



Universidade do Minho  
Escola de Engenharia

**Mestrado Integrado em Engenharia Informática**  
**Sistemas Inteligentes / Agentes Inteligentes**

---

# THERMOSENS

Pedro Cunha, a67677

Vasco Monteiro, a51772

**29 de Novembro de 2015**

---

---

# Índice

[Resumo](#)

[Abstract](#)

[Introdução](#)

[Motivação](#)

[Apresentação do caso de estudo](#)

[Contexto](#)

[Objetivos](#)

[Estrutura do Documento](#)

[Desenho e implementação da arquitetura do sistema](#)

[Desenho da arquitetura](#)

[Implementação do sistema](#)

[Agentes Desenvolvidos:](#)

[Comportamentos](#)

[Comportamentos do Coordenador](#)

[Comportamentos da Interface](#)

[Estudo do Agente Sensor](#)

[Protocolos de Comunicação](#)

[Interface<->Coordenador](#)

[Coordenador <-> Sensor](#)

[Validação de dados de leituras](#)

[Camada de apresentação](#)

[Instruções para instalação](#)

[Conclusões](#)

[Trabalhos Futuros](#)

---

## Resumo

O presente documento visa reportar os passos seguidos no decorrer da implementação do trabalho prático da Unidade Curricular de Agentes Inteligentes. Pretende-se o desenvolvimento de um sistema multi-agente de suporte à monitorização de vários sensores virtuais, simulados, de temperatura. Este projecto serve como introdução à ferramenta de desenvolvimento deste tipo de agentes, usada neste trabalho, JADE.

---

## **Abstract**

The following document intends to present a report of the implementation of the project created in the context of the Intelligent Agents course, by utilizing a multi-agent system developed by ourselves that supports the monitorization of various virtual sensors, simulating temperature values. This project is a start point to the learning process of the development tool for this kind of agents, JADE.

---

## **Introdução**

### **Motivação**

O desenvolvimento deste projeto foi impulsionado pelo desejo a criar uma plataforma que monitoriza as diversas leituras obtidas por sensores de temperatura. Isto cativou-nos devido às implicações práticas que isto tem em comum com a Unidade Curricular de Agentes Inteligentes que na qual este projeto se enquadra e, também, pelas aplicações práticas que este tipo de sistemas têm nas mais variadas áreas do mundo real.

### **Apresentação do caso de estudo**

Neste projeto é pretendida a concepção e desenvolvimento de um sistema multi-agente utilizada para a monitorização de vários sensores virtuais de temperatura. Adicionalmente foi fornecido o código necessário para a utilização de um sensor (a classe “Sensor”), sendo para isso necessário desenhar e implementar o protocolo de comunicação que vai sustentar a troca de informação, em forma de mensagens, entre os agentes do sistema. Como tal, é necessária a implementação de um coordenador que vai mediar a comunicação entre o utilizador e os sensores, e vice-versa. Um dos requisitos principais é o de que o número de sensores existentes possa ser variável, pelo que será necessária a utilização do agente DF para que o agente Coordenador possa descobrir os sensores existentes. O agente DF (Directory Facilitator) vai funcionar como as páginas amarelas do Coordenador, sempre que este precisar de consultar a lista atual de sensores disponíveis.

### **Contexto**

Um sistema multiagente passa pela integração das funcionalidades e competências de entidades computacionais independentes e que podem comunicar entre si, denominadas de agentes, para que seja possível a resolução de problemas mais complexos recorrendo à contribuição dos vários agentes. Uma das várias ferramentas que permite a implementação

---

deste tipo de sistema e que será utilizada no desenvolvimento do projeto será o JADE (Java Agent DEvelopment).

