



Departamento  
Engenharia  
Informática

# Agentes Autónomos e Sistemas Multiagente

2º Semestre, 2014/2015

*Relatório do Projecto*

## Agentes em *Os Almeida*s

Grupo N°2 – Alameda

*José Raposo - 64801*  
*Rodrigo Monteiro - 73701*  
*Tomás Pinho - 73009*

## Sumário

Este projecto consiste na criação de um cenário de jogo em que duas equipas de varredores de lixo devem competir para apanhar a maior quantidade de lixo, acumulando pontos para ganhar o jogo. Com este ambiente é possível estudar o comportamento de agentes inteligentes que devem pesar as suas necessidades juntamente com as da equipa para obter uma vitória, testando para o efeito várias arquitecturas diferentes.

Para a resolução do problema, as várias arquitecturas a serem testadas foram implementadas por ordem crescente de complexidade, começando num cenário em que os varredores agem de forma reactiva, até ao ponto em que têm uma implementação híbrida e comunicam entre si. De forma a estudar as capacidades cooperativas dos vários agentes, foram implementados os modelos Puramente Reactivos, Deliberativos e Híbridos, com comunicação nos dois últimos. Foi possível observar o efeito que o aumento da complexidade de comportamentos nos agentes teve na sua eficácia de acções e coordenação de equipa, e ainda verificar a existência de comportamentos emergentes não previstos na programação inicial.

### Palavras-chave:

Varredores de lixo, competição, comunicação, java, agentes reactivos, agentes deliberativos, agentes híbridos.

## Índice

Introdução	4
O Ambiente	5
Arquitecturas	6
Arquitectura Reactiva	6
Arquitectura Deliberativa BDI	6
Arquitectura Híbrida	6
Comunicação e Cooperação	7
Estudo Comparativo	8
Conclusões	9
Referências	10

## Introdução

O cenário do jogo em estudo é um tabuleiro com várias casas, que podem estar sujas. Duas equipas lutam entre si para apanharem o máximo de lixo possível, sendo que o jogo acaba quando não existir forma de uma das equipas conseguir ganhar, seja por diferença grande e pontos ou falta de elementos em jogo. Cada equipa tem 2 tipos de agentes diferentes que têm funções específicas no jogo. Existem vários elementos bónus que as equipas podem apanhar para fortalecer os agentes.

O objectivo do nosso projecto foi o de criar um cenário de jogo, com duas equipas virtualmente iguais, e observar a sua eficácia na competição contra a outra, que factores podem dar a vitória a uma em de vez de outra, e que estratégias criam para tentarem sair vencedoras.

Começámos por desenvolver agentes puramente reactivos. De seguida criámos agentes deliberativos e implementámos comunicação entre eles. Por fim foi criada a arquitectura híbrida, que faz a ponte entre as duas em cima.

O relatório começa por apresentar o cenário de jogo e os seus componentes, seguido das regras e funcionalidades possíveis. De seguida descreve as diferentes arquitecturas implementadas com detalhe para as percepções e acções dos agentes nelas. Continua com a especificação do protocolo de comunicação e termina com um estudo comparativo da competição em cada arquitectura e uma análise de dados e respectivas conclusões posteriores.

## O Ambiente

O ambiente de jogo criado para este projecto foi completamente feito de raiz. Consiste num tabuleiro quadrado com  $m \times m$  casas (o tamanho é variável). Nas casas pode estar lixo, e o objectivo principal dos agentes é recolher esse lixo e trazê-lo de volta à casa inicial.

