



Department of Artificial Intelligence E.T.S. Ingenieros Informáticos Universidad Politécnica de Madrid



Sistemas Inteligentes

Agentes JADE

Javier Bajo
Catedrático de Universidad
jbajo@fi.upm.es



Index



- 1. Introducción**
- 2. Estructura de la plataforma JADE**
- 3. Instalación de JADE**
- 4. Creación de Agentes**
- 5. Jade Add-Ons**
- 6. Ejemplos de Agentes JADE**



1. Introducción

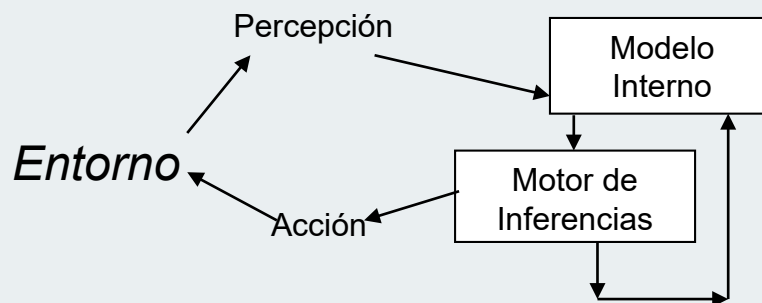
Sistema Inteligente



• Sistema Inteligente

Idea intuitiva: Sistemas y métodos que simulan aspectos del comportamiento inteligente, con la intención final de aprender de la naturaleza para poder diseñar y construir arquitecturas computacionales más potentes.

- Un sistema inteligente se basa en 2 conceptos principales:
 - **Representar el conocimiento.** El Sistema Inteligente utiliza un **Modelo Interno de representación** o **Base de Conocimiento**.
 - **Razonar.** El Sistema Inteligente utiliza un **motor de inferencias**.





1. Introducción

Sistema Inteligente



- **Sistema Inteligente**

Inteligencia Artificial Centralizada: Los conceptos de representación de conocimiento y de razonamiento se encuentran centralizados.

- Ejemplo: Toma de decisiones individuales

Inteligencia Artificial Distribuida: Los mecanismos de representación de conocimiento y de razonamiento se encuentran distribuidos. Aparecen las capacidades sociales, decisión colectiva,....

- Inspirados en sociedades animales
- Inspirados en comportamientos humanos y 2



1. Introducción

Agentes y Sistemas Multi-Agente

5/59

- **Agente**

- *Wooldridge [Wooldridge, 2002] proporciona una definición:*

"Un agente es un sistema computacional que está situado en algún entorno, y que es capaz de actuar de forma autónoma en este entorno para poder alcanzar sus objetivos de diseño"

- Existen otras definiciones para el término agente. Una de las más aceptadas es la que proporcionan Russel y Norving [1959]:

"La noción de un agente aparece como una herramienta para analizar sistemas, no una caracterización absoluta que divida el mundo en agentes y no agentes"

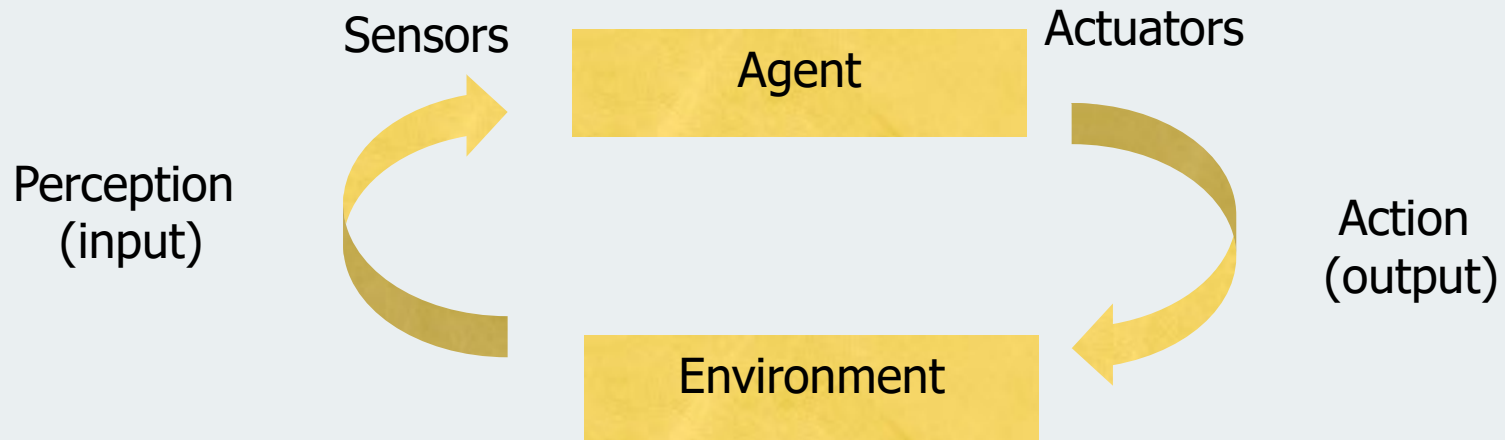


1. Introducción

Agentes y Sistemas Multi-Agente

6/59

- Un agente tiene una estructura con 4 elementos claramente definidos: habilidad de percepción, habilidad de actuar, objetivos y entorno.





1. Introducción

Agentes y Sistemas Multi-Agente

7/59

- **Agentes Inteligentes.**

Autonomy

Sociability

Mobility

Reactivity

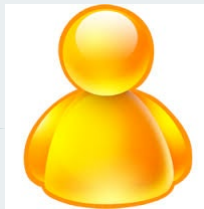
Pro-activity

Veracity

Rationality

Organization

Learning



Un agente es una entidad computacional que está situado en un entorno y que es capaz de actuar de forma autónoma para alcanzar sus objetivos (Wooldridge, 2002).



1. Introducción

Agentes y Sistemas Multi-Agente



- **Agente**

Un agente abarca una gran variedad de disciplinas de las ciencias de la computación, como son la Inteligencia Artificial, la Ingeniería del Software, las Bases de Datos, los Sistemas Distribuidos, pero, además, también abarca otras áreas de conocimiento como son la Psicología o la Sociología.

Normalmente para definir un agente, se hace a través de sus capacidades:

- **Autonomía:** Un agente actúa sin la necesidad de intervenciones externas (humanos u otros agentes).
- **Situación:** Los agentes se encuentran situados dentro de un entorno, ya sea real o virtual.
- **Reactividad:** Los agentes perciben el entorno y son capaces de actuar sobre el entorno.
- **Pro-Actividad o Racionalidad:** Los agentes no actúan simplemente en respuesta a su entorno, sino que son capaces de definir metas y planes para poder alcanzar sus objetivos.



1. Introducción

Agentes y Sistemas Multi-Agente



- **Agente**

- **Habilidad social:** Los agentes interactúan con otros agentes (y posiblemente también con humanos) por medio de alguna clase de lenguaje de comunicación.
- **Inteligencia:** Los agentes se rodean de conocimiento (creencias, deseos, intenciones y metas).
- **Organización:** Los agentes se organizan dentro de sociedades que siguen unas estructuras definidas en sociedades humanas o ecológicas.
- **Movilidad:** Los agentes son capaces de moverse de unos sitios a otros. El movimiento supone que tanto el código como los estados mentales de los agentes sean trasladados junto con el agente.
- **Aprendizaje:** El agente puede adaptarse progresivamente a cambios en entornos dinámicos mediante técnicas de aprendizaje.

- **Sistema Multi-Agente**

Cuando dos o más agentes son capaces de trabajar de forma conjunta con el objetivo de resolver un problema se habla de un sistema multiagente (Al menos uno de los agentes debe de ser autónomo y debe existir al menos una relación entre dos agentes en la que se cumpla que uno de los agentes satisface el objetivos del otro).



1. Introducción

Sistema Multiagente

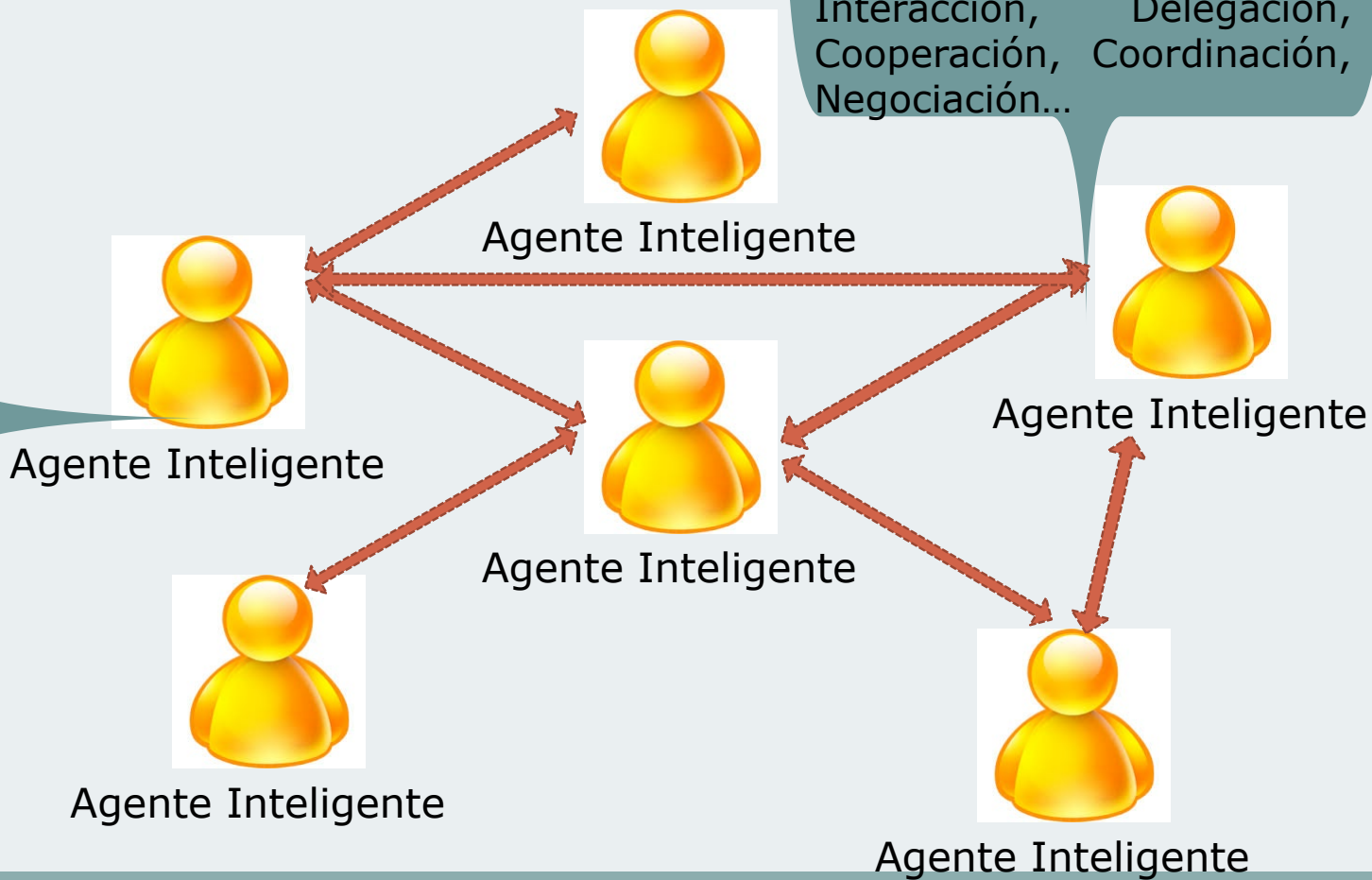
• Sistema Multiagente

Capacidades:

Autonomía
Sociabilidad
Movilidad
Reactividad
Pro-actividad
Veracidad
Racionalidad
Organización
Aprendizaje

Habilidades
Interacción,
Cooperación,
Negociación...

Sociales:
Delegación,
Coordinación,





1. Introducción

Agentes y Sistemas Multi-Agente



- **Plataformas**

Existen muchas plataformas para implementar agentes y sistemas multi-agente. En nuestro caso nos centraremos en JADE:



Se trata de:

- Una plataforma gratuita
- Se ejecuta sobre maquinas virtuales Java
- Ofrece una gran cantidad de funcionalidades
- Tiene una gran comunidad que la utiliza y mantiene.
- Tiene una versión Python (PADE - <https://pypi.org/project/pade/>)



Index



- 1. Introducción**
- 2. Estructura de la plataforma JADE**
- 3. Instalación de JADE**
- 4. Creación de Agentes**
- 5. Jade Add-Ons**
- 6. Ejemplos de Agentes JADE**



2. Estructura de la Plataforma JADE

Introducción



- **JADE (Java Agent Development Framework)** es un middleware que facilita el desarrollo de sistemas multiagente bajo el estándar FIPA (Foundation for Intelligent Physical Agents).
- **FIPA** (www.fipa.org) es un comité de estandarización IEEE que define una serie de especificaciones que tienen que seguir las plataformas de agentes:
 - *Agent Management specifications (FIPA no. 23),*
 - *Agent Communication Language ACL Message Structure (FIPA no. 61)*
- Desarrollado en Java.
- Facilita la comunicación entre agentes y permite la detección de servicios que se proporcionan en el sistema.