## Objetivos

Concluído este projecto, esperamos alcançar as seguintes metas:

- Aprofundamento do conhecimento adquirido sobre as normas FIPA e tipos de comportamentos que um agente pode possuir;
- Desenho e implementação de uma arquitetura aplicacional que permita a correta representação do problema;
- Implementação de um protocolo de comunicação que permita diminuir a carga sobre o meio de comunicação entre os agentes;
- Desenvolvimento de uma camada aplicacional que suporte o bom funcionamento e reaja corretamente ao comportamento dos agentes;
- Implementação de uma camada de apresentação intuitiva e apelativa ao utilizador, multi-plataforma e online que permita ao utilizador aceder à informação que a aplicação fornece em qualquer local do planeta, desde que tenha uma conexão à internet.

## Estrutura do Documento

Terminada esta secção, iremos então dar início à fase de desenvolvimento. Nela estarão presentes as seguintes secções:

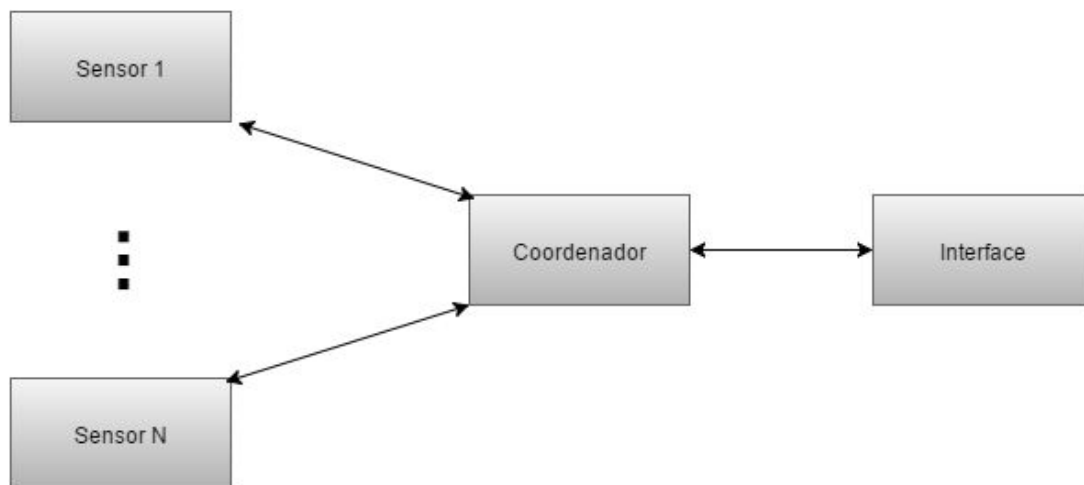
- Desenho e implementação da arquitetura da aplicação, onde serão enumerados e explicados os agentes e comportamentos a desenvolver;
- Estudo do agente *Sensor*, onde irão ser encontradas algumas das justificações necessárias à validação e representação dos valores lidos;
- Protocolos de comunicação, onde serão definidos os métodos de comunicação entre os vários agentes;
- Camada de apresentação, onde será apresentada brevemente a camada de utilização, bem como um manual de preparação para utilização do produto desenvolvido;
- Conclusões sobre o trabalho realizado e trabalhos futuros.

---

## Desenho e implementação da arquitetura do sistema

### Desenho da arquitetura

Em conformidade com os exercícios realizados no decorrer da unidade curricular o sistema a desenvolver deverá seguir a seguinte arquitetura:



O agente *Interface* vai ser a entidade principal do sistema, ele vai ser o agente que comunica com as necessidades e pedidos do utilizador, assim como os pedidos de alteração dos estados dos sensores, pedidos de temperaturas atuais, entre outros. O *Interface* vai guardar a informação de todas as leituras válidas e sobre o estado atual de cada sensor.

No entanto estes pedidos não vão diretos para os sensores e, como tal, foi criado um agente *Coordenador* que vai ter como sua função ser o mensageiro entre a *Interface* e os *Sensores*, bem como a entidade que, caso a *Interface* peça, vai procurar os sensores atuais no DF (como descrito anteriormente).

Os agentes *Sensor*, fornecidos para o projecto pelo corpo docente, são entidades bastante simples que têm dois estados, *online* e *offline*, e que, quando solicitado pelo *Coordenador*, respondem com um valor aleatório entre 0 e 100.

