



Universidade do Porto

---

Faculdade de Engenharia

# FEUP

## Santiago



Ricardo Silva – ei11079@fe.up.pt  
Hugo Freixo – ei11086@fe.up.pt

# Índice

Introdução.....	3
Objetivo.....	5
Especificação.....	6
Desenvolvimento.....	7
Comunicação Jogador/Gestor.....	10
Conclusões.....	11
Melhoramentos.....	11
Recursos.....	12

## Introdução

O tema do jogo Santiago é o cultivo e irrigação de campos. Para isso, um número de peças representando diferentes tipos de plantação vão sendo acrescentadas ao jogo em cada ronda. As peças são distribuídas pelos jogadores, sendo depois colocadas no tabuleiro de jogo juntamente com um marcador do jogador que indica igualmente a quantidade produzida nessa plantação. Em cada ronda, quem licitar menos toma o papel de “aguadeiro” (administrador do canal de água) e decide para onde é que o canal será expandido nessa ronda. Os outros jogadores podem fazer sugestões, propondo pagar determinadas quantias para que a decisão decorra em seu favor. A decisão final é sempre do aguadeiro.

No final de cada ronda, os jogadores verificam o fornecimento de água às diferentes posições do tabuleiro. Se uma plantação não estiver a ser irrigada, a sua produção é reduzida drasticamente; se isso acontecer por mais do que uma vez, a plantação pode secar. No final do jogo, apenas as áreas cultivadas contam. Cada plantação é contada de acordo como o seu tamanho (área contígua do mesmo tipo). Os marcadores dos jogadores presentes em cada área ditam o número de pontos obtidos por cada jogador.

O tabuleiro do jogo tem este aspeto:



(Figura 1: Tabuleiro do jogo Santiago)

Sendo as telas castanhas campos ainda não cultivados, as telas de cores campos cultivados, os quadrados dentro dos campos cultivados trabalhadores desses campos e as linhas azuis o fluxo do rio.

O objetivo deste trabalho é criar um conjunto de agentes JADE que simulem vários jogadores e um agente que controle as diferentes fases do jogo para que, seja possível simular um jogo completo de Santiago.

No final do trabalho o grupo espera conseguir simular várias partidas de Santiago e verificar que diferentes agentes com diferentes estratégias de jogo consigam melhores ou piores resultados consoante a sua estratégia.

## Objetivo

O objetivo do projeto consiste na implementação de um Sistema Multiagente (SMA) que realize um jogo de tabuleiro. Um SMA é um sistema computacional onde os agentes cooperam ou competem para atingir objetivos pessoais e/ou coletivos.

Os Sistemas Multiagente são sistemas compostos por múltiplos agentes, que exibem um comportamento autónomo mas ao mesmo tempo interagem com os outros agentes presentes no sistema. Estes agentes exibem duas características fundamentais:

- São capazes de agir de forma autónoma, tomando decisões, levando à satisfação dos seus objetivo;
- São capazes de interagir com outros agentes utilizando protocolos de interação social inspirados nos dos humanos e requerendo coordenação, cooperação e negociação;

No caso concreto do Sistema a desenvolver, poderão participar três a cinco jogadores (agentes *Player*). O Sistema deverá permitir a participação de qualquer tipo de agente que obedeça a um conjunto de critérios e que reconheça alguns tipos de mensagens, sendo garantida a segurança das regras do jogo por um agente exterior (agente *Manager*) responsável por todas as tarefas de controlo e gestão do jogo.

Como objetivo do projeto foi tomado contacto com a programação usando sistemas multiagente, no caso concreto a plataforma utilizada para o desenvolvimento dos agentes foi o JADE. Os agentes desenvolvidos são suficientemente inteligentes de forma a serem capazes de competir entre si para obtenção de uma maior pontuação individual.

## Especificação

Para desenvolver o jogo Santiago foram criados 2 tipos de agentes, o agente *Manager* e o agente *Player*. O *Manager* gere todos os agentes *Player* e faz com que toda a lógica de jogo funcione. O agente *Manager* envia mensagens aos jogadores onde pergunta quais são os subornos dos *Players*, envia as escolhas de terrenos possíveis e o estado atual do tabuleiro para que o agente *Player* escolha a sua jogada. O Agente *Player* envia mensagens ao *Manager* para responder aos seus pedidos, dá o seu suborno, escolhe o campo e a posição do mapa para colocar o campo de cultivo, dá o suborno para o aguadeiro e escolhe a posição da água caso seja o aguadeiro.

Os agentes *Player* têm diferentes inteligências e agem de forma diferente consoante a sua lógica. Existem 4 tipos de lógica: *Random*, *Spender*, *Saver* e *Mixed*. O *Random* joga aleatoriamente em todos os momentos do jogo.

