

---

**Universidad Complutense de Madrid**  
**Facultad de Informática**



---

**GUÍA DOCENTE**  
**2011/2012**

**Máster en Investigación en Informática**

---



# Guía Docente 2011/12

## Máster en Investigación en Informática

1. La Facultad de Informática de la Universidad Complutense de Madrid .....	4
1.1. Localización .....	4
2. Información docente .....	5
2.1. Objetivos formativos del Máster en Investigación en Informática.....	5
2.2. Estructura de los estudios.....	5
2.3. Horarios .....	9
2.4. Fichas Docentes .....	12
3. Recursos y Servicios de la Facultad de Informática.....	54
3.1. Aulas .....	55
3.2. Laboratorios .....	55
3.3. Biblioteca .....	55
3.4. Organización académica .....	56
3.5. Departamentos .....	57
3.6. Asociaciones .....	57
3.7. Delegación de alumnos .....	58
3.8. Teléfonos y direcciones de interés.....	58
3.9. MIGS: Museo de Informática García Santesmases.....	59
3.10. Otros servicios .....	59
Servicio de comedor y cafetería .....	59
Espacio WIFI .....	59
Servicio de reprografía.....	60
Cajero automático.....	60
4. Normas académicas.....	60
4.1. Calendario Académico del curso 2011/12.....	60
4.2. Normas y periodos de matrícula 2011/12.....	60
Proceso de admisión .....	60
4.3. Trabajos de Fin de Máster.....	61
Normativa para los Trabajos de Fin de Master .....	61
Calendario para trabajos de fin de máster 2011 y 2012 .....	64
4.4. Reglamento de Laboratorios.....	65
4.5. Reglamento de Biblioteca .....	65
Mediateca y Salas de Trabajo en Grupo .....	66
Salas de Lectura .....	66
4.6. Normas básicas de convivencia en la Facultad de Informática .....	66

4.7. Defensor del universitario .....67

## 1. La Facultad de Informática de la Universidad Complutense de Madrid

La Universidad Complutense de Madrid (UCM) fue pionera en España en el área de la Informática al introducirla como tema de investigación y docencia hace más de medio siglo. Ya en los años 50 existía un grupo de investigación de la UCM en estrecha relación con otros grupos de Cambridge y Harvard, universidades donde se estaba gestando lo que hoy conocemos como Informática. En la década de los 60 se impartieron cursos de doctorado en estos temas, así como cursos internacionales en el ámbito de la UNESCO. A principios de los 70, antes de la creación de las primeras Facultades de Informática del país, la UCM instaura