---

## Implementação do sistema

### Agentes Desenvolvidos:

Coordenador e Interface

### Comportamentos

Em JADE, a um agente podem estar associados um ou mais comportamentos. Estes comportamentos vão ditar a maneira como o agente funciona. Há vários tipos de comportamentos que caracterizam diferentes tipos de acções, como por exemplo os comportamentos cíclicos (que representam acções contínuas e repetitivas), comportamentos “one shot” (realizados uma só vez ou esporadicamente), entre outros.

### Comportamentos do *Coordenador*

O *Coordenador* vai ter um comportamento principal associado a ele e uma data de comportamentos secundários:

- *RecebePedidosInterfaceBehaviour*

Este é o comportamento principal, implementado como um *CyclicBehaviour* ele vai estar continuamente à escuta de pedidos da *Interface* e vai atuar de maneira diferente, isto é, chamar um *behaviour* secundário que satisfaça esse pedido.

- *PedeLeiturasBehaviour*

Este comportamento é chamado quando a *Interface* pede os valores atuais dos sensores, cada nome da lista de sensores passada pela *Interface* vai ser destinatário de uma mensagem com o conteúdo “value”, isto vai fazer com que cada *Sensor*, caso esteja online, responda com um valor, válido ou não. Estas respostas são depois reencaminhadas para a *Interface*. Este comportamento funciona como um *OneShotBehaviour*.



- 
- *ListaAgentesBehaviour*

Um *OneShotBehaviour* que vai ser chamado logo no arranque da aplicação e sempre que for solicitada, pelo utilizador, a inserção de um novo sensor no sistema. A sua função é fazer com que o *Coordenador* procure no DF a lista atual dos sensores do tipo “temperatura” e, de seguida, passe essa informação para a *Interface* para que esta possa verificar se houve alterações na lista de sensores.

- *PoeOnlineBehaviour/PoeOfflineBehaviour*

Ambos comportamentos *OneShot*, fazem o que o nome indica, enviam a uma lista passada pela *Interface* uma mensagem com os valores “online” ou “offline”, respectivamente.

### Comportamentos da *Interface*

A *Interface* é inicializada com um *SequentialBehaviour* composto por dois comportamentos, o primeiro um *OneShot*, que vai pedir a lista de agentes disponíveis ao coordenador, e o segundo um *ParallelBehaviour*. Este segundo, *InterfaceParaleloBehaviour*, é também ele composto por dois comportamentos, que vão correr paralelamente: um cíclico e outro temporizado.

- *InicialInterfaceBehaviour*

O tal comportamento sequencial supracitado composto por: *PedeListaAgentesBehaviour* e *InterfaceParaleloBehaviour*.

- *PedeListaAgentesBehaviour*

*OneShotBehaviour* que vai, quando chamado, mandar uma mensagem ao *Coordenador* e esperar que este responda com a lista de sensores encontrados no DF.

- *InterfaceParaleloBehaviour*

Como foi dito anteriormente, este comportamento é composto por dois comportamentos fundamentais para a *Interface*: *RecebeInfoDoCoordenadorBehaviour* e *PedeTemperaturasBehaviour*.

- 
- *RecebeInfoDoCoordenadorBehaviour*

Comportamento cíclico que vai escutar continuamente as respostas do *Coordenador* a pedidos da própria *Interface*. Há 6 respostas que podem vir do *Coordenador*: “sensores: <lista de nomes>”, “online:nome\_sensor”, “onlineLista:<lista\_nomes>”, “temperaturas: <lista nomes e leituras>”, “offline:nome\_sensor”, “offlineLista:<lista nomes>”. Cada uma destas respostas vai possibilitar a atualização do estado conhecido dos sensores e, no caso particular das temperaturas, as leituras válidas serão atualizadas no mapa de leituras do sensor respectivo.

- *PedeTemperaturasBehaviour*

*TickerBehaviour* que vai de X em X tempo enviar uma mensagem ao *Coordenador* a pedir as leituras atuais dos sensores.