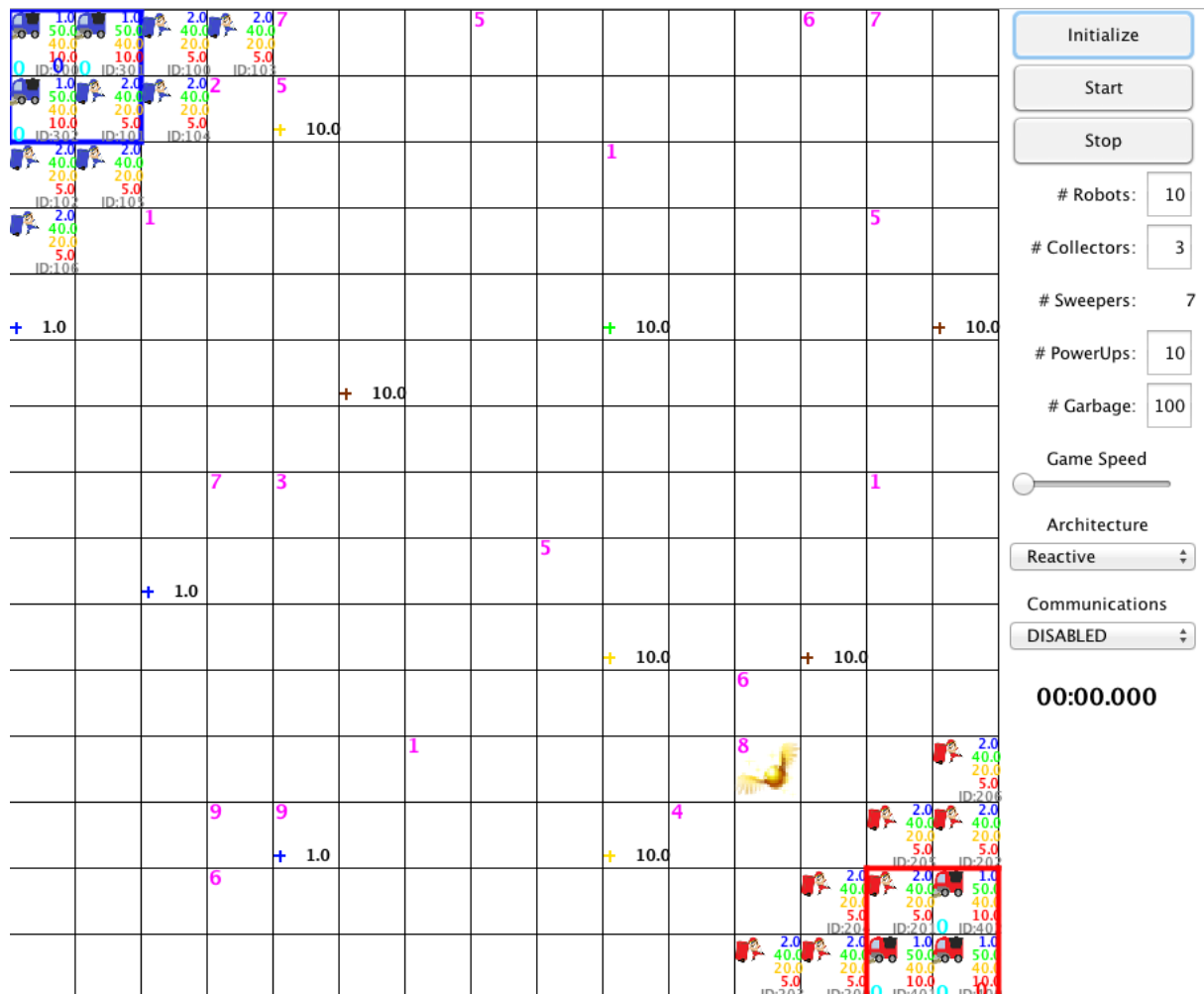


Figura 1: Cenário de jogo inicial, com ambas as equipas ao redor da sua base.

## Elementos de jogo

Os elementos do jogo são:

- o tabuleiro, limitado. Tem uma zona base constituída por 4 quadrados no canto de cada equipa;
- os agentes, 2 tipos por equipa, 2 equipas;
- os itens das casas:
  - Lixo (varrido ou por varrer)

- Power Ups (1 diferente para cada atributo do agente)
- *Snitch*

### **Regras**

O objectivo do jogo é ser-se a equipa vencedora, por ter o maior número de pontos. Os pontos são obtidos com a limpeza do lixo, ou apanhando um item especial, a *snitch*.

Todos os agentes têm 4 atributos, que influenciam as suas no decorrer do jogo: energia, resistência, velocidade e ataque.

Cada equipa tem dois tipos de agente:

- Os *Sweepers* são agentes mais ágeis, movimentam-se mais rapidamente, mas têm menos resistência. A função destes agentes é varrer o lixo.
- Os *Collectors* são agentes mais robustos, com mais resistência, mas mais lentos. O seu objectivo é ir apanhar o lixo nas casas em que já foi varrido, e transportá-lo e depositá-lo na base.

O lixo é então capturado em dois passos:

1. Um agente *sweeper* vai varrer lixo numa casa com lixo por varrer. Sinaliza essa casa ao *collector*;
2. O *collector* move-se então para essa casa, apanha o lixo e transporta-o para a base. Só no momento em que larga o lixo na base é que os pontos são incrementados na equipa.

O lixo varrido não tem cor, logo pode ser apanhado por um *collector* de qualquer equipa.

Por fim, cada acção consome energia. Se um agente ficar sem energia, tem que ficar parado durante um certo tempo enquanto a recupera.

Dois agentes não podem estar na mesma casa. Se tiverem em casas adjacentes, dois agentes adversários podem entrar em combate. Neste caso, a resistência de cada agente vai reduzindo consoante a força do adversário.

Quando um agente ficar sem resistência, morre e desaparece do jogo.

