



FACULTAD DE INFORMÁTICA

Proyecto de Innovación Educativa:

Paradigmas de Programación

“PDP-eLearning” (PIE 2010)

PDP-eLearning virtualizado” (PIE 2011)



Objetivos “PDP-eLearning/virtualizado”:

- Impartir una enseñanza b-learning complementaria con la enseñanza presencial.
- **Titulaciones objetivo:**
 - Máster Universitario de Investigación en Software y Sistemas (MUSS)
 - Máster Universitario Profesional en Ingeniería Informática (MUII)



● **Actividades:**

- 1. Diseñar los contenidos educativos** ajustados a una enseñanza e-learning (PIE 2010).
- 2. Implementar los contenidos educativos** de modo que se ajusten eficientemente a una enseñanza impartida en la modalidad e-learning (PIE 2010).
- 3. Depurar los contenidos educativos de la asignatura** (PIE 2011).
- 4. Virtualizar los contenidos educativos** y ubicarlos en la plataforma MOODLE, Aula Virtual, de la Facultad de Informática (PIE 2011).
- 5. Impartir** una enseñanza e-learning complementaria con la enseñanza presencial (PIE 2011)
- 6. Evaluar los resultados** de esta enseñanza b-learning comparándolos con la enseñanza presencial (al finalizar el curso 2012).



- **Diseñar e Implementar los contenidos educativos de cada U.D.(PIE 2010):**
 - Definición del *esquema* de contenidos, *presentación* y *objetivos* de la U.D.
 - Definición en detalle de los *contenidos educativos* de la Unidad
 - Actividades de *Autoevaluación*
 - Conclusiones (conceptos básicos a retener)
 - Bibliografía
 - Se escribieron 13 unidades didácticas
- **Depurar los contenidos educativos de la asignatura (PIE 2011).**
 - Se han depurado las 13 unidades didácticas y se ha incorporado una mas (Paradigma End User Development)
- **Virtualizar los contenidos educativos** y ubicarlos en la plataforma MOODLE, Aula Virtual, de la Facultad de Informática (PIE 2011)
 - Se ha realizado esta tarea
- **Se está publicando un libro de 418 páginas** (PIE 2011)



Temario

Pág

Pt

1.- Paradigmas de Programación. Concepto y Tipos de Paradigmas	24	14
2.- El Paradigma Funcional	28	18
3.- El Paradigma Funcional: Metodología y Entorno	32	19
4.- El Paradigma Lógico	26	12
5.- El Paradigma Lógico: Metodología y Entorno	28	28
6.- El Paradigma Orientado a Agentes: Agentes y Sistemas Multiagentes	32	25
7.- El Paradigma Orientado a Agentes: La Comunicación entre Agentes	32	23
8.- El Paradigma Orientado a Agentes: Metodología y Entorno	32	18
9.- El Paradigma Demostrativo: La Programación Genética	24	13
10.- El Paradigma Dirigido por Eventos de Interfaz de Usuario	26	11
11.- El Paradigma Dirigido por Eventos de Interfaz de Usuario: Tratamiento de Eventos y Threads en Java	42	23
12.- El Paradigma Orientado a Objetos y Eventos: Metodología y Entorno	30	9
13.- El Paradigma "Dataflow"	32	17
14.- El Paradigma "EUD"	30	13
Total	418	243



POLITÉCNICA



Paradigmas de programación

Fernando Alonso Amo
Juan Alfonso Lara Torralbo
David Lizcano Casas
Loïc Martínez Normand



Navegación

- Página Principal
 - Área personal
 - Páginas del sitio
 - Mi perfil
 - Mis cursos
 - SSII-2011
 - ParadigProg**
 - Participantes
 - ✓ Informes
 - General
 - Tema 1
 - Tema 2
 - Tema 3
 - Tema 4
 - Tema 5
 - Tema 6
 - Tema 7
 - Tema 8
 - Tema 9
 - Tema 10
 - Tema 11
 - Tema 12
 - Tema 13
 - Tema 14
 - Tema 15
 - Tema 16
 - Tema 17
 - Cursos
 - Grado en Ingeniería Informática
 - Grado en Matemáticas e Informática
 - Ingeniería Informática
 - Másteres Universitarios
 - Actividades Paralelas

Programa de Programación

Programa de Programación

Temas de programación.

Temas de programación.

Procedimentales u Operacionales.

Imperativo.

Orientado a Objetos.

Declarativos.

de Transformación.

Basados en Formularios.

Basado en Restricciones.

Declarativos

Control en paradigmas

Examen de programación.

Examen de programación

Programación.

Programación.

Programación.



tema 8

Tema 8 El Paradigma Orientado a Agentes. Metodología y Entorno.

📄 Introducción y objetivos.