- *PedeOnlineBehaviour/PedeOfflineBehaviour*

Ambos os comportamentos têm dois modos de funcionamento: caso o conteúdo da mensagem que segue para o *Coordenador* neste pedido seja somente “online” ou “offline”, todos os sensores, independentemente do seu estado atual vão receber uma ordem do *Coordenador* para se ligarem ou desligarem, respectivamente; caso a mensagem contenha online/offline seguido de uma lista (que pode conter só um elemento) de nomes de sensores então só os sensores contidos nessa lista é que receberão essa ordem do *Coordenador*.

---

## Estudo do Agente Sensor

Após inspeccionar o código do agente Sensor foi possível retirar as seguintes informações:

- Possui um estado indicativo de online ou offline. Por omissão um agente iniciado estará offline.
- Responde a mensagens com a performative “REQUEST”. Dentro destes pedidos são possíveis três pedidos:
  - ◆ online - Para colocar o estado de ligação do sensor a “Online”
  - ◆ offline - Para colocar o estado de ligação do sensor a “Offline”
  - ◆ value - Para pedir o valor “atual” da temperatura medido pelo sensor.
- Quando existe o método value é necessário que o sensor esteja “Online”. Como tal, antes de se realizar qualquer leitura é necessário colocá-lo online.
- O pedido “Value” envia, como resposta valores resultantes de um processo aleatório. Caso um valor gerado esteja entre 0 e 9 a resposta terá o conteúdo XXXXX, que deverá ser interpretado como erro. Caso o valor gerado esteja entre 10 e 19 será enviado o conteúdo com o valor negativo do número gerado, que deverá ser interpretado como leitura inválida. Entre 20 e 90 será enviado o valor gerado, como resultado de uma leitura correta. Caso esteja entre 91 e 100 estaremos num caso que como não vai responder, vai gerar um timeout.
- Dado o domínio dos valores aceitáveis das leituras dos sensores foi necessário encontrar uma escala de temperaturas em que o intervalo [20,90] seja aceitável no que toca a temperaturas em casas. Como tal interpretámos que os valores lidos pelos sensores estão na escala Fahrenheit (°F).

---

## Protocolos de Comunicação

### Interface<->Coordenador



Para a comunicação entre a Interface e o Coordenador e para permitir operações (ligar, desligar, pedir valores) a vários sensores ao mesmo tempo foram utilizadas mensagens com o seguinte formato:

operação:sensor1;restantes sensores necessários. (P.e.: temperaturas: Quarto;Cozinha;Sala)

Após o coordenador obter as respostas ao pedido realizado enviará uma resposta com o seguinte formato:

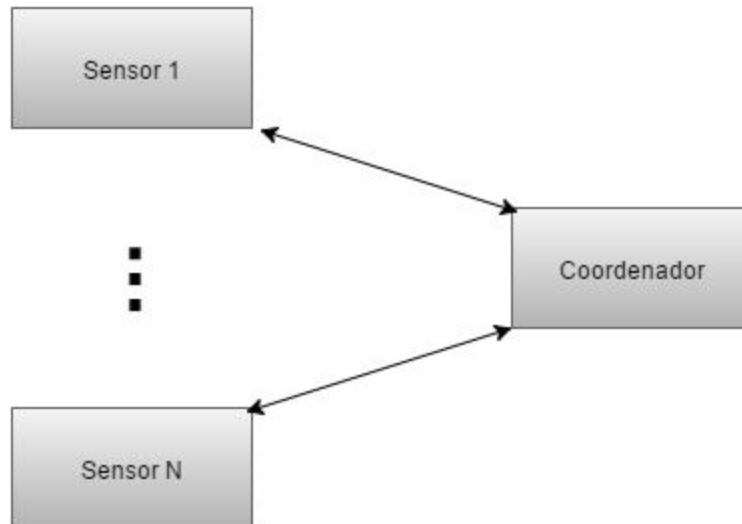
operacao:campoOp1;restantes. Existirá o caso especial para as leituras de temperaturas em que se esperará um par nome-temperatura (P.e.: temperatura:Quarto,45;Cozinha,XXXXX;Sala,-15). Os restantes ocorrerão como o primeiro formato apresentado.