2. Estructura de la Plataforma JADE

Introducción



- **JADE incluye:**
 - **Entorno de ejecución** en el que los agentes de JADE se ejecutan.
 - **Bibliotecas de clases** para la creación de agentes mediante la herencia y la redefinición de comportamientos.
 - Conjunto de **herramientas gráficas** para la monitorización y administración de la plataforma de agentes.



2. Estructura de la Plataforma JADE Plataforma



- **Características principales de JADE:**
 - Plataforma de agentes distribuida – Permite implementar **sistemas inteligentes distribuidos**
 - GUI – proporciona un **interfaz gráfico**
 - Un **contenedor** por cada host en el que se están ejecutando agentes
 - Un **contenedor principal** que engloba a todos los contenedores
 - **Herramientas de depuración**
 - **Movilidad** de agentes de código y contenido
 - Ejecución paralela de **comportamiento de los agentes**
 - Estándar FIPA
 - ✦ **AMS (Agent Management System)**
 - ✦ **DF (Directory Facilitator)**
 - **Registro** automático de los agentes en el AMS
 - Soporte para la definición de **lenguajes y ontologías**



2. Estructura de la Plataforma JADE Plataforma

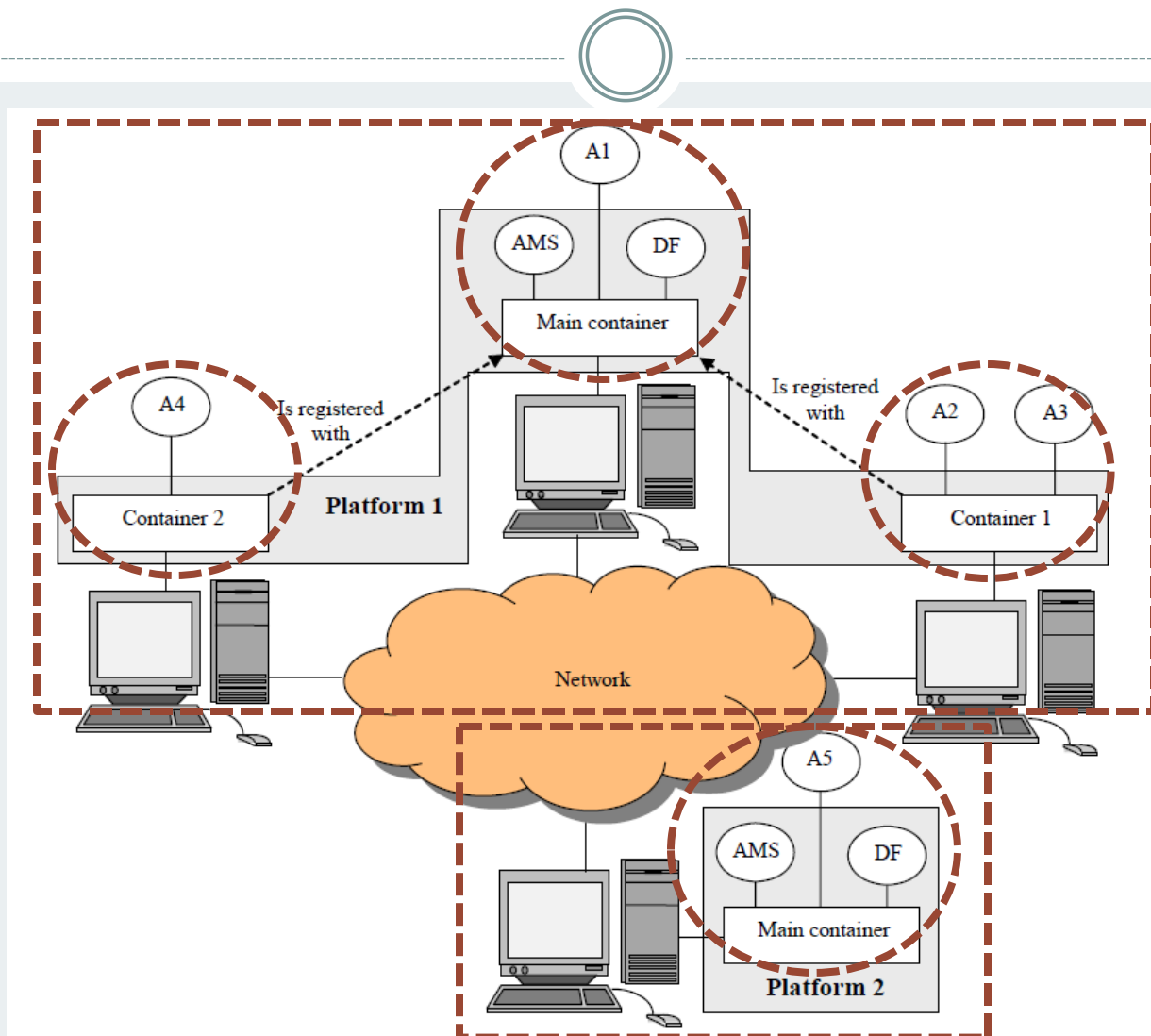


- **Componentes JADE:**

- **Agente.** Entidades autónomas que pueden incorporar cierto grado de inteligencia y que se identifican de forma única (ej: A1, A2, ...).
- **Plataforma.** Proporciona a los agentes servicios básicos, tales como el intercambio de mensajes. JADE es una plataforma que sigue los estándares FIPA para la ejecución de agentes, y que puede ejecutarse sobre una o más máquinas virtuales java. Está formada por contenedores.
- **Contenedor.** Una plataforma se compone de uno o más contenedores. Cada contenedor contiene cero o más agentes. Es una instancia del entorno de ejecución, en la que los agentes pueden ejecutarse e intercambiar mensajes.
- **Contenedor principal.** Es un contenedor especial que contiene agentes, pero que:
 - ✦ Es el primero que se crea y el resto de contenedores hacen referencia a él cuando se crean.
 - ✦ Incorpora dos tipos especiales de agentes:
 - **AMS.** Agente encargado de la gestión del sistema. Tiene la autoridad en la plataforma para realizar acciones tales como por ejemplo la creación o la eliminación de agentes. Gestiona un servicio de nombres, asegurando que cada agente en la plataforma tenga un nombre único.
 - **DF.** Agente encargado de realizar labores de directorio de servicios. Ofrece un servicio de páginas amarillas.
- **MTS.** Sistema de transporte de mensajes. La comunicación entre agentes se realiza mediante paso de mensajes. El ACC (Agent Communication Channel) es el software encargado de gestionar el paso de mensajes.

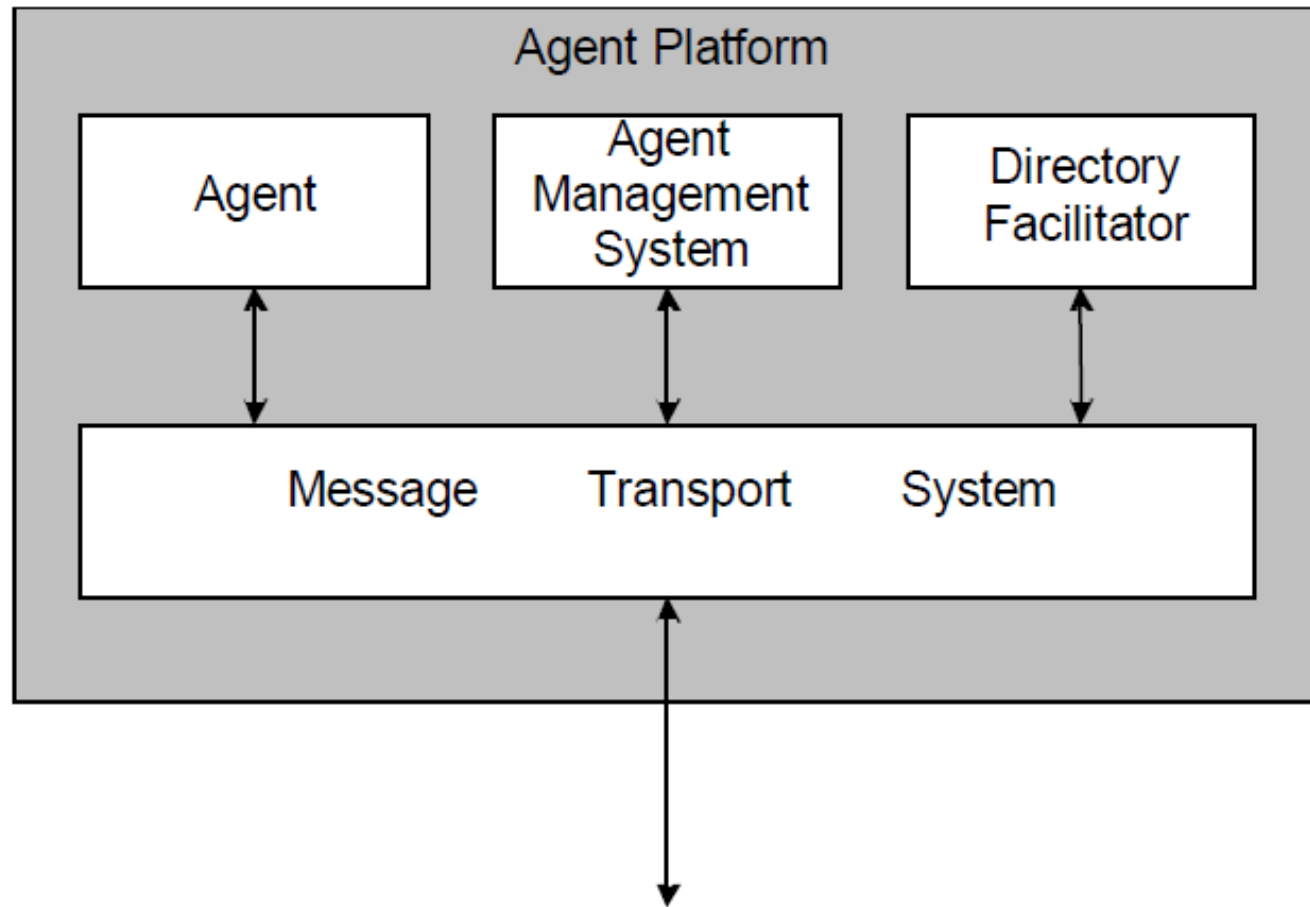


2. Estructura de la Plataforma JADE Plataforma





2. Estructura de la Plataforma JADE Plataforma



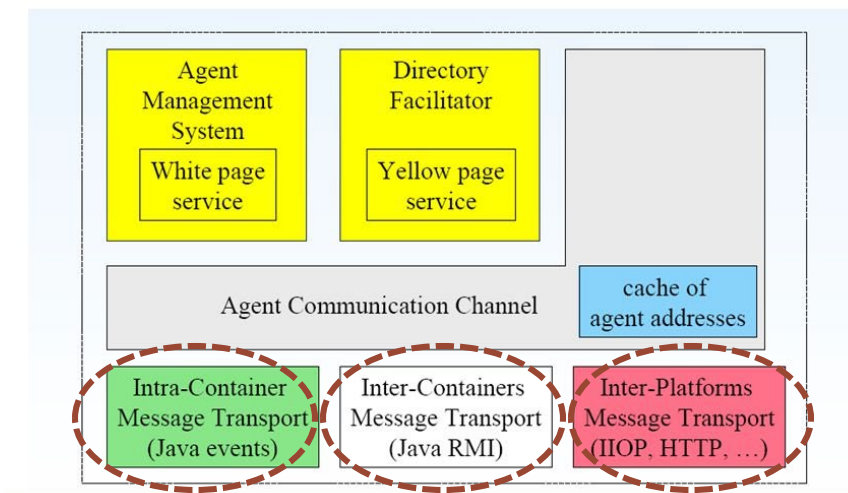


2. Estructura de la Plataforma JADE Introducción



- La plataforma proporciona un **servicio de transporte de mensajes** que tiene tres variantes:
 - Agentes situados en un mismo contenedor: Java Events
 - Agentes situados en 2 contenedores distintos: Java RMI
 - Agentes situados en 2 plataformas distintas: IIOP, HTTP, ...

