tiver também uma *Performative Inform* mas o seu conteúdo seja um *Incendio* então podemos inserir na estrutura *incendios* os dados do incêndio de modo a colocarmos o mesmo no mapa e também temos as informações necessárias para as estatísticas.
- Caso a mensagem seja de *Performative Confirm* então podemos alterar a informação do incêndio indicado na mensagem, como o seu estado e tempo que esteve ativo. Informações úteis para as estatísticas.
- Por fim caso tenha *Performative Request* então é recebida informação de um incêndio que se encontra em combate, permitindo assim atualizar as estatísticas.

Com toda a informação processada e guardada tornou-se possível implementar um mapa 2D bem como as várias estatísticas que são apresentados na secção seguinte deste documento.

8 Demonstração

8.1 Mapa 2D

Com o intuito de expor os resultados da simulação, optamos por desenvolver uma interface gráfica do mapa 2D, em Swing, que ilustra a participação dos agentes na simulação. Assim, o mapa desenvolvido possui uma dimensão de 500 por 500, em que representamos os agentes participativos a verde, os incêndios ativos a vermelho, os locais de abastecimento de combustível a laranja e os de água a azul. É importante referir que quando um incêndio é extinto a posição volta ao normal e é possível ver os agentes em movimento quer para extinguir o incêndio bem como irem abastecer. Assim sendo, de seguida é possível observar uma parte do mapa que contém todos os componentes apresentados anteriormente.

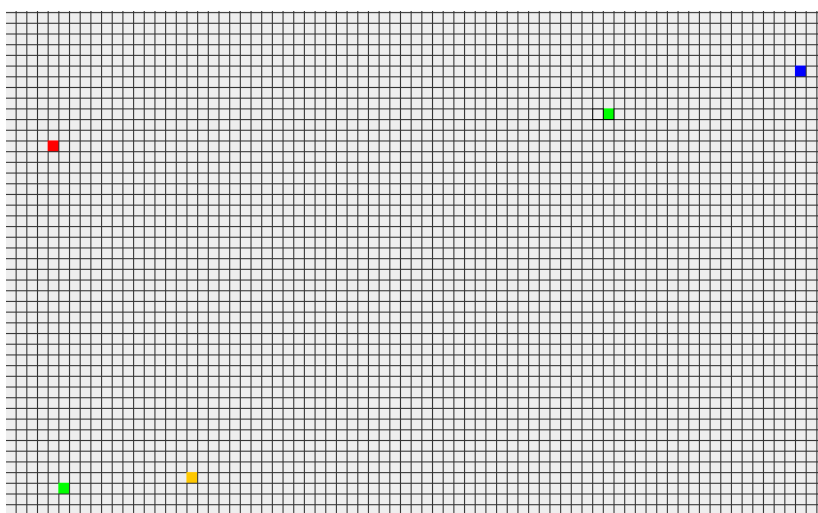


Figura 16. Mapa 2D

8.2 Estatísticas

De forma a apresentar a análise de performance dos agentes participativos, decidimos fazê-lo através de gráficos, recorrendo à API *JFreeChart*. Deste modo, desenvolvemos três gráficos. Um de barras, que contém o número de incêndios ativos, em combate e extintos. Outro também de barras, com o tempo mínimo em que um incêndio foi extinto, o tempo médio de extinção e o tempo máximo atingido. Por fim construímos um gráfico circular, que representa a percentagem dos tipos de agentes participativos que têm ido combater os incêndios.

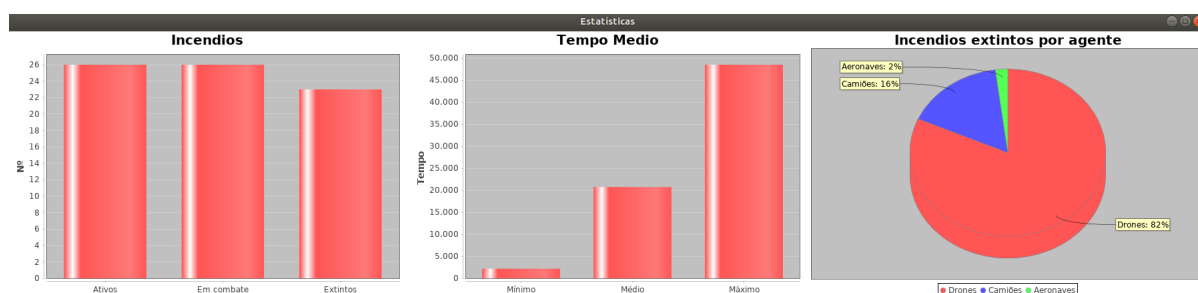


Figura 17. Resultados estatísticos

9 Conclusões e Trabalho Futuro

Concluído o projeto, podemos agora apercebermo-nos que foi adquirido conhecimento sobre os conceitos de Agentes, tendo sido estes aplicados quer na modelação como posteriormente na implementação deste que é um problema real. Para inicializar este projeto começamos por investigar quais as soluções existentes e em desenvolvimento no mercado para o combate a catastrofes naturais de modo a termos uma noção de como poderíamos desenvolver este projeto.

Assim, de seguida começamos a idealizar como iríamos estrutura-lo, através da modelação UML, que tornou assim possível demonstrar a estratégia por nós definida. Esta sofreu pequenas alterações de uma fase para a outra, uma vez que procuramos tentar simplificá-la ao máximo e ao mesmo tempo obter os melhores resultados possíveis. Como tal, as etapas mais trabalhosas foram a coordenação dos agentes, uma vez que pretendíamos minimizar o tempo médio dos incêndios, contudo existem incêndios com maior gravidade que devem ter maior prioridade e ao longo do tempo o número de incêndios vai aumentando bastante pelo que o número de agentes disponíveis em relação ao número de incêndios vai diminuindo bastante pelo que foram preciso tomar decisões importantes como deixar alguns agentes de reserva e apenas atuam em caso de emergência. Outro problema associado foi que uma vez que decidimos que o agente tomaria as suas próprias decisões relativamente a como proceder para apagar o incêndio, tivemos de decidir qual seria a melhor forma deste o fazer, como por exemplo o agente apenas precisa de abastecer água no fim da extinção do incêndio uma vez que poupamos combustível e tempo que poderia ser necessário para combater outro incêndio. Basicamente, foram vários os pequenos detalhes que tivemos de ter conta de modo a delinear a melhor estratégia de movimento e abastecimento.

Em última instância, consideramos que este foi um projeto bem conseguido em parte tendo em conta as estatísticas e resultados obtidos, contudo existem pontos que gostaríamos de ter melhorado e que seriam de fácil implementação contudo não foi possível por escassez de tempo, como por exemplo existirem vários agentes incendiários e os incêndios poderem se alastrar.

Bibliografia

- [1] Swati Basak, Neelam Modanwal, Bireshwar Dass Mazumdar, Multi-Agent Based Disaster Management System: A Review
- [2] Lei Zhou, Xianhua Wu, Zeshui Xu, Hamido Fujita, Emergency decision making for natural disasters: An overview. *International Journal of Disaster Risk Reduction*