Caso exista timeout em algum dos sensores o elemento que forma par com o nome local do sensor causador é "timedout".

Essencialmente a comunicação é feita à base de mensagens que contêm strings, strings essas que são tratadas no destino consoante a maneira como são formatadas pela origem, isto é, são feitas separações por ",", "." ou ";" que no destino depois são separadas devidamente para tratamento adequado.

---

## Coordenador <-> Sensor



Para este caso serão utilizadas mensagens simples com os receptores necessários (Coordenador -> Sensores) com o pedido. A resposta virá sob a forma de uma mensagem com a performative “INFORM” com o valor da leitura.

## Validação de dados de leituras

Apesar do intervalo de valores aceitáveis ser [20,90] °F assumiu-se que os valores próximos dos limites também se tratariam de leituras erradas. Como tal, decidiu-se que o intervalo aceitável para leituras de temperaturas seria de [41,86] °F ([5,30] °C). Leituras com valores numéricos negativos ou textuais serão descartados e considerar-se-á que ocorreu um erro na leitura. Como os sensores não são reais e simulam valores de temperaturas com base na geração de números aleatórios, este intervalo “aceitável” de 5°C até 30°C embora já seja mais plausível, continua a ser um intervalo bastante irreal. No entanto, se reduzissemos, ainda mais, esse intervalo o número de erros, que já não é pequeno, seria ainda maior.

---

## Camada de apresentação

Para facilitar a dinamicidade do programa desenvolvido bem como melhorar a aparência estética do produto desenvolvido ficou decidido que o mesmo iria ficar separado em duas partes:

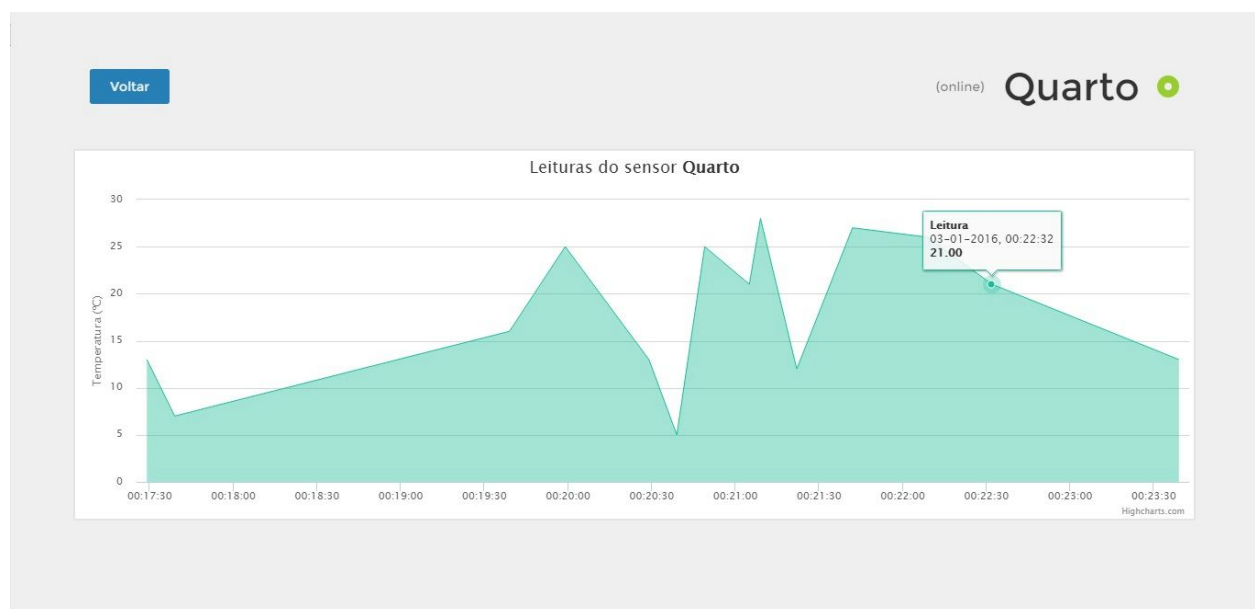
- Para a camada de apresentação, foram utilizadas as tecnologias HTML5, CSS, PHP e Javascript/ JQuery.
- Para a comunicação entre a camada de apresentação e a aplicação propriamente dita, uma vez que são dimensões completamente separadas e independentes, foi criado, com o auxílio da ferramenta Spark, um Java Web Server que vai permitir, através de pedidos AJAX (Javascript), aceder aos dados das bases de informação da aplicação JADE.