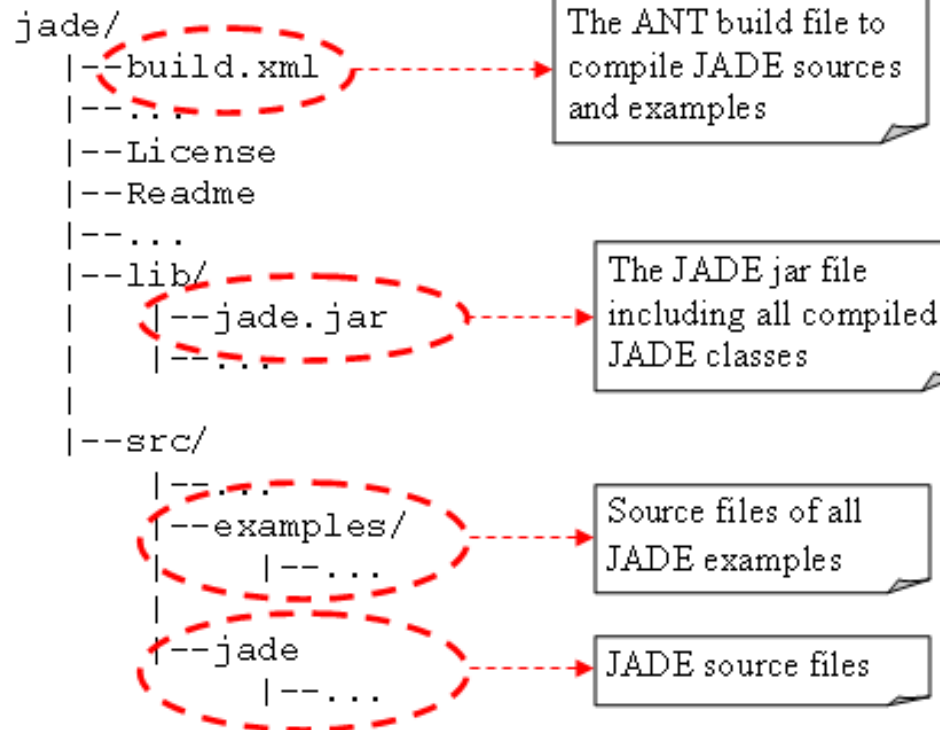
Internal architecture of the JADE run-time





2. Estructura de la Plataforma JADE Paquetes

- La organización de paquetes JADE es la siguiente:





2. Estructura de la Plataforma JADE Paquetes



De cara a la programación de sistemas multi-agente en JADE, debemos tener en cuenta los siguientes paquetes:

- **jade.core**: kernel de la plataforma
 - jade.core.behaviours: para redefinir los **comportamientos**
- **jade.lang.acl**: paquete para la **comunicación** según el estándar FIPA.
- **jade.content**: para la **definición de ontologías y lenguajes**. Se definen codificadores y decodificadores para los mensajes según el lenguaje por ejemplo jade.content.lang.sl.
- **jade.domain**
 - clases que **definen los agentes** en Agent Management specifications (FIPA no. 23). Ejemplo AMS (ciclo de vida de los agentes), DF (páginas blancas y amarillas).
 - Clases para la **interacción entre las herramientas de depuración y el kernel**.



2. Estructura de la Plataforma JADE Paquetes



- **jade.gui**: clases para la creación de **GUI** que permiten la edición de características de los agentes o envío de mensajes.
- **jade.mtp**: interfaces que deben cumplir **Message Transport Protocol**
- **jade.proto**: define los **protocolos estándar de comunicación** definidos por FIPA *fipa-request*, *fipa-query*, *fipa-contract-net*, *fipa-subscribe*.
- **jade.wrapper**: manejo de JADE como biblioteca de alto nivel.



2. Estructura de la Plataforma JADE Paquetes



- **Jade.tools:**

- **Remote Management Agent (RMA)** interfaz gráfico para el manejo y control de agentes. Permite la gestión, monitorización y control del estado de los agentes de la plataforma.



- **Dummy Agent:** herramienta de depuración para consultar el intercambio de mensajes y realizar envíos. Permite la gestión de mensajes ACL (inspección, creación, envío, modificación, recepción).



- **Sniffer Agent** : muestra gráficamente el intercambio de mensajes en tiempo de ejecución.

- **DF GUI:** interfaz gráfico para gestionar las páginas amarillas de los servicios y agentes. Permite al usuario interactuar con el DF (de/registrar agentes, modificar, etc.)



- **Introspector Agent:** Permite monitorizar el ciclo de vida de un agente y los mensajes que intercambia.



- **LogManagerAgent:** gestiona la información de depuración.

- **SocketProxyAgent:** agente que permiten la interconexión de la plataforma con socket TCP/IP, codifica y decodifica los mensajes a texto para entrada y salida.



2. Estructura de la Plataforma JADE Agentes DF y AMS



- Para acceder al agente DF se usa la clase **jade.domain.DFService** y los métodos estáticos
 - register
 - deregister
 - modify
 - Search
- Para acceder a la información del AMS se utiliza AMSService
 - Cuando se crea un agente se ejecuta automáticamente el **método register()** del AMS por defecto antes de ejecutar el **método setup()** del nuevo agente
 - Cuando se destruye un agente se ejecuta su **método takedown()** y automáticamente se llama a **deregister()** del AMS por defecto.



Index



- 1. Introducción**
- 2. Estructura de la plataforma JADE**
- 3. Instalación de JADE**
- 4. Creación de Agentes**
- 5. Jade Add-Ons**
- 6. Ejemplos de Agentes JADE**



3. Instalación de JADE Ejecución



- En primer lugar tendremos que decidir el **entorno de ejecución**. Hay dos opciones para la **Ejecución** de jade en nuestro equipo:

1. Utilizar ficheros de ejecución (Tal y como aparece en la carpeta JADE)

Configurar el CLASSPATH:

JADE_HOME = c:\JADE (suponemos que hemos copiado JADE en la carpeta c:\JADE)

JADE_LIB = %JADE_HOME%\lib

JADE_CLASSPATH =

%JADE_LIB%*.jar;%JADE_LIB%\crimson.jar;%JADE_LIB%\base64.jar;%JADE_LIB%\http.jar;%JADE_LIB%\iiop.jar;%JADE_LIB%\jade.jar;%JADE_LIB%\jadeTools.jar

Ejecutamos:

```
java -cp %JADECLASSPATH% jade.Boot
```

```
java -cp %JADECLASSPATH% jade.Boot -gui
```

- #### 2. Utilizar un IDE como Eclipse o Netbeans → Se trata del caso más habitual y es el que vamos a utilizar en clase, se muestra en las siguientes transparencias.



3. Instalación de JADE

Descarga



- Web oficial <https://jade.tilab.com/>
- Descargar JADE:

[Home](#) [Documentation](#) [Support](#) [Download](#) [Developers](#) [News](#) [Related Projects](#)

Jade Download

From this page you can now download JADE.
We suggest to unzip the files by using the 'jar xvf' command rather than the winzip application because some incompatibilities have been reported in the past.

Current version of JADE is 4.5.0 (08/06/2017).

Look at the [ChangeLog](#) for the list of improvements with respect to the previous versions.

Note: All the binaries (lib/ jar) have been generated by using the JDK1.6 compiler.*

JADE	~ File size	Description of the content
jadeAll	17.7 MB	This file contains all JADE, i.e. it is just composed of the 4 files below. If it is too large for downloading, the 4 files below might be downloaded instead.
jadeBin	2.5 MB	This file contains JADE already compiled and ready to be used, i.e. a set of JAVA archive JAR files.
jadeDoc	12.8 MB	This file contains all the JADE documentation included the Administrator's Guide and and the Programmer's Guide. NOTICE THAT all the documentation is also available on-line.
jadeSrc	2.3 MB	This file contains all the JADE source code.



3. Instalación de JADE

Descarga



- Web oficial <http://jade.tilab.com/>
- Add-Ons:

[Home](#) [Documentation](#) [Support](#) [Download](#) [Developers](#) [News](#) [Related Projects](#)

Add-Ons

This page contains all **add-ons** for JADE. These are additional modules implementing features that are not required to run JADE, but which provide interesting new pieces of functionality.

On this additional page you can find **3rd party software**, which can be used with JADE.

How to contribute?

Contributions from the JADE community are always welcome and the entire JADE Team tries to make its best effort to integrate them into the JADE main stream. Those users wishing to contribute code are invited to post an e-mail to jade-develop@avalon.tilab.com

Add-ons have a uniform style that simplifies users in using, compiling, and accessing their documentation. In particular, an add-on is required to have the following structure of sub-directories:

- **src** (this is the directory that contains all the source files of the add-on)
- **doc** (this is the directory where the API documentation of the add-on must be created by the ant build process)
- **lib** (this is the directory where the .jar file of the add-on must be created by the ant build process)
- **classes** (this is the directory where the .class files of the add-on must be created by the ant build process)

Furthermore, each add-on must include a file called **build.xml** which is an ant build file with the following targets:

- **compile** (default target) to create all the .class files
- **lib** to create the .jar file of this add-on



3. Instalación de JADE

Descarga



- Web oficial <http://jade.tilab.com/>
- Documentación:

The screenshot shows the JADE website interface. At the top is the JADE logo, followed by a navigation bar with links: Home, Documentation, Support, Download, Developers, News, and Related Projects. Below the navigation bar is the main content area titled "Tutorials & Guides" in green. Under this title, there are two sections: "Basic Documentation" and "Additional Tutorials".

Tutorials & Guides

Basic Documentation

- [Introduction to JADE](#)
- [JADE Administration Tutorial](#) by Ryerson University
- [JADE Programming Tutorial](#)
- [JADE administrator's Guide](#)
- [JADE Programmer's Guide](#)
- [API Documentation](#)

Additional Tutorials

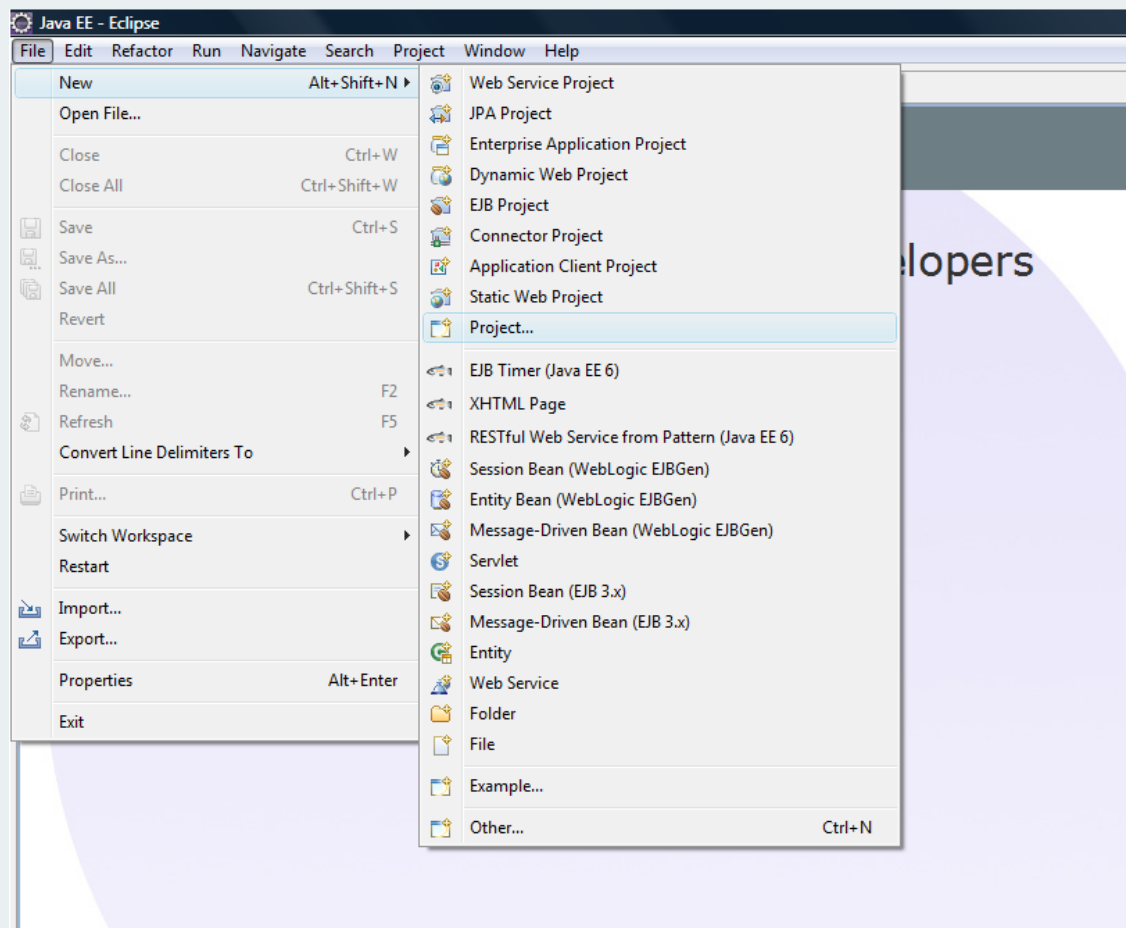
- **Standards**
 - [FIPA specifications](#)
- **Methodology**
 - [A Methodology for the Analysis and Design of Multi Agent Systems using JADE](#)
By Magid Nikraz – Murdoch University
- **Configuration**
 - [Using arguments and properties to configure your agents](#)
By Dick Cowan – Menehune Software Inc
- **Content Languages & Ontologies**
 - [Creating and using applications-specific ontologies](#)