O *Spender* gasta o máximo de dinheiro possível de forma a ir buscar mais pontos à soma final de pontuação, nos subornos tenta sempre dar mais que os outros para ter a vantagem de um campo extra, na colocação dos campos tenta colocar ao lado de um campo do mesmo tipo, não se importando com o fluxo da água, para ir buscar o máximo de pontuação possível, no suborno para a água tenta oferecer mais que os outros em zonas à beira de campos dele e quando é o aguadeiro procura ajudar-se a si próprio ignorando os subornos.

O *Saver* poupa o máximo possível de dinheiro para ter mais pontuação durante o jogo e não vai buscar tantos pontos no fim, nos subornos se 20% do seu dinheiro ultrapassar todos os adversários ele paga esse valor de resto procura ser o aguadeiro dando 0 *Escudos*, na colocação dos campos procura campos que já estejam irrigados para não ser necessário pagar subornos ao aguadeiro, no suborno para a água utiliza o mesmo algoritmo dos subornos normais e quando é o aguadeiro escolhe o maior suborno oferecido.

O *Mixed* tem características do *Saver* e do *Spender*, nos subornos age como um *Saver* e em todos os outros momentos age como um *Spender*.

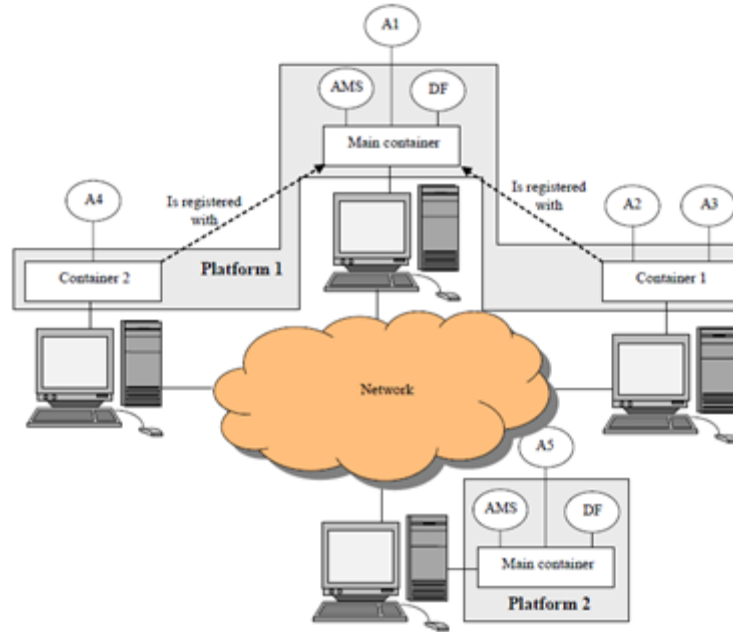
## Desenvolvimento

Para a realização deste projeto iremos utilizar a ferramenta Java Agent DEvelopment Framework (JADE). Esta ferramenta inclui:

- Um *runtime environment* onde os agentes JADE podem “viver” devem estar a correr numa máquina antes de criar outros agentes;
- Uma biblioteca em que os programadores podem/devem usar para a criação de novos agentes;
- Um conjunto de ferramentas gráficas que permitem a monitorização da atividade dos agentes em execução.

Cada instância de JADE em execução é chamado de *Container*, pois pode conter vários agentes. O conjunto de todos os *Containers* ativos fazem parte de uma *Platform*. Um único *Main-Container* terá de estar sempre ativo na *Platform* e todos os *Containers* terão de ser registados neste no momento que são executados. Como resultado o primeiro *Container* numa *Platform* terá de ser o *main* e os restantes serão designados como *normal (non-main)*, estes terão de ser “avisados” de como encontrar o *main (host e port)* para se poderem registar.

Caso outro *Main-Container* seja iniciado algures na rede irá constituir outra *Platform* na qual novos *Containers* se podem registar. A figura abaixo ilustra este conceito através de um cenário exemplo em que duas *Platform* JADE são compostas, respectivamente, por 3 e 1 *Containers*. Agentes JADE são identificados por nomes únicos e desde que tenham conhecimento do nome de outros agentes eles podem comunicar entre si independentemente da sua localização: no mesmo *Container* (agentes A2 e A3), em diferentes *Containers* mas na mesma *Platform* (agentes A1 e A2), ou em diferentes *Platforms* (agentes A4 e A5).