A interface é bastante simples, constituída pela zona central (a branco) onde temos um icon que nos indica que há situações de anomalia, isto é, há “zonas da casa” com temperaturas que fogem do normal observado (aqui o normal observado estabeleceu-se algo que fuja muito da temperatura média, tanto para valores mais baixos como mais altos); carregando nesse botão poder-se-ão ver os avisos que o sistema tem para o utilizador. Tem também a informação sobre a localização do serviço, bem como a temperatura e estados do tempo (aproximado) nessa localização; contem informação, se expandido, sobre a probabilidade de precipitação, se está sol, chuva, neve, se é de dia ou de noite, se o céu está nublado, entre outras características metereológicas.



Na zona central estão ainda os quadros dos sensores que são atualizados de X em X segundos, (idealmente esta atualização será feita ligeiramente mais devagar que a atualização dos pedidos do agente *Interface*), com a informação da temperatura atual da divisão ao qual o sensor diz respeito, bem como a temperatura média até então observada, o estado do sensor na última atualização realizada: online, offline, timed out, error ou desconhecido são os estados possíveis.

Carregando num desses campos, o utilizador será levado para a página diferente, onde será possível observar o gráfico que representa o histórico de medições do sensor escolhido:



---

## **Instruções para instalação**

As instruções para configurar a aplicação com o respectivo Java Web Server, assim como a configuração do servidor apache local que permite correr o front-end da aplicação com pedidos Ajax cross-domain, seguem em anexo e em inglês num ficheiro de texto chamado "README\_INSTRUCTIONS".



---

## Conclusões

Confrontando os resultados obtidos com os objetivos propostos ao início do programa podemos concluir que:

- A arquitetura de agentes desenvolvida tornou fácil a separação e utilização correta dos comportamentos colocados nos mesmos.
- Face ao modo como ficou decidido que o programa iria correr, o protocolo de comunicação desenvolvido foi um apoio importante para que não houvesse sobre-carregamentos nos receptores. Tal foi possível verificar através do funcionamento do programa.
- Com a camada de apresentação desenvolvida acreditamos que seja possível ao utilizador aceder aos dados das leituras de uma forma simples e intuitiva, ver, em qualquer momento os gráficos temporais das temperaturas numa divisão de sua casa.

## Trabalhos Futuros

De forma a aprimorar os resultados obtidos na realização deste projeto foram encontrados os seguintes pontos:

- ❖ Adaptação do design da aplicação para dispositivos móveis;
- ❖ Utilização de sensores reais, com o uso da biblioteca OSGi, permitiria colocar na casa de cada pessoa, um sistema real, com leituras reais, que possibilitaria ao utilizador a qualquer momento e num sítio com acesso à internet verificar o estado;
- ❖ Maior pormenor nos avisos apresentados pelo sistema, com possibilidade de chamada automática do corpo de bombeiros caso se verificasse uma temperatura demasiado elevada numa ou mais divisões da casa, a título de exemplo;
- ❖ Capacitar o sistema de sugerir um esquema de climatização para a casa baseado na previsão do estado do tempo do dia seguinte para a localização geográfica em questão;
- ❖ Capacitar o sistema de um sistema de adaptação de temperatura ambiente das divisões de transição da casa (como o hall, ou divisões com entradas/saídas para o exterior) para amenizar as diferenças, muitas vezes bruscas, de temperatura que se podem sentir ao entrar/sair de casa.