3. Instalación de JADE Configuración



- Descarga Eclipse: <https://eclipse.org/downloads/>

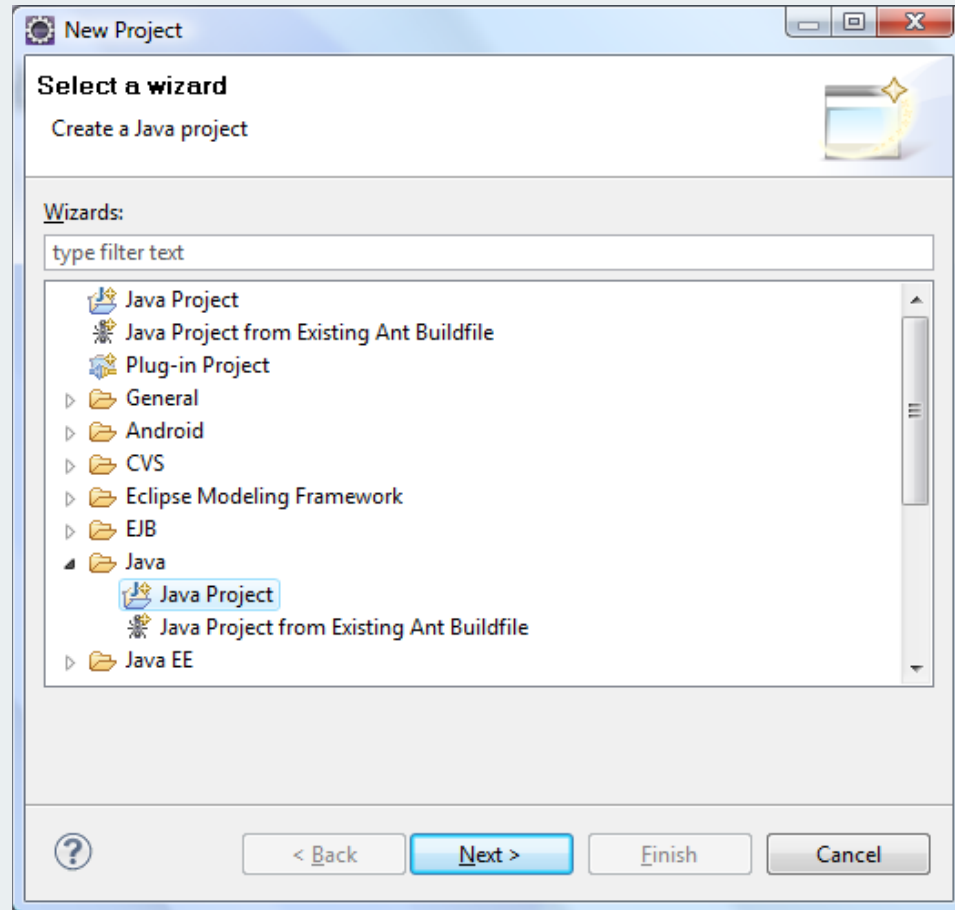




3. Instalación de JADE Configuración



- Creación de un nuevo proyecto en Eclipse:





3. Instalación de JADE Configuración



New Java Project

Create a Java Project
Create a Java project in the workspace or in an external location.

Project name:

☒ Use default location

Location: [Browse...](#)

JRE

☒ Use an execution environment JRE:

☐ Use a project specific JRE:

☐ Use default JRE (currently 'jdk1.6.0_18') [Configure JREs...](#)

Project layout

☐ Use project folder as root for sources and class files

☒ Create separate folders for sources and class files [Configure default...](#)

Working sets

☐ Add project to working sets

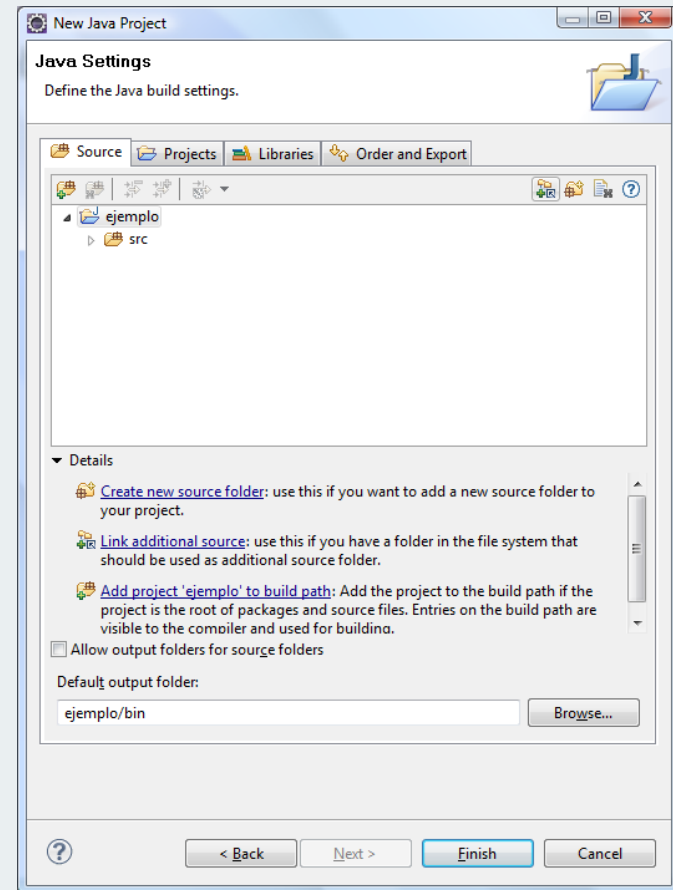
Working sets: [Select...](#)

[?](#) [< Back](#) [Next >](#) [Finish](#) [Cancel](#)



3. Instalación de JADE Configuración

- Vamos a las propiedades del proyecto (botón derecho → propiedades)
- Seleccionamos Java Build Path
- Ahora vamos a incorporar JADE

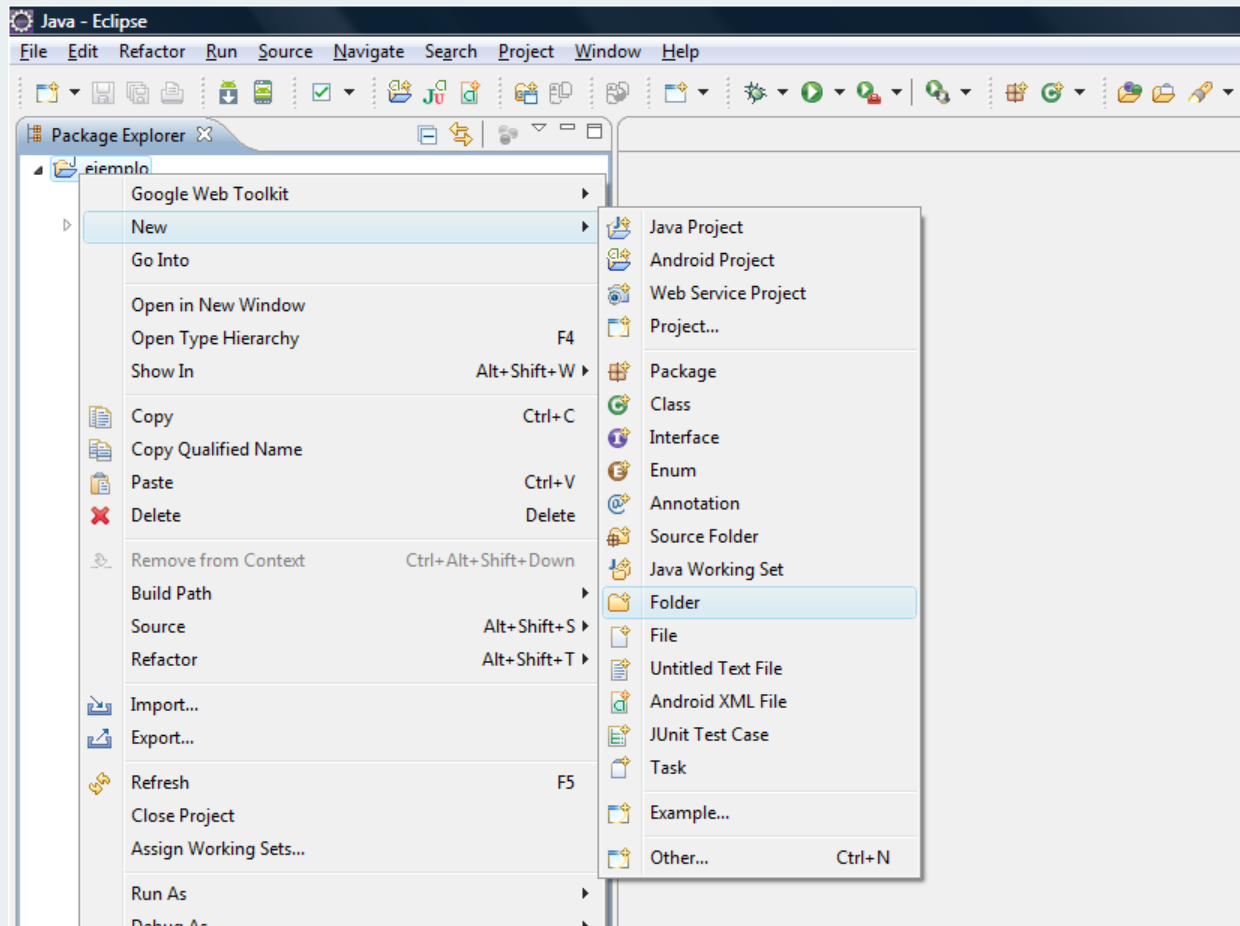




3. Instalación de JADE Configuración

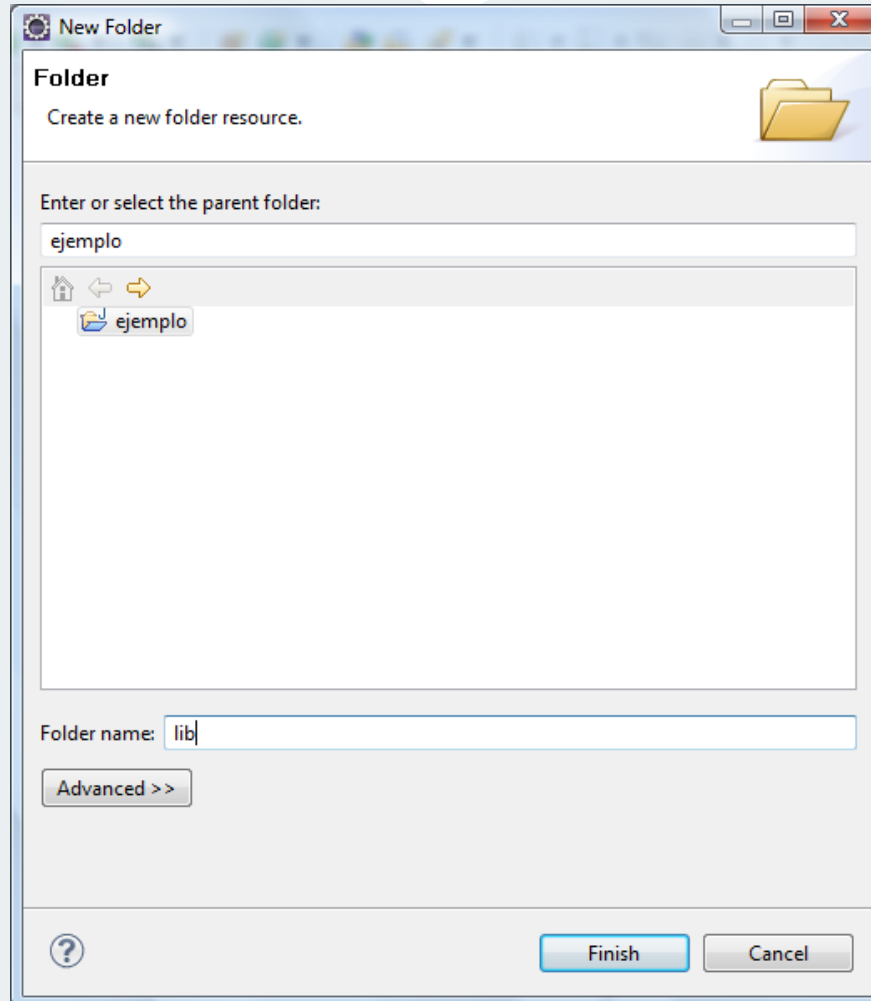


- Inclusión de bibliotecas Jade:





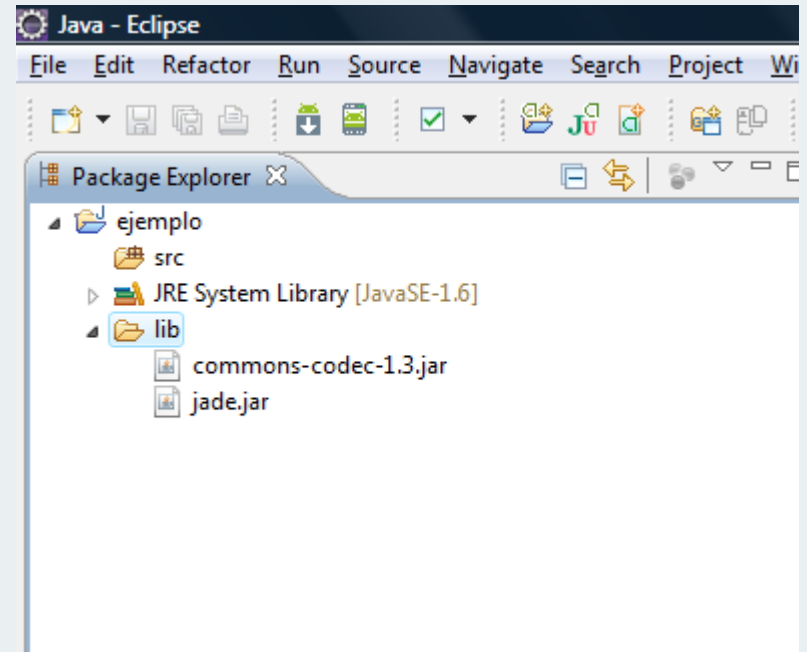
3. Instalación de JADE Configuración





3. Instalación de JADE Configuración

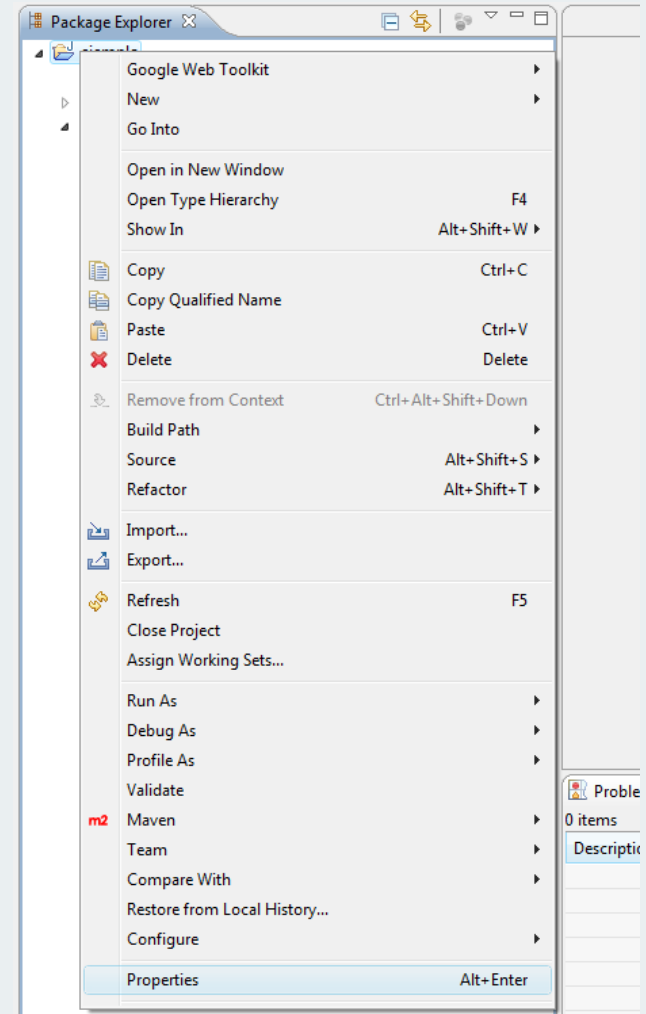
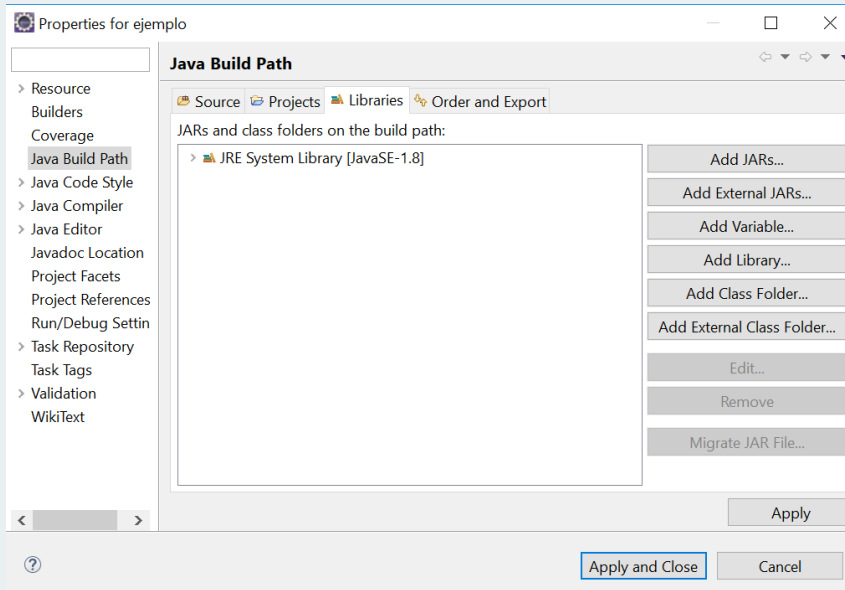
- Dentro de la carpeta *lib* copiamos los ficheros .jar que hay en la carpeta JADE que hemos descargado. Más concretamente, vamos a copiar:
 - *jade.jar*
 - *commons-codec.jar*
- Posteriormente es necesario incluir en el classpath los ficheros .jar que hemos copiado en la carpeta lib. Para ello tenemos que incluirlos en el classpath de nuestro proyecto siguiendo las indicaciones que se muestran en la siguiente transparencia.





3. Instalación de JADE Configuración

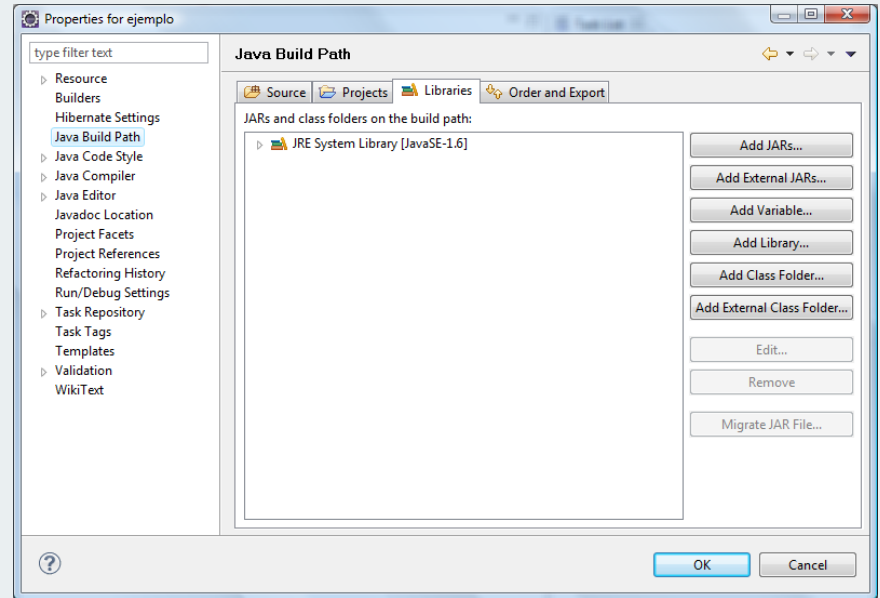
- Botón derecho del ratón sobre el proyecto y abrimos sus propiedades.
- Nos situaremos en la pestaña libraries y añadiremos los dos ficheros *.jar*.





3. Instalación de JADE Configuración

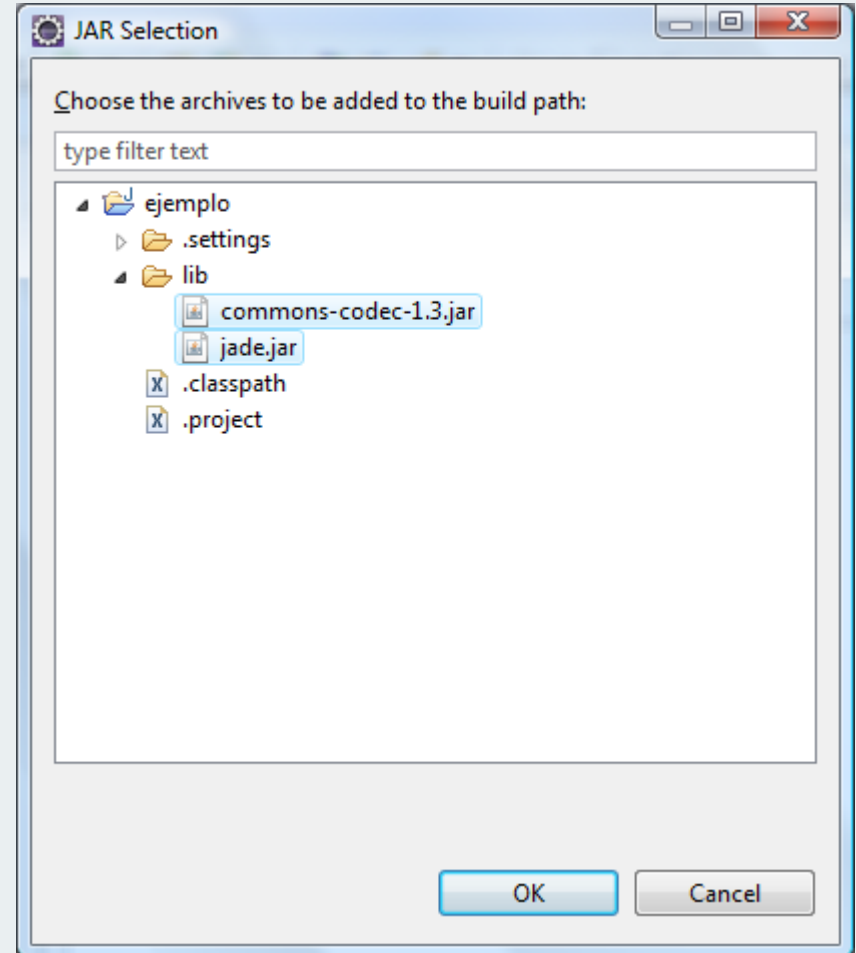
- Para añadir elementos nuevos, es necesario pulsar sobre *add jars* y seleccionar los ficheros .jar que se han copiado anteriormente a la carpeta lib.





3. Instalación de JADE Configuración

- Seleccionamos las 2 bibliotecas y las añadimos.
- Pulsamos en *Apply*.
- Pulsamos en *Ok*.
- Ya tenemos añadido JADE en el classpath de nuestro proyecto y podemos ejecutar JADE como parte del proyecto.

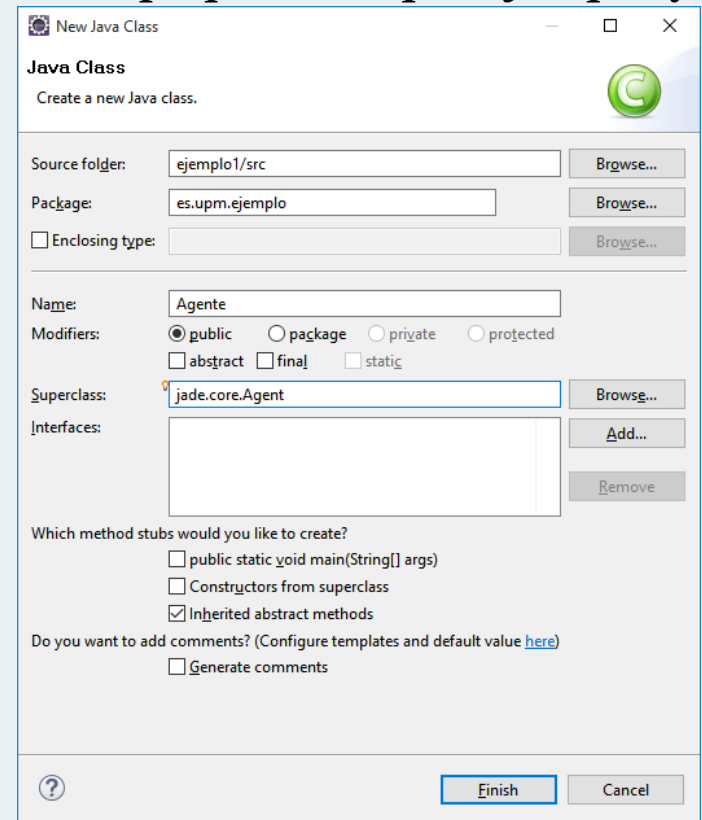
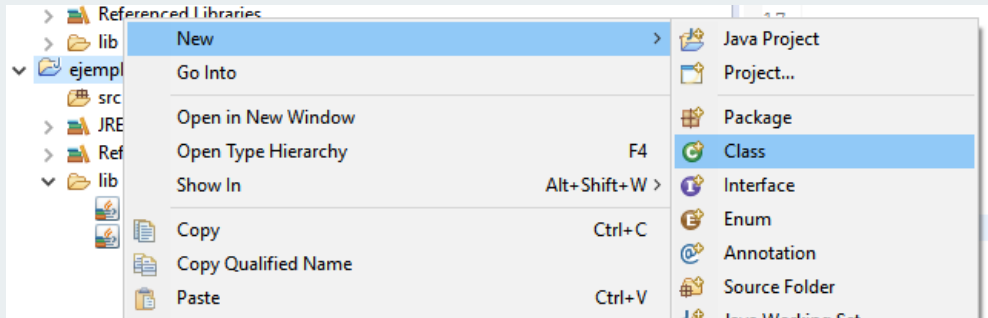




3. Instalación de JADE Configuración



- Ahora ya es posible **crear el primer agente**. Solamente es necesario crear una clase que herede de Agent pulsando con el botón derecho sobre el proyecto.
- Creamos un nuevo agente *Agente*, lo incluimos en el paquete *es.upm.ejemplo* y hereda de *jade.core.Agent*





3. Instalación de JADE Configuración

- Vamos a crear un agente llamado *Agente* que ejecuta un comportamiento cíclico (está ejecutándose de forma infinita) y que en el comportamiento simplemente se bloquea, es decir no hace nada.

```
package es.upm.ejemplo;

import jade.core.Agent;
import jade.core.behaviours.CyclicBehaviour;

public class Agente extends Agent {

    protected CyclicBehaviour cyclicBehaviour;

    public void setup()
    {
        System.out.println("Soy el Agente 1");
        cyclicBehaviour = new CyclicBehaviour(this)
        {
            public void action()
            {
                block();
            }
        };

        addBehaviour(cyclicBehaviour);
    }
}
```

Un agente extiende la clase Agent de JADE

Cuando se crea un agente se invoca el método setup()
Cuando se destruye un agente se invoca el método takedown()