### **Fim do jogo**

O jogo acaba quando:

- Todo o lixo foi varrido e apanhado;

- Morrem todos os *collectors* da equipa que está a perder logo não é possível apanhar mais lixo;
- O lixo disponível no tabuleiro já não permite dar a volta ao jogo;
- Morrem todos os agentes da equipa que está a ganhar.

### **Interacção dos agentes com o ambiente de jogo**

As percepções dos agentes mudam consoante a arquitectura implementada, logo serão descritas nas secções respectivas.

As acções possíveis resumem-se em três grupos.

- Movimentação no cenário – os agentes podem mover-se em 4 direcções (cima, baixo, esquerda e direita);
- Iteracção com os itens no cenário – os agentes podem varrer/colectar lixo e podem apanhar os restantes itens;
- Atacar agentes – agentes de equipas adversárias podem-se atacar.

### **Classificação do ambiente**

O ambiente pode ser descrito como:

- Inacessível: os agentes só têm conhecimento do que se passa nas casas imediatamente à sua volta na arquitectura reactiva, ou num raio maior e à volta dos amigos na arquitectura deliberativa;
- Não determinístico: a mesma acção dos agentes não produz sempre o mesmo resultado (o estado interno pode não estar actualizado).
- Dinâmico: as posições dos jogadores, a posição da snitch e o estado do lixo vai-se alterando ao longo do jogo;
- Discreto: é um ambiente finito, com um conjunto reduzido de acções e um conjunto um pouco maior de resultados possíveis;
- Não-episódico: cada acção de cada agente tem impacto no decorrer do jogo.

## Arquitecturas

No nosso projecto desenvolvemos agentes puramente reactivos, deliberativos, e com arquitectura híbrida.

O agente reactivo descreve-se por ter uma acção para cada percepção.

O agente deliberativo tem um estado interno com um conjunto de crenças (percepção do estado do jogo) e tem um conjunto de desejos, que usa para traçar um plano de acção para atingir objectivos a médio/longo prazo.

O agente híbrido funciona como o deliberativo mas tem também a componente reactiva que o ajuda a responder rapidamente a vários eventos do ambiente.

### ***Arquitectura Reactiva***

#### **Interacção do agente com o ambiente**

##### ***Percepções***

Nesta arquitectura os agentes conseguem apenas ver o conteúdo detalhado da casa em que estão (se tem um powerUp, de que tipo é; se tem lixo, se está varrido ou não), ou se está um inimigo nalguma casa adjacente à dele. Os collectors conseguem ainda perceber se estão na base.

##### ***Acções***

Ambos os agentes podem apanhar powerUps e snitch se estiverem na mesma casa, e podem atacar inimigos se tiverem em casas adjacentes. Ambos podem-se mexer 1 casa de cada vez nas 4 direcções principais.

Os *Sweepers* podem varrer lixo em casas que tenham lixo por varrer.

Os *Collectors* podem apanhar lixo nas casas que o tenham já varrido.

Os agentes podem ainda descansar se ficarem sem energia.

A arquitectura dos agentes reactivos é bastante simples, pois não tem em conta o histórico de acções que o agente executou. Interessa-lhe apenas o presente. Assim, para cada percepção do ambiente, o agente executa uma acção.



Ambos os tipos de agentes têm um comportamento semelhante em tudo menos na interação com o lixo. O comportamento geral é descrito pelas seguintes acções:

- *checkPowerUp* → *grabPowerUp* - Se existir um *powerUp* na casa, independentemente do tipo, o agente apanha-o
- *checkSnitch* → *grabSitch* - Se o agente estiver na casa da *snitch*, apanha-a
- *checkEnemy* → *attack* - Se existir um inimigo numa das 4 casas adjacentes, o agente ataca-o
- *else* → *move* - se nenhuma das anteriores se verificar, o agente move-se para uma direcção aleatória
- *energy == 0* → *rest* - na situação especial em que o agente fica sem energia, tem que ficar parado durante algum tempo

Cada agente tem, contudo, uma ou duas acções/reacções específicas.

Os *sweepers* são os agentes que varrem o lixo, por isso têm a regra extra:

- *checkGarbage* → *sweepGarbage* - varrem lixo, 1 a 1 para o estado de lixo varrido

Os *collectors* precisam de apanhar lixo varrido, ignorando o outro. E se estiverem na casa principal, precisam de o depositar lá:

- *checkSwipedGarbage && capacity < full* → *grabGarbage* - se tiver encontrado lixo varrido e ainda não tiver atingido a capacidade de lixo que pode carregar, apanha o lixo.
- *hasGabage && checkBase* → *dropTheGarbage* - se estiver na base, e tiver lixo consigo, deposita-o.