8.1. Metodologías de desarrollo de sistemas multiagente

📄 8.1. Metodologías de desarrollo de sistemas multiagente

8.2. La metodología SONIA

📄 8.2. La metodología SONIA

📄 8.2.1. Fases de la metodología SONIA

📄 8.2.2. Ejemplo de aplicación: proyecto ALBOR

8.3. Lenguajes de programación orientados a agentes

📄 8.3. Lenguajes de programación orientados a agentes

📄 8.3.1. GOAL

📄 8.3.2. Agent0

📄 8.3.3. 2APL

📄 8.3.4. Bibliotecas de programación

8.4. Entorno de Desarrollo: JADE

📄 8.4. Entorno de Desarrollo: JADE

📄 8.4.1. Arquitectura de JADE

📄 8.4.2. Agentes en JADE

📄 8.4.3. Comportamientos

📄 8.4.4. Comunicación entre agentes

📄 8.4.5. Ejemplo

8.5. Ejercicios de Auto comprobación

📄 8.5. Ejercicios de Auto comprobación

8.6. Conclusiones y bibliografía

📄 8.6. Conclusiones y bibliografía



8.4.1. Arquitectura de JADE

La plataforma de agentes JADE consta de una serie de agentes que se pueden comunicar gracias al sistema de transporte de mensajes que proporciona la plataforma. Además, a través de este sistema es posible comunicar dos o más plataformas de agentes. Por otro lado, JADE aporta un sistema de gestión de los mismos de agentes y un servicio de páginas amarillas que permite a los agentes conocerse entre sí. La figura 14 muestra la arquitectura de la plataforma JADE.

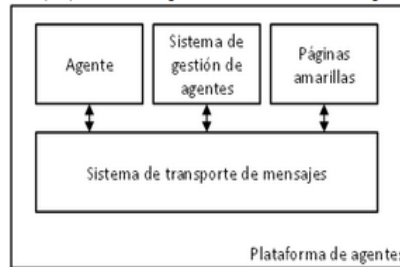


Figura 14. Arquitectura de JADE

En JADE, además de los agentes propiamente dichos, existen diversos agentes y herramientas propios de la plataforma que tienen determinadas funciones que facilitan la depuración de los sistemas implementados. Dichos agentes son los siguientes:

- a) **Remote Monitoring Agent (RMA).** Se trata de un agente que proporciona una interfaz gráfica de la plataforma en la que se pueden visualizar los agentes de la misma y realizar diferentes acciones sobre dichos agentes. También permite lanzar otras herramientas de JADE útiles para los desarrolladores.
- b) **Dummy Agent.** Este agente es una herramienta para la monitorización y depuración. Consta de un terminal gráfico de un agente que permite enviar mensajes a otros agentes y mantener un registro de todos los mensajes intercambiados entre los agentes.
- c) **Interfaz de Directory Facilitator (DF GUI).** Se trata de una interfaz gráfica que, a través de los servicios del agente facilitador, permite examinar y buscar agentes, registrar y borrar agentes, así como cambiar los detalles de los agentes registrados.
- d) **Sniffer Agent.** Es un agente capaz de interceptar todos los mensajes enviados y representarlos gráficamente mediante un diagrama de interacción UML. Resulta de gran utilidad para la depuración. La figura 15 muestra un ejemplo de diagrama generado por el agente Sniffer. En dicha figura se aprecia un intercambio inicial de mensajes entre los agentes *ma* y *ams*. Seguidamente, del instante 3 al 5 se produce un intercambio de mensajes entre un agente denominado *peter* y el agente *ams*. Seguidamente, éste último envía dos mensajes al agente *ma* y, finalmente, es el propio *sniffer0* el que recibe un mensaje de *ams* en el instante 8.

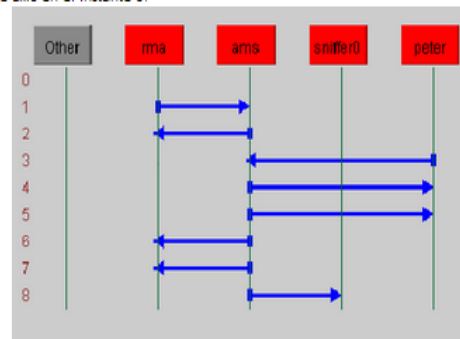


Figura 15. Ejemplo de traza obtenida por el Sniffer Agent

- e) **Introspector Agent.** Permite monitorizar y controlar el ciclo de vida de los agentes, así como gestionar las colas de entrada y salida de mensajes de los agentes.