Un agente puede ejecutar comportamientos

Es posible bloquear un agente en un comportamiento con block()



3. Instalación de JADE Configuración



- **Sintaxis de ejecución.** Para ejecutar el agente que hemos creado tenemos que solicitar su ejecución en la plataforma JADE:

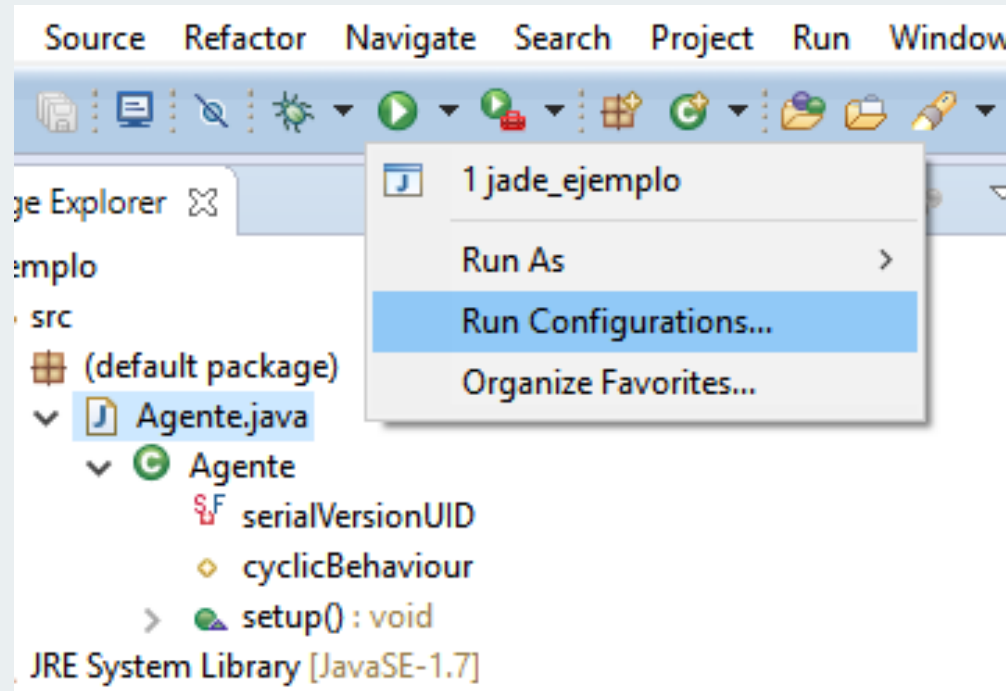
java jade.Boot [- opción] [lista agentes]

- Las opciones disponibles son:

- ✦ -container (contenedor en el que crear los agentes)
- ✦ -host (nombre del host en el que registrar los agentes)
- ✦ -port (puerto de conexión)
- ✦ -name (nombre de la plataforma)
- ✦ -gui (lanza el RMA, que es el interfaz gráfico de JADE)
- ✦ -mtp (especifica el protocolo de transporte de mensajes)
- ✦ -container-name (nombre del contenedor)
- ✦ -local-host (host-local para el contenedor)
- ✦ -local-port (puerto local para el contenedor)
- ✦ -conf (si se utiliza un fichero de configuraciones. En caso de no indicar el fichero se introduce la configuración a través de un interfaz)



3. Instalación de JADE Configuración

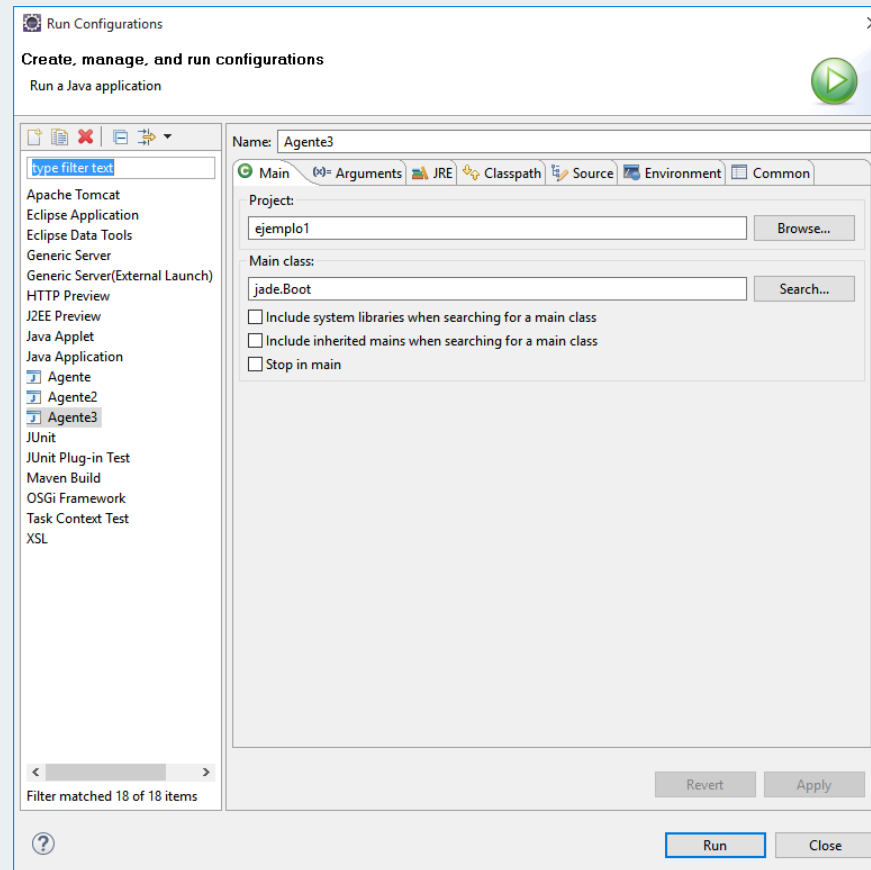




3. Instalación de JADE Configuración



- Seleccionar Java Application y pulsar sobre la opción de nuevo. Indicar la siguiente configuración en la pestaña Main

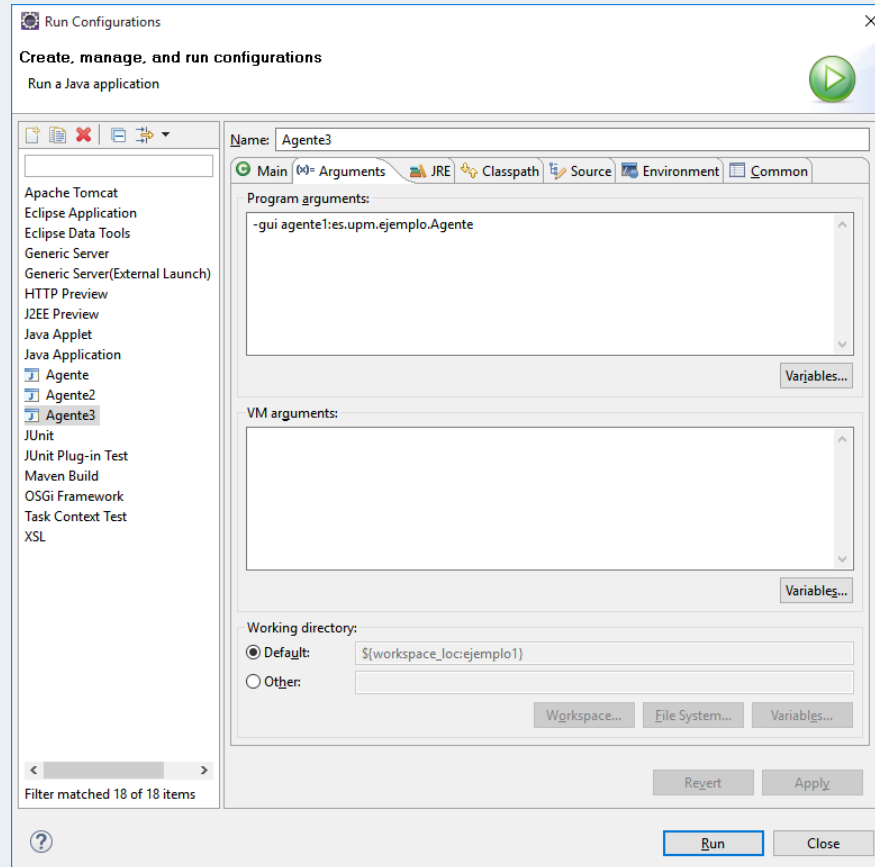




3. Instalación de JADE Configuración



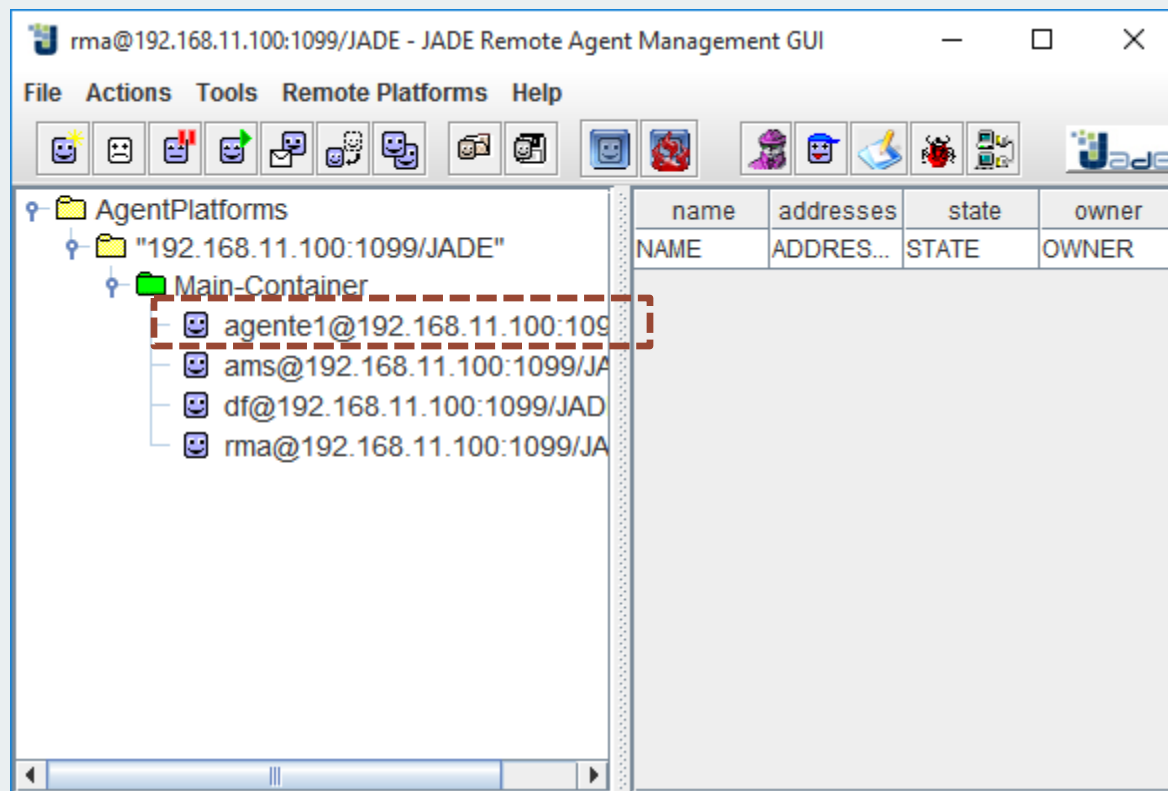
- Establecer los argumentos (opciones y lista de agentes) para iniciar el agente y el contenedor principal





3. Instalación de JADE Configuración

- El resultado final tras ejecutar el comando es este:





3. Instalación de JADE Configuración

- Podemos observar que:
 - El interfaz gráfico es en realidad un agente llamado RMA.
 - Existe un agente AMS que gestiona el sistema.
 - Existe un agente DF que ofrece un servicio de directorio (páginas amarillas)

The screenshot shows the JADE Remote Agent Management GUI window titled "rma@192.168.11.100:1099/JADE - JADE Remote Agent Management GUI". The window has a menu bar (File, Actions, Tools, Remote Platforms, Help) and a toolbar with various icons. The main area is divided into two panes. The left pane shows a tree view of AgentPlatforms:

- AgentPlatforms
 - "192.168.11.100:1099/JADE"
 - Main-Container
 - agente1@192.168.11.100:1099
 - ams@192.168.11.100:1099/JA
 - df@192.168.11.100:1099/JAD
 - rma@192.168.11.100:1099/JA

The right pane shows a table with columns: name, addresses, state, owner. The table has a header row with these columns and a body row with the same headers in all caps. The body row is empty.