Collector	Sweeper
<i>rest</i>	<i>rest</i>
<i>grabSnitch</i>	<i>grabSnitch</i>
<i>grabGarbage</i>	<i>sweepGarbage</i>
<i>dropTheGarbage</i>	<i>GrabPowerUp</i>
<i>grabPowerUp</i>	<i>attack</i>
<i>attack</i>	<i>move</i>
<i>move</i>	

*Tabela 1: Ordem de importância das acções por agente, do mais importante em cima para o menos em baixo.*

Como é possível ver na tabela 1, o mais importante é o *rest*, por uma razão: é obrigatório. Se um agente fica sem energia, tem que descansar, não pode fazer mais nada.

De seguida, apanhar a *snitch* está no topo da lista por dar uma grande quantidade de pontos imediatos. As acções relativas ao lixo vêm a seguir (varrer ou apanhar/largar) precisamente por ser o objectivo principal do jogo, e o que permite acumular pontos para a vitória de alguma equipa.

Seguidamente o duplo *grabPowerUp* e *attack*, têm esta ordem por ser considerado que um *powerUp* dá sempre uma vantagem estratégica numa luta, aumentando qualquer um dos 4 atributos. Assim se um agente puder apanhar um tanto antes, melhor. E só depois ataca. Por fim, se nada mais resta a fazer, o agente move-se de forma aleatória, para uma das 4 posições adjacentes que estejam disponíveis, isto é, dentro dos limites do tabuleiro e não ocupadas por nenhum outro agente.

## Arquitectura Deliberativa BDI

### Interacção do agente com o ambiente

#### Percepções

Na arquitectura deliberativa, o alcance de visão dos agentes aumenta para um raio de 2 casas em redor do agente, como é possível ver na fig.2. Sendo assim, ele tem percepção do conteúdo de todas essas casas. Para além disso e devido à comunicação, ele sabe o conteúdo de outras casas no seu mapeamento interno do tabuleiro, se outro agente lhe comunicou. Desta forma, as crenças dos agentes são os conteúdos das casas que este tem mapeadas, ou seja, se as casas têm powerUps, a snitch, lixo varrido, lixo por varrer, agentes amigos, agentes inimigos ou se faz parte da base.

Ambos os agentes têm as mesmas crenças, mas os sweepers não dão atenção ao lixo varrido e à base, tal como os collectors não se importam com o lixo por varrer.

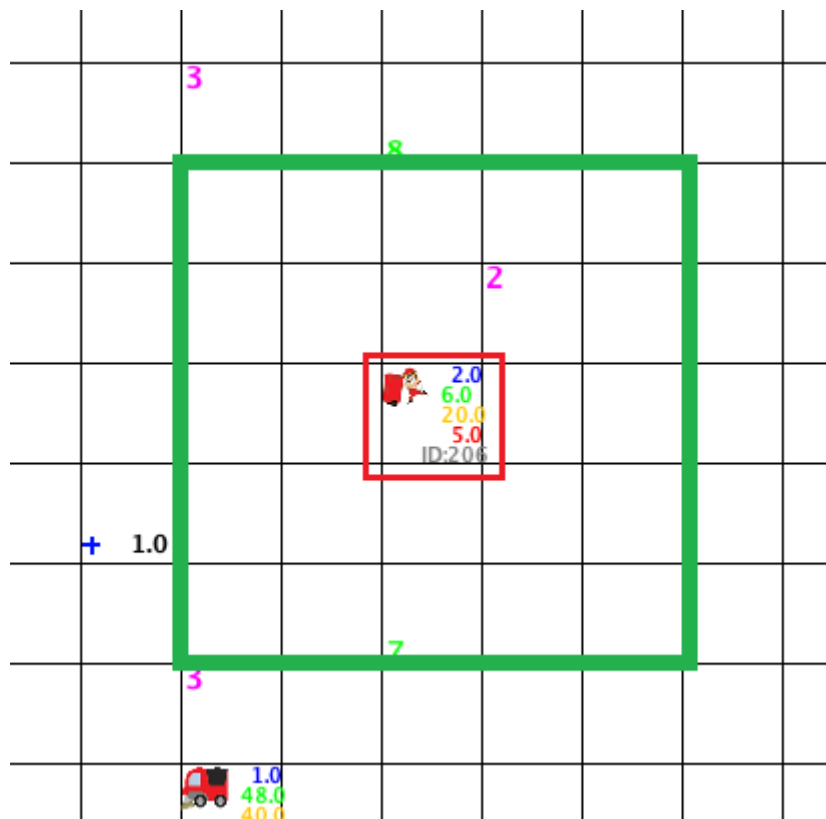


Figura 2: O agente no meio consegue perceber todas as casas dentro do quadrado verde.

#### Acções

As acções disponíveis são as mesmas que na arquitectura reactiva.

### ***Desejos***

Os desejos de ambos os agentes são os seguintes: Apanhar\_powerUps, Apanhar\_lixo, Varrer\_lixo, largar\_lixo, proteger\_amigo, atacar\_inimigo, apanhar\_snitch, explorar, seguir\_amigo.

### ***Intenções***

As intenções são dos agentes são o movimento final de um plano, criado após filtragem das intenções. Por exemplo, a intenção de um *collector* apanhar o lixo na casa (x,y), que está a várias casas de onde ele está agora.

### ***Geração de Desejos***

Para gerar os desejos do agente, uma função de opções avalia as crenças do agente e usa um conjunto de restrições para dar um peso a cada desejo. Ou seja, cada desejo tem um peso base associado, e consoante a situação, o desejo é adicionado ou não a uma lista de desejos. Se na crença de um agente é verificado que existe um powerUp numa casa que o agente conhece, então “apanhar powerUp” é adicionado como desejo à lista dos desejos. Se o collector tiver atingido a capacidade máxima de lixo que pode transportar, então o desejo de largar o lixo na base também é adicionado, com um peso supostamente ainda maior (algumas situações podem alterar este peso, como o agente ter pouca vida e ter um powerUp de vida perto, então adiciona o desejo de “apanhar powerUp” com mais peso).

Quando a lista de desejos é calculada, uma função de filtragem vai pegar nela e simplesmente olhar para o desejo com maior peso. Escolhe esse desejo e cria a intenção de o concretizar, associando uma acção (grabPowerUp) a uma casa (x,y).

Por fim, esta intenção é passada para uma função de planeamento que gera o caminho até à intenção final tendo em conta as crenças actuais do agente.

### ***Planeamento***

Os tipos de planos construídos são caminhos da posição actual do agente até à casa da posição final, concluindo com a acção a realizar (grabSnitch, dropTheGarbage, etc.). Os caminhos são calculados com base na distância de Manhattan mais curta entre ambas as posições, evitando agentes que estejam no caminho imediato (amigos ou inimigos, a menos que a intenção seja atacar um agente, claro).

Devido à facilidade da intenção ser alterada, ou deixar de ser possível executar (por exemplo um agente percebe a snitch duas casas à frente da sua, mas no movimento seguinte ela já não está lá, ou um agente atravessa-se no caminho), é necessário replanear com frequência. Por isso a validade das intenções está sempre a ser reavaliada e a reajustada conforme as necessidades.

## ***Arquitectura Híbrida***

A arquitectura híbrida construída é do tipo “horizontal layering - one pass”.

Tem 3 camadas de controlo:

- Comportamental: componente reactiva do agente;
- Planeamento: componente deliberativa;
- Comunicação

As percepções efectuadas pelo agente chegam ao mesmo tempo à camada comportamental como à de planeamento.

O objectivo é que quando ocorre uma percepção que exige uma acção imediata (por exemplo a snitch está na casa onde o agente se encontra, ou está um inimigo a atacá-lo) a camada comportamental reage imediatamente. Quando não é necessário, a camada de planeamento continua a construir planos a médio/longo prazo, permitindo assim a continuação do comportamento do agente em direcção a um objectivo.

De resto a arquitectura híbrida é em tudo semelhante à deliberativa.

## Comunicação e Cooperação

A comunicação entre agentes, na arquitectura Deliberativa e na Híbrida é feita segundo um protocolo de broadcast, dentro da mesma equipa. Ou seja, quando um evento ocorre que justifique uma comunicação, o agente que percepção o evento envia 1 de 3 tipos de mensagens para todos os membros da sua equipa, o que provoca uma actualização das crenças de todos os elementos (pela função BRF). As mensagens são todas acompanhadas das respectivas percepções e posição das mesmas.

Quando um agente recebe a mensagem, revê as crenças que tem e revê os desejos, podendo ou não alterar o plano. Isto significa que o agente não é obrigado a “responder” a nenhuma mensagem (i.e. agindo de acordo com a necessidade da mensagem).

Os tipos de mensagens existentes são:

- *WARNING*: quando detecta inimigos perto, isto é, no seu alcance de visão;
- *HELP\_REQUEST*: é enviada quando o agente tem pouca vida e precisa de ajuda para se defender;
- *INFORMATION*: informação sobre o conteúdo de casas. A mensagem base a ser enviada caso um agente percepcione lixo, powerUps ou snitch.

## Estudo Comparativo

### *Preparação*

Nos testes seguintes vamos comparar a execução de 5 variantes das 3 arquitecturas implementadas, nomeadamente:

- Arquitectura Reactiva
- Arquitectura Deliberativa sem comunicação
- Arquitectura Deliberativa com comunicação
- Arquitectura Híbrida sem comunicação
- Arquitectura Híbrida com comunicação

As métricas que vamos utilizar para medir a performance dos agentes em cada cenário são:

- Tempo decorrido até o jogo terminar;
- Motivo do jogo acabar;
- Ponderação dos pontos obtidos
- Ponderação do lixo por varrer restante no jogo
- Ponderação dos agentes que sobreviveram

Para obter estas métricas vamos avaliar o mesmo cenário de jogo com as seguintes opções:

- 10 agentes por equipa (3 collectors e 7 sweepers)
- 100 unidades de lixo
- 10 powerUps
- tabuleiro de 15\*15 casas

Os gráficos e tabelas vão usar a seguinte legenda:

<b>Arquitectura</b>	Reactiva	Deliberativa com comunicação	Deliberativa sem comunicação	Híbrida com comunicação	Híbrida com comunicação
<b>Abreviatura</b>	React.	Delib. + Com	Delib	Hibr + Com	Hibr

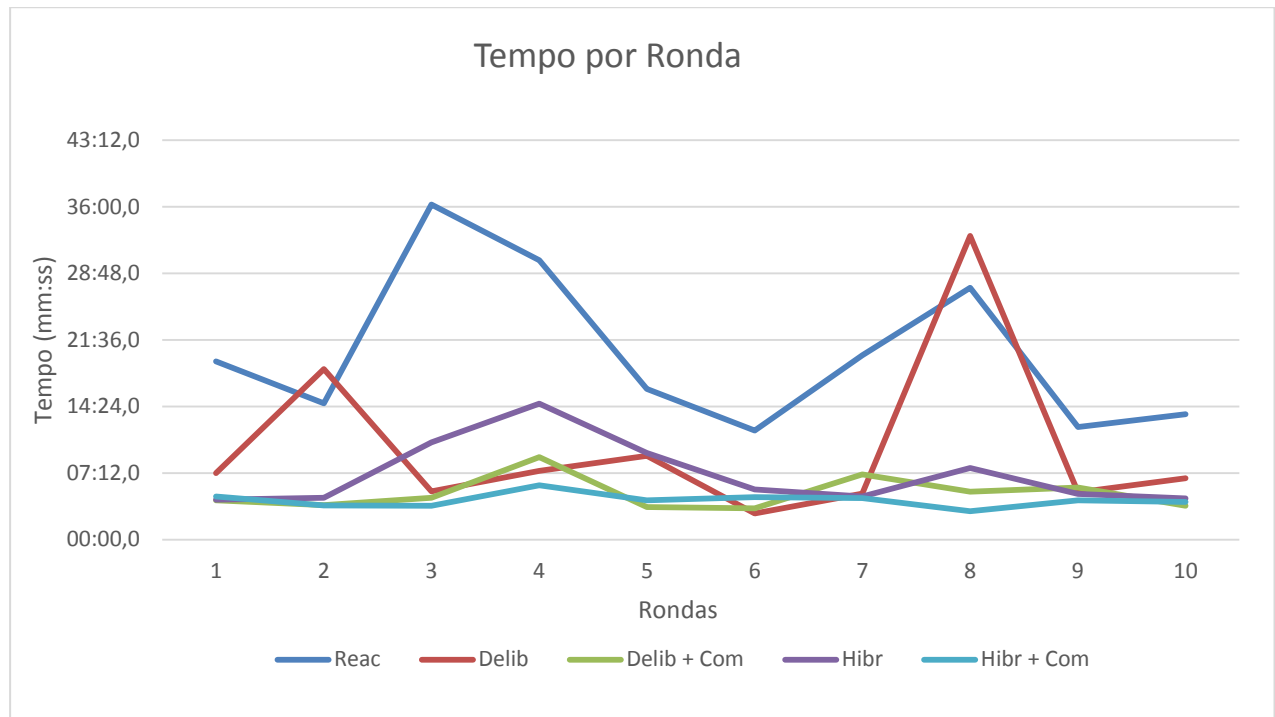
**Testes**

Figura 3: Tempo em cada arquitectura por ronda, com valores medidos em minutos:segundos.

Na fig.1, onde é medido o tempo, conseguimos observar a evolução da eficiência em ambas as equipas de chegarem ao final do jogo rapidamente. A arquitectura reactiva tem tempos substancialmente superiores a todas as outras. A híbrida com comunicação é a mais rápida. É interessante verificar que as arquitecturas sem comunicação atingem tempos superiores e menos constantes relativamente às suas pares com comunicação.

<b>Causas de morte</b>	React	Delib	Delib+Com	Hibr	Hibr+Com
Já não há lixo suficiente para alterar o resultado	1	10	9	7	9
Acabou o lixo e a Snitch foi apanhada	0	0	1	3	1
Uma equipa foi eliminada	7	0	0	0	0
Já não há Collectors numa das equipas	2	0	0	0	0

Tabela 2: Causas de morte mais comuns por arquitectura.

Pela tabela 2 vemos imediatamente uma diferença grande entre as arquitecturas Reactiva e as restantes, que é a causa de morte. Na reactiva, a maioria das mortes é por eliminação dos agentes, raramente há uma equipa que sobrevive até ao fim, enquanto nas outras arquiteturas o jogo é mais civilizado e termina maioritariamente por falta de lixo.

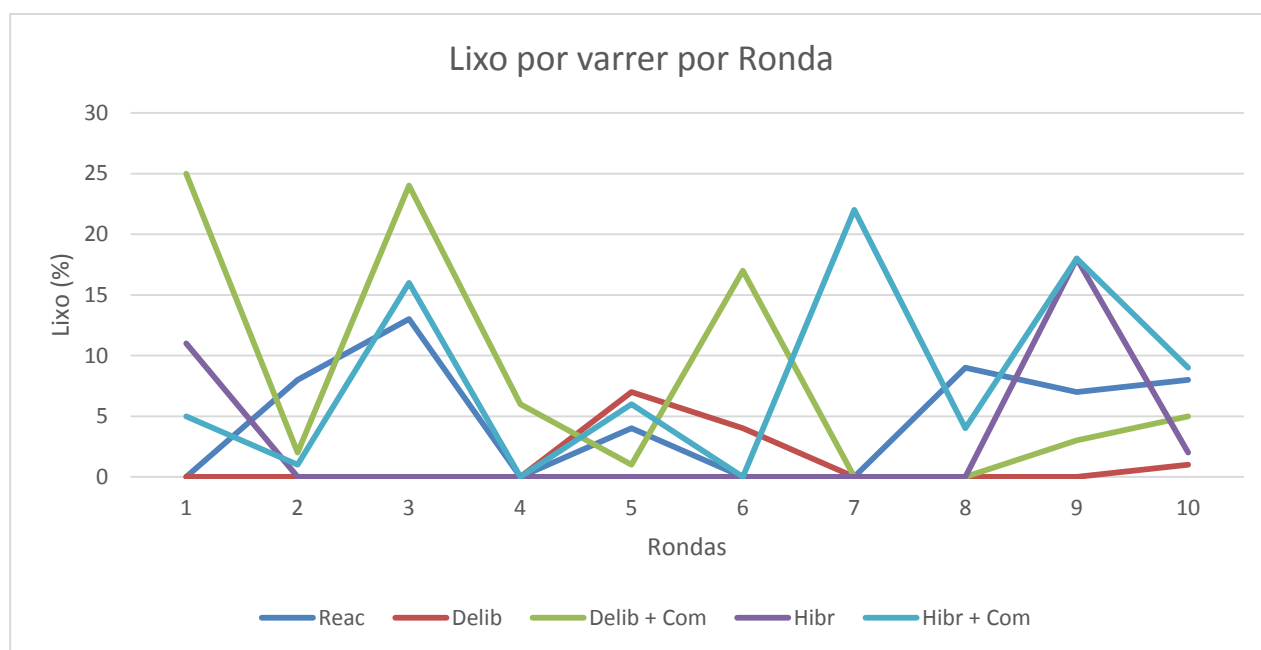


**Agentes Vivos por Ronda (equipas AZUL - VERMELHA)**

Reac	5-0	1-4	0-2	4-0	0-3	0-5	1-2	2-1	0-5	2-4
Delib	8-9	6-9	10-10	9-6	7-9	7-7	9-7	8-6	8-8	4-7
Delib + Com	8-9	10-10	8-9	9-10	10-7	10-9	9-10	10-9	6-6	10-9
Hibr	8-9	9-9	9-10	9-9	8-10	7-7	8-8	8-10	8-10	8-10
Hibr + Com	8-8	8-9	10-9	9-6	9-8	9-8	7-9	7-10	8-10	10-7

*Tabela 3: Número de agentes que sobreviveu até ao fim do jogo. Não foram discriminados entre tipos pois em todos os testes foi relativamente aleatório que agentes sobreviveram.*

O número de agentes vivos dá uma justificação às observações anteriores, pois aqui é possível verificar que quando se sai da arquitectura reactiva, a quantidade de agentes que sobrevive em ambas as equipas aumenta imenso, principalmente nas equipas com comunicação em que quase sempre uma das equipas chega ao fim inteira (apesar da diferença para as pares sem comunicação não é significativa). A arquitectura reactiva, é uma chacina e são poucos os agentes que resistem ao confronto



*Figura 4: Percentagem de lixo que, no fim do jogo, não foi varrido.*

O comportamento mais curioso que se observa na fig. 4 é que as arquitecturas com comunicação deixam mais lixo por varrer no fim, enquanto que nas restantes o lixo é mais frequentemente varrido (independentemente de ser apanhado ou não).

Pontos por Ronda (equipas AZUL - VERMELHA)

Reac	0-25	0-28	29-0	25-10	0-25	38-0	0-25	25-0	29-0	0-51
Delib	13-54	52-37	57-25	1-36	0-55	64-0	28-53	61-47	30-52	14-51
Delib + Com	67-15	52-36	11-53	40-52	52-7	9-53	72-31	55-25	54-22	34-53
Hibr	30-58	7-51	35-62	25-55	53-31	39-63	56-48	38-55	27-59	53-39
Hibr + Com	57-35	56-19	18-60	62-49	23-55	19-52	22-52	28-61	55-0	56-16

Tabela 4: Pontos finais por cada ronda, por equipa.

Por fim, esta tabela não nos diz muito mais que o que foi possível observar nos dados anteriores. Na arquitectura reactiva os pontos são sempre inferiores aos das restantes, porque o jogo acaba muito antes de se terem apanhado os pontos todos, como foi possível verificar nas causas de fim de jogo. Nas restantes arquitecturas não é visível um padrão que as distinga umas das outras, senão que mais frequentemente os pontos estão mais equilibrados (não esquecendo que a snitch dá uma vantagem de pontos extra, e regra geral garante a vitória se for apanhada).

Pelas observações anteriores, é possível avaliar e justificar algumas alterações curiosas no comportamento dos agentes ao longo das diferentes implementações. Na arquitectura reactiva os agentes agem de forma semi-aleatória (se existir lixo varrem/apanham, se não movem-se aleatoriamente, etc.), isto faz com que sempre que entram em contacto com um agente inimigo lutam até à morte, e que pode estar o lixo todo varrido no campo, que se nenhum colector passar por lá ninguém o apanha, e mesmo que algum colector tenha lixo consigo não vai tentar ir para a casa base, mas continua a mover-se aleatoriamente, justifica a grande quantidade de mortes, e a reduzida quantidade de pontos, para não falar do tempo extra que demora o jogo a terminar. Quando é implementado um comportamento deliberativo (e posteriormente híbrido) existe também um aumento das capacidades perceptivas dos agentes, e uma intenção associada às suas acções. A primeira consequência disto é a redução substancial do tempo de jogo. A segunda é um curioso comportamento emergente que leva os agentes a não se matarem tanto, fugindo mesmo das lutas e priorizando a recolha de pontos acima de tudo. Finalmente com a implementação da comunicação, nota-se uma segunda redução significativa no tempo de jogo (agora os agentes sabem o que os outros agentes andam a fazer. Se um acabou de limpar lixo, outro pode ir logo lá apanhá-lo), e ao mesmo tempo houve um pequeno aumento da quantidade de lixo por varrer que restou no fim de jogo.

Isto pode ser justificado por mais rapidamente os agentes realizarem a dupla varrer/colectar lixo e assim mais rapidamente atingirem uma pontuação que lhes permite ganhar a ronda sem ter que apanhar o lixo todo.

Uma última nota, relativamente às arquitecturas híbrida e deliberativa, não se nota muita diferença entre ambas sem ser numa pequena melhoria de tempo. Isto pode indicar que neste cenário em particular, o planeamento é muito mais importante e poucas são as situações em que o comportamento reactivo é preferível, sendo a arquitectura híbrida apenas um versão ligeiramente mais eficiente que a deliberativa.

## Conclusões

O trabalho realizado foi ao encontro das expectativas iniciais, visto que foi possível implementar soluções para todas as propostas apresentadas. Foram implementadas suficientes variantes das arquitecturas para serem observáveis as vantagens de utilizarmos arquitecturas mais complexas. A competição tornou-se cada vez mais eficiente e o comportamento dos agentes mais elaborado e interessante.

Foi claramente visível, pelo estudo comparativo, a importância da implementação da deliberação para a realização de tarefas mais complexas e comportamentos a médio/longo prazo.

Curiosamente, ou não, a implementação de capacidades de comunicação nos agentes, mesmo num cenário de jogo relativamente simples, teve um impacto grande e proporciona uma interessante lição.

Finalmente, o resultado final do projecto não está isento de falhas. Planeamentos mais complexos para atingir a vitória poderiam ser implementados, nomeadamente mais técnicas de combate, ou estratégias “anti-jogo”. Algo que poderia ser interessante seria dar estratégias com pesos diferentes a ambas as equipas e observar quais saíam vitoriosas.

## Referências

- [1] Michael Wooldridge, An Introduction to MultiAgent Systems, John Wiley and Sons, 2002.
- [2] Diapositivos de apoio às aulas teóricas da cadeira de Agentes Autónomos e Sistemas Multi-Agente