8.5. Ejercicios de Autoevaluación

1. ¿Es cierto que el paradigma orientado a agentes es el más conveniente para resolver problemas del mundo real? No. El paradigma orientado a agentes permite resolver problemas particulares en los que su uso está indicado. De hecho una de las primeras cuestiones que cabe plantearse es la necesidad o no de utilizar este paradigma para resolver un problema determinado.
2. ¿Una buena metodología de agentes es aquella que puede ser aplicada aún sin tener conocimientos profundos del paradigma orientado a agentes? *Sí, es una de sus características deseables.*
3. ¿Cuántos tipos de metodologías se suelen emplear para el desarrollo de programas orientados a agentes? Tres. Las metodologías basadas en agentes, las metodologías basadas en objetos y las metodologías basadas en Ingeniería del Conocimiento.
4. ¿De cuántas fases se compone la metodología SONIA? Se compone de dos fases: Análisis y Diseño. La fase de Análisis, a su vez, se divide en las etapas de Conceptualización y Análisis Extendido. Por su parte, la fase de Diseño se divide en las etapas de Síntesis, Diseño Arquitectónico y Diseño de Sociedad.
5. ¿Cómo se determinan los agentes con SONIA? Analizando las responsabilidades y estudiando qué posibles objetos o grupos de objetos son candidatos de ser considerados como agentes autónomos.
6. ¿Es cierto que no hay ningún lenguaje específico para la programación orientada a agentes? No. Aunque normalmente se suelen utilizar librerías suministradas por lenguajes para otro propósito, sí existen algunos lenguajes minoritarios propios de este paradigma, tales como GOAL, Agent0 o 2APL.
7. ¿Cuáles son las diferentes secciones en que se divide un programa escrito en lenguaje GOAL? Las secciones son conocimiento, creencias, objetivos, reglas de acción, especificaciones de acción y reglas de percepción.
8. ¿Cuáles son las características mentales propias de los agentes que se incluyen en el lenguaje Agent0? Las creencias y los compromisos, además de las capacidades de cada agente.
9. ¿Cuáles son las principales partes de que se compone la especificación de un agente con el lenguaje 2APL? Creencias, objetivos, acciones, planes y reglas de razonamiento.
10. ¿Cuál es el entorno de desarrollo más extendido en el paradigma orientado a agentes? JADE.
11. ¿JADE está basado en algún lenguaje específico para la programación de aplicaciones con agentes? No. JADE es una plataforma de agentes desarrollados utilizando una librería que proporciona el lenguaje Java.
12. ¿Pueden los agentes JADE ejecutar diferentes tareas simultáneamente? A pesar de poseer un único hilo de ejecución, los agentes pueden ejecutar diferentes tareas a través de diferentes comportamientos cuya ejecución es planificada por la plataforma JADE siguiendo una política Round-Robin.
13. ¿Cuál es el mecanismo de comunicación entre agentes que contempla la plataforma JADE? El paso de mensajes utilizando un lenguaje estándar denominado Agent Communication Language (ACL).
14. ¿Qué son las performativas de los mensajes enviados entre agentes JADE? Son las intenciones comunicativas del mensaje. Resultan esenciales para la comunicación ya que para que un agente reciba un mensaje, éste debe poseer una performativa que coincida con la que el agente está esperando.



- **Impartir la enseñanza e-learning** (curso 2012-2013)
- Se ha iniciado la enseñanza b-learning utilizando el AulaVirtual de la FI:
 - Foros de información y debate
 - Cuestionarios semanales de 5 preguntas
 - Contenidos educativos virtualizados
 - Transparencias de clase



- **Se han definido y diseñado los contenidos educativos e-learning de la asignatura “Paradigmas de Programación”(PIE 2010)**
- **Se han virtualizado estos contenidos educativos, desarrollando 14 unidades didácticas con un volumen de 263 pantallas (PIE 2011).**
- **Se ha escrito un libro “Paradigmas de Programación” de 418 páginas (PIE 2011).**
- **Se ha iniciado una enseñanza e-learning utilizando el AulaVirtual de la FI y las transparencias de clase (PIE 2011).**



- **Implantar una enseñanza e-learning exige un gran esfuerzo por parte del profesor:**
 - ***Diseñar y crear* los contenidos educativos e-learning precisa muchas horas de trabajo (1 pág > 2h.)**
 - ***Impartir* la enseñanza *e-learning* exige gran dedicación permanente por parte de los profesores (cuestionarios semanales , contestar preguntas en los foros, etc)**
 - **Se pasa de una atención *puntual* en el aula con la enseñanza presencial a una atención *continua* con la enseñanza e-learning.**



- **Hasta no evaluar a los alumnos en el año académico 2012/13 no se pueden valorar los resultados de este proyecto b-learning**

- **En la experiencia realizada con la asignatura MDP (PIE 2008/9) se vio que una enseñanza b-learning, *con evaluación continua*, es una poderosa herramienta para:**
 - **Incentivar al alumno captando su atención por la asignatura.**
 - **Que el alumno adquiriera una mejor formación y obtengo mejores notas.**
 - **Y como consecuencia de lo anterior, disminuya el fracaso escolar:
Con el proyecto de innovación educativa MDP se *disminuyó el fracaso escolar* del orden del **20%** en relación con una enseñanza presencial.**



FACULTAD DE INFORMÁTICA

Proyecto de Innovación Educativa:

Paradigmas de Programación

“PDP-eLearning”