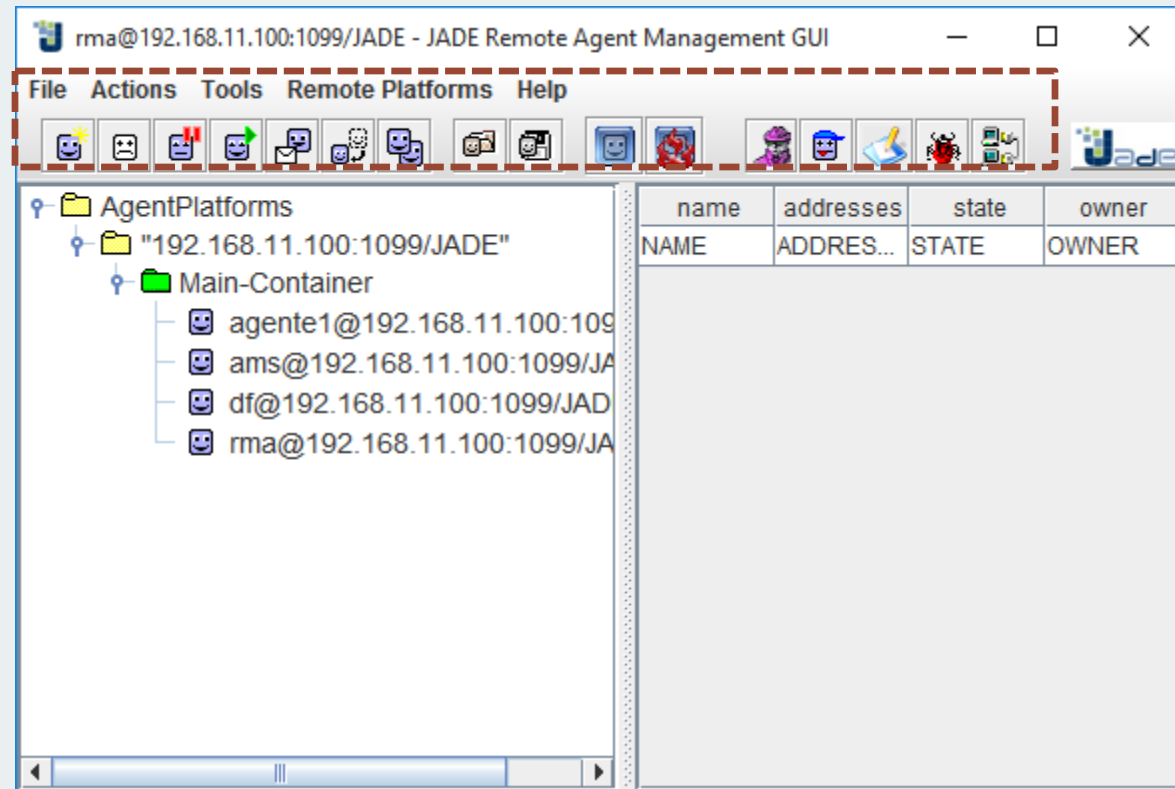
Below the screenshot is a diagram of the system architecture. It shows a central "Agent Communication Channel" box. Above it are two yellow boxes: "Agent Management System" (containing "White page service") and "Directory Facilitator" (containing "Yellow page service"). To the right of the "Agent Communication Channel" is a blue box labeled "cache of agent addresses". Below the "Agent Communication Channel" are three boxes: "Intra-Container Message Transport (Java events)" (green), "Inter-Containers Message Transport (Java RMI)" (white), and "Inter-Platforms Message Transport (IIOP, HTTP, ...)" (pink).



3. Instalación de JADE Configuración



- Podemos observar que:
 - Desde el menú, o bien desde los iconos de acceso rápido del agente RMA podemos ejecutar acciones sobre los agentes.
 - También podemos acceder a agentes especiales como son:
 - ✦ Sniffer agent
 - ✦ Dummy agent
 - ✦ Log Manager agent
 - ✦ Introspector agent



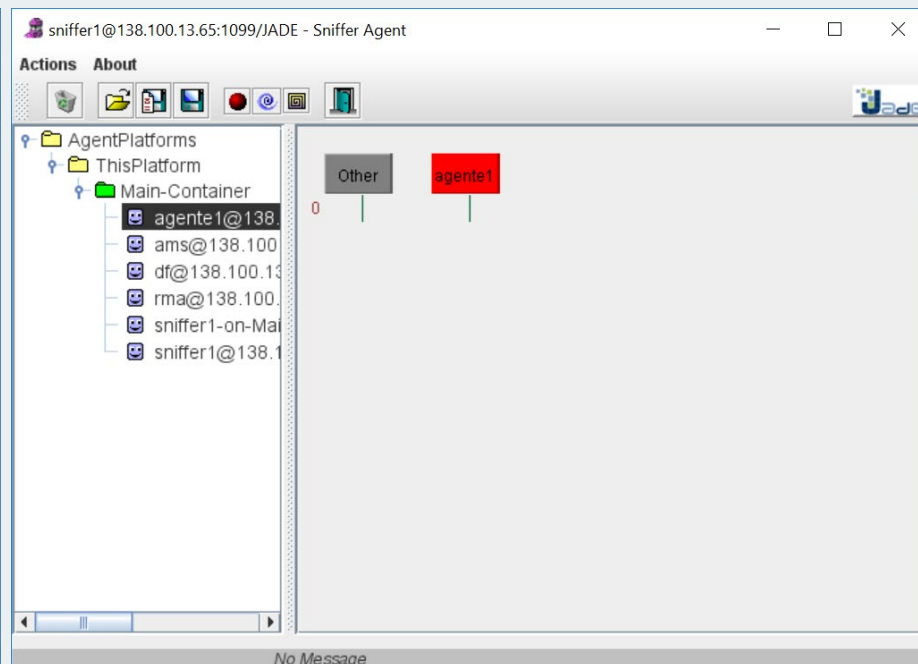
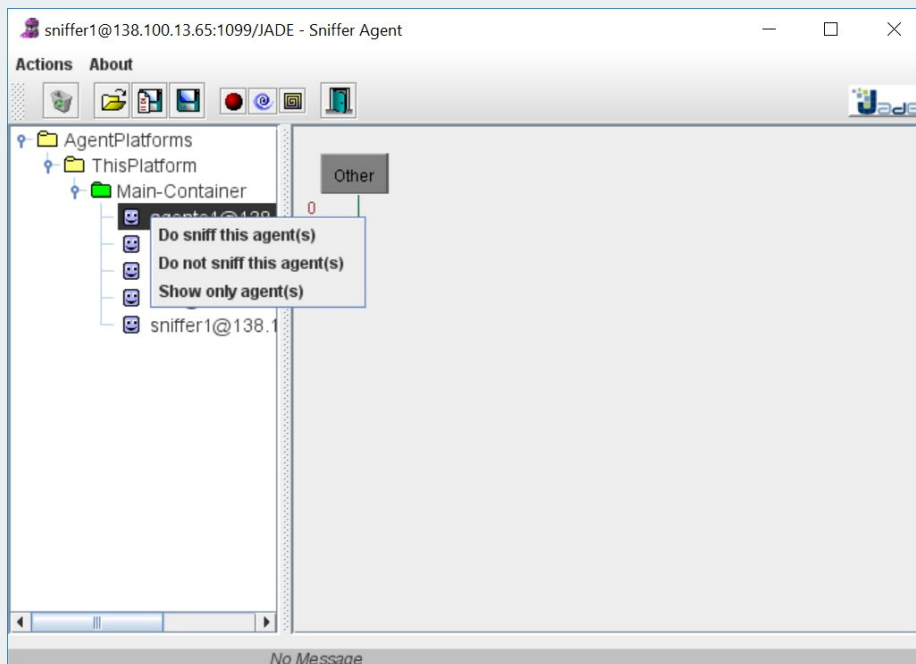


3. Instalación de JADE Configuración

- Podemos observar que:



✧ **Sniffer agent.** Se trata de un agente que permite observar todos los mensajes intercambiados en la plataforma para los agentes seleccionados:



- ✧ El agente sniffer proporciona una herramienta muy interesante a la hora de analizar interacciones sociales.



3. Instalación de JADE Configuración



- Podemos observar que:

✧ **Dummy agent.** Se trata de un agente de interfaz de usuario, que proporciona al usuario un interfaz para interaccionar con otros agentes.

The screenshot shows the JADE DummyAgent interface. The title bar reads "da0@138.100.13.65:1099/JADE - DummyAgent". Below the title bar are tabs for "General", "Current message", and "Queued message". The "Current message" tab is active, displaying a toolbar with icons for file operations and message actions. The main area is divided into two panes: "ACLMessage" and "Envelope". The "ACLMessage" pane contains fields for "Sender:" (Set da0@138.100.13.65:1099/JAC), "Receivers:", "Reply-to:", "Communicative act:" (accept-proposal), "Content:" (a large text area), "Language:", "Encoding:", "Ontology:", "Protocol:" (Null), "Conversation-id:", "In-reply-to:", "Reply-with:", "Reply-by:" (Set), and "User Properties:". The "Envelope" pane is currently empty.

- ✧ En la mayor parte de las ocasiones se utiliza para realizar pruebas.



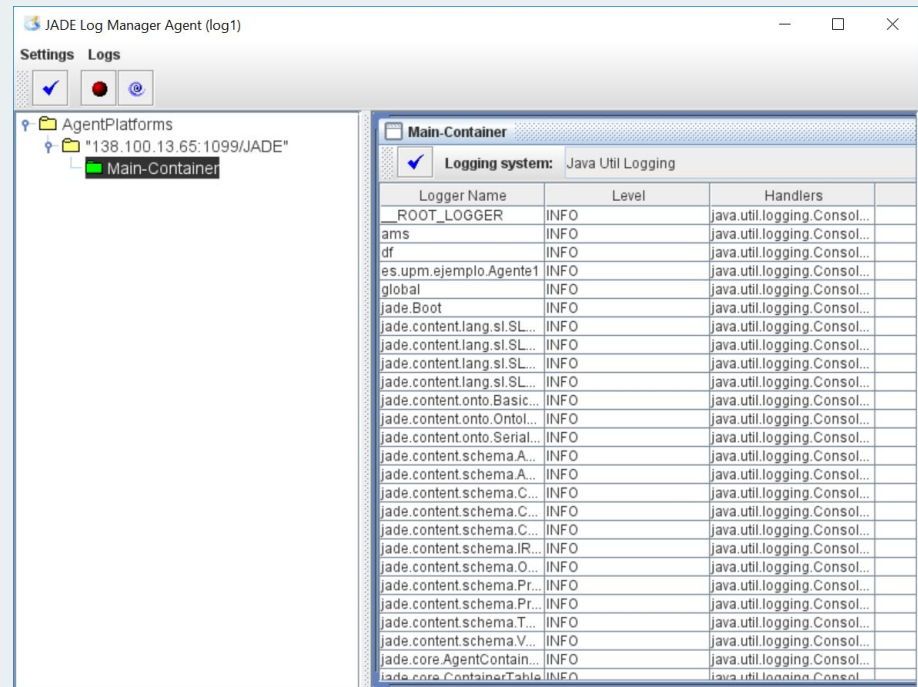
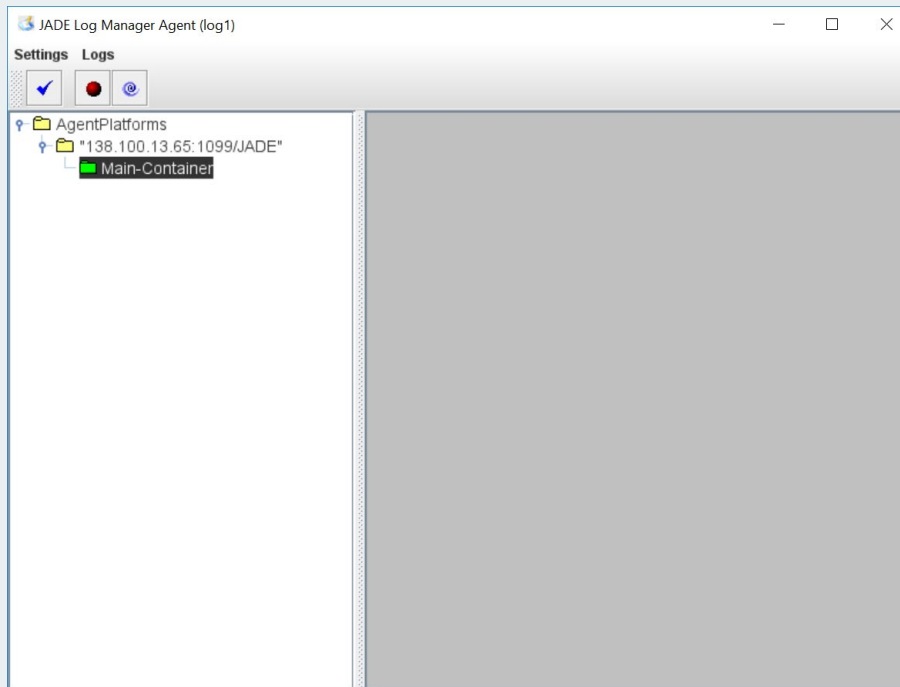
3. Instalación de JADE Configuración



- Podemos observar que:



✦ **Log Manager agent.** Es un agente que lleva a cabo un log de todos los eventos que ocurren en la plataforma.