(Figura 2: Exemplo de agentes JADE em diferentes Containers e Platforms)

Para além de aceitar registos de outros Containers o *Main-Container* difere dos outros pois contém dois agentes especiais, que são inicializados automaticamente quando o *Main-Container* é executado, AMS e DF (Figura 3).

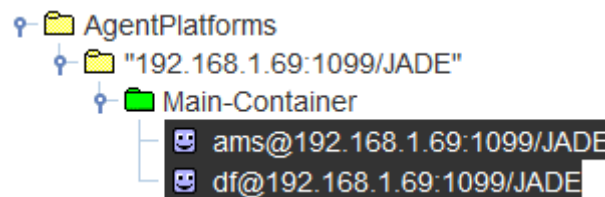
*Agent Management System (AMS):*

- Supervisão sobre o acesso e uso da plataforma;
- Mantém diretoria de identificadores de agentes e o seu estado;
- Todos os agentes se registam em AMS;

Em cada plataforma existe apenas um AMS.

*Directory Facilitator (DF):*

- DF fornece serviços de páginas amarelas;
- Agentes podem registar os seus serviços no DF ou questiona-lo;
- Em cada plataforma existe pelo menos um DF.



(Figura 3: AMS e DF)

Para além destes o JADE disponibiliza outros agentes já predefinidos, tais como, RMA , *Dummy Agent* e o *Sniffer Agent*.



*Remote Monitoring Agent (RMA):*

- Consola gráfica para controlo da plataforma;
- Visualização do estado de agentes e *Containers*.

*Dummy Agent:*

- Ferramenta de monitorização e *debugging*;
- Permite enviar, receber e visualizar mensagens ACL.

*Sniffer Agent:*

- Quando se “espia” um agente as mensagens dirigidas ou enviadas a este são visualizadas;
- Quando um agente ou *Container* é criado ou destruído o *Sniffer* é informado automaticamente pelo AMS.

## Comunicação Jogador/Gestor

De forma a manter atualizado o estado de jogo em cada um dos jogadores e a executar as jogadas que estes pretendam fazer foi estipulada uma arquitetura de comunicação baseada em mensagens ACL.

Inicialmente os jogadores enviam uma mensagem PROPOSE ao gestor para poderem se “registar” no jogo. Caso o limite de jogadores ainda não tenha sido alcançado o Gestor envia uma mensagem ACCEPT\_PROPOSAL, caso contrário envia REJECT\_PROPOSAL.

Após atingir o limite de jogadores o Gestor inicia o jogo com uma ronda nova, enviando para cada Jogador mensagens INFORM com o tabuleiro, as plantações para a ronda e informações sobre os outros jogadores (ex.: dinheiro). De seguida envia a cada Jogador uma mensagem REQUEST para obter as licitações para as plantações, o Jogador por sua vez envia uma mensagem INFORM com esse valor.

No momento em que todas as licitações são recebidas o Gestor ordena os jogadores com base na sua licitação e, por essa nova ordem, envia uma mensagem REQUEST a pedir a plantação e o local onde esta vai ser plantada, o Jogador ao receber esta mensagem envia uma mensagem INFORM com a informação necessária.

Este processo repete-se para a licitação e colocação dos canais de água.

Após a colocação do canal de água o Gestor verifica se o jogo já terminou, caso tenha terminado este faz os cálculos para determinar o vencedor, caso contrário inicia uma nova ronda.

## Conclusões

A realização deste projeto foi bastante benéfica para a nossa aprendizagem. Deparamo-nos com variados problemas relativos ao JADE principalmente a comunicação entre agentes de forma distribuída. Programar para sistemas distribuídos não é muito comum no nosso curso e, por essa razão foi necessário mais tempo para que o projeto completasse o seu objetivo.

Perante o trabalho efetuado pode-se concluir que os objetivos inicialmente propostos foram cumpridos.

## Melhoramentos

Como em qualquer projeto existem muitos melhoramentos possíveis que podem ser efetuados com mais tempo e estudo. Para o caso do nosso trabalho esses pontos a melhorar são:

- Melhorar a interface gráfica;
- Permitir a participação de um jogador humano;
- Melhorar ou aumentar o número de lógicas para os agentes;

## Recursos

- <http://paginas.fe.up.pt/~eol/AIAD/aiad1415.html>
- <http://boardgamegeek.com/boardgame/8125/santiago>
- <http://jade.tilab.com/>
- [http://en.wikipedia.org/wiki/Java\\_Agent\\_Development\\_Framework](http://en.wikipedia.org/wiki/Java_Agent_Development_Framework)
- <http://www.cs.uu.nl/docs/vakken/map/JADEProgramming-Tutorial-for-beginners.pdf>
- <http://paginas.fe.up.pt/~eol/SMA/20062007/jade.pdf>