Main-Container		
<input checked="" type="checkbox"/>	Logging system:	Java Util Logging
Logger Name	Level	Handlers
ROOT_LOGGER	INFO	java.util.logging.Console...
ams	INFO	java.util.logging.Console...
df	INFO	java.util.logging.Console...
es.upm.ejemplo.Agente1	INFO	java.util.logging.Console...
global	INFO	java.util.logging.Console...
jade.Boot	INFO	java.util.logging.Console...
jade.content.lang.sl.SL...	INFO	java.util.logging.Console...
jade.content.lang.sl.SL...	INFO	java.util.logging.Console...
jade.content.lang.sl.SL...	INFO	java.util.logging.Console...
jade.content.lang.sl.SL...	INFO	java.util.logging.Console...
jade.content.onto.Basic...	INFO	java.util.logging.Console...
jade.content.onto.Ontol...	INFO	java.util.logging.Console...
jade.content.onto.Serial...	INFO	java.util.logging.Console...
jade.content.schema.A...	INFO	java.util.logging.Console...
jade.content.schema.A...	INFO	java.util.logging.Console...
jade.content.schema.C...	INFO	java.util.logging.Console...
jade.content.schema.C...	INFO	java.util.logging.Console...
jade.content.schema.C...	INFO	java.util.logging.Console...
jade.content.schema.IR...	INFO	java.util.logging.Console...
jade.content.schema.O...	INFO	java.util.logging.Console...
jade.content.schema.Pr...	INFO	java.util.logging.Console...
jade.content.schema.Pr...	INFO	java.util.logging.Console...
jade.content.schema.T...	INFO	java.util.logging.Console...
jade.content.schema.V...	INFO	java.util.logging.Console...
jade.core.AgentContain...	INFO	java.util.logging.Console...
jade.core.ContainerTabl...	INFO	java.util.logging.Console...



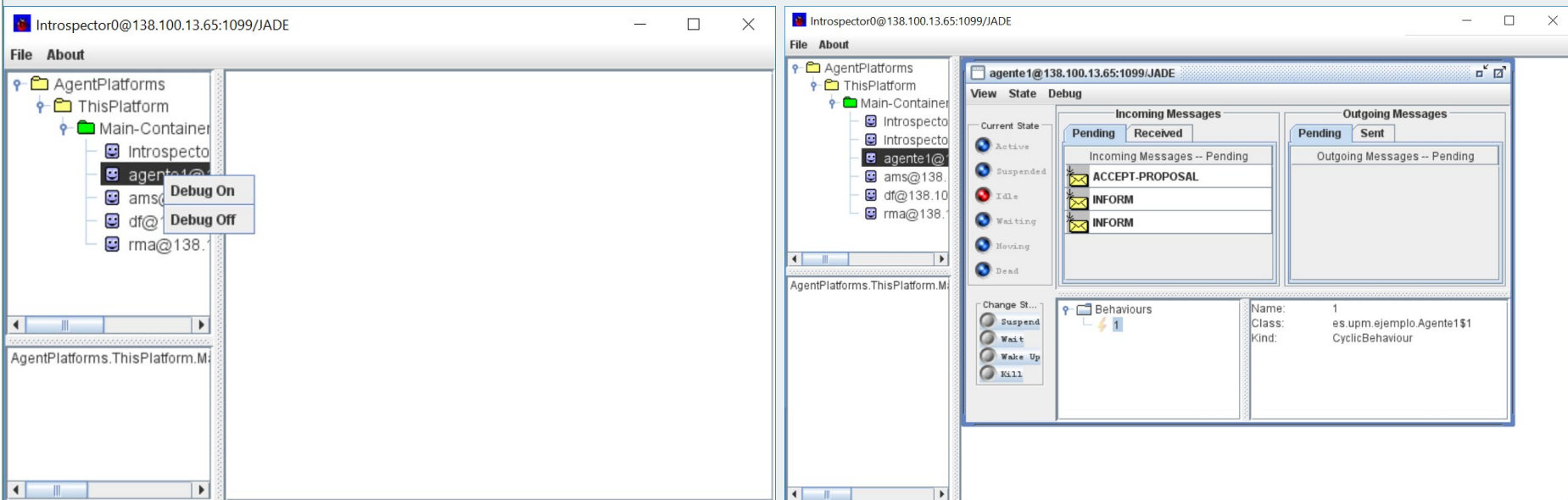
3. Instalación de JADE Configuración



- Podemos observar que:



✦ **Introspector agent.** Permite observar el estado interno y el comportamiento de un agente (y forzar a cambiar el estado del agente).



- ✦ Es posible observar (cada una de estas propiedades la analizaremos a medida que avance la asignatura):
 - Estado interno del agente (current status)
 - Mensajes recibido y mensajes enviados
 - Comportamientos del agente



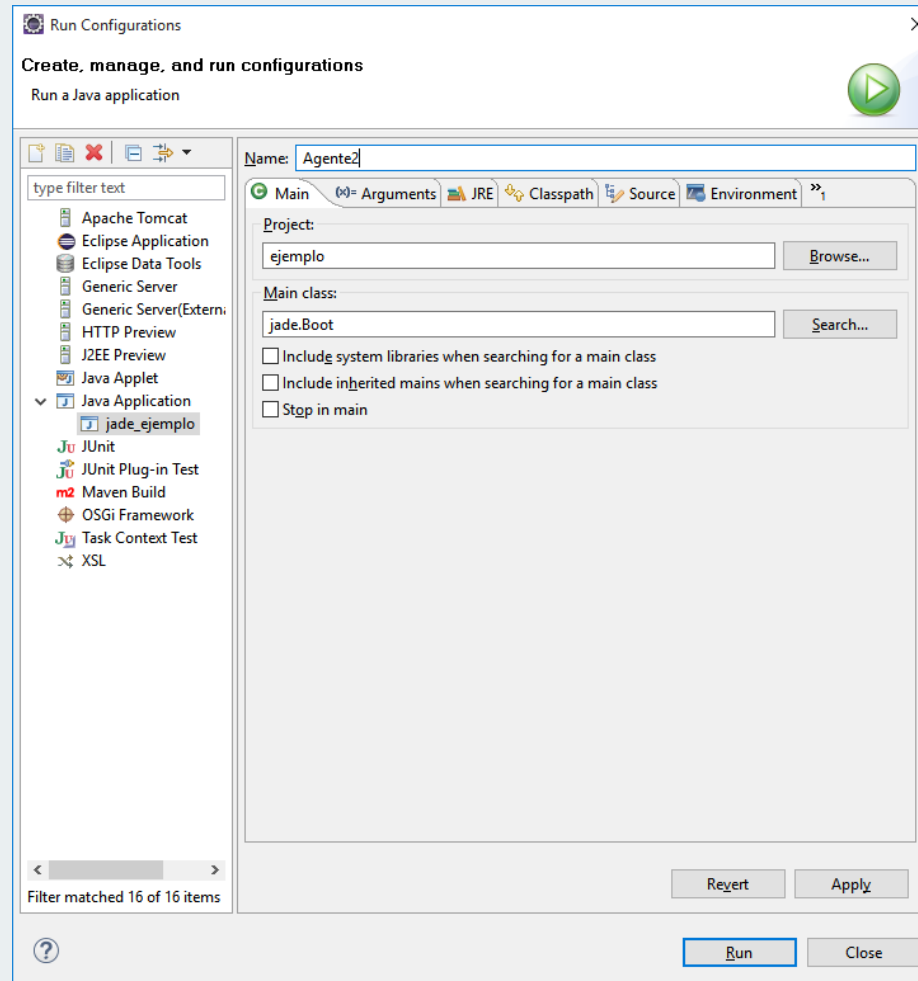
3. Instalación de JADE Configuración



- De cara a seguir analizando el comportamiento de la plataforma, vamos a comprobar ahora como podemos conectar un nuevo agente a la plataforma de forma sencilla. Es decir, tenemos la plataforma en ejecución con nuestro agente Agente1 y vamos a añadir un nuevo agente Agente2
- Solamente es necesario modificar los parámetros que se pasan para la ejecución.
- Copiamos la clase Agente.java con el nombre Agente2.java y ejecutamos el agente para que se una a la plataforma ya creada.
- Para ello sólo hay que indicar los parámetros `-main false -host 127.0.0.1 nombre2:es.upm.ejemplo.Agente2` en lugar de los indicados en la ejecución anterior.



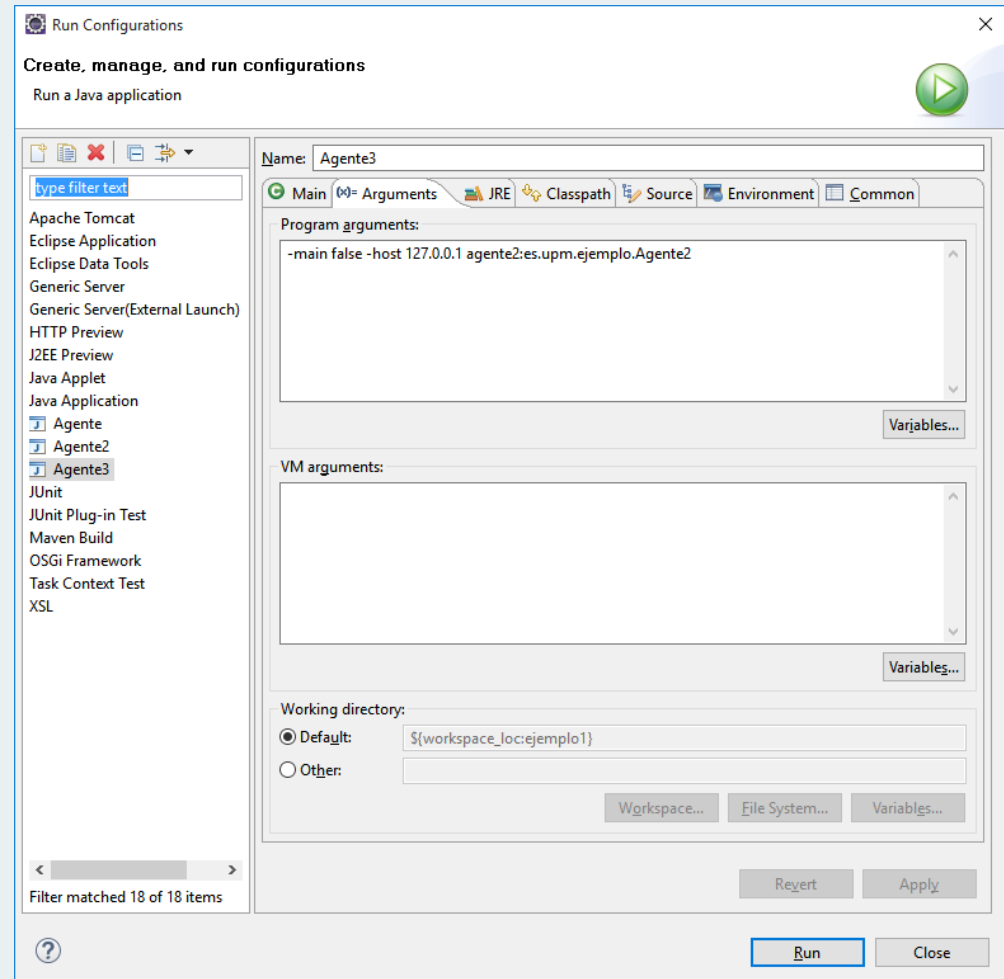
3. Instalación de JADE Configuración





3. Instalación de JADE Configuración

- Añadimos el agente Agente 2 en un nuevo contenedor de la plataforma.
- Es necesario indicar `-main false` para crear un nuevo contenedor.
- Es necesario indicar el host en el que se encuentra el contenedor principal (en nuestro caso es localhost o 127.0.0.1).
- Indicamos la clase correspondiente al nuevo agente que queremos crear.

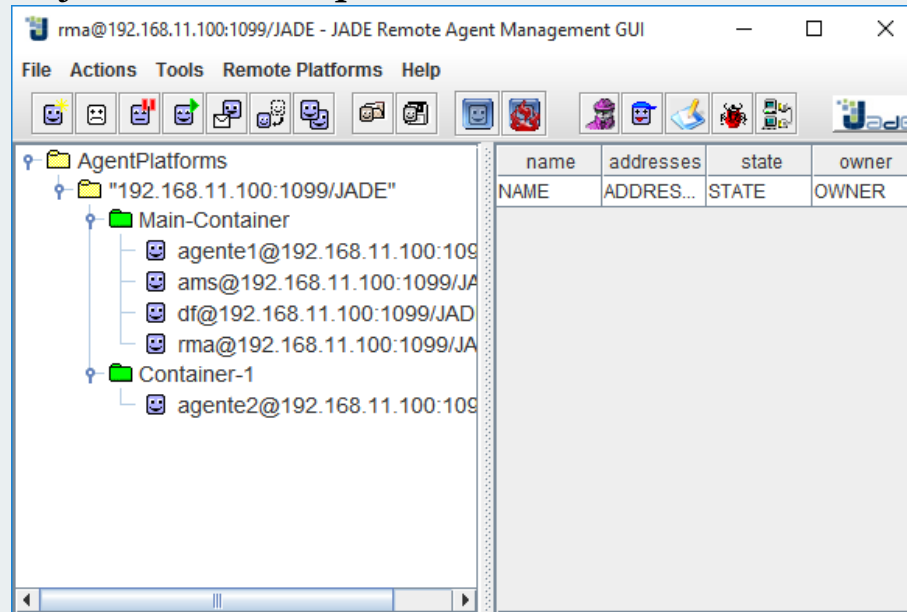




3. Instalación de JADE Configuración



- El resultado de la ejecución es el que se muestra a continuación:



- Se puede apreciar que los dos agentes se encuentran en ejecución en la plataforma.
- Hasta este momento hemos creado agentes con comportamientos muy limitados y que no tienen comunicación entre ellos.
- El siguiente paso consiste en analizar como podemos trabajar con las capacidades y comportamientos de los agentes, así como con las interacciones que se pueden establecer a través del paso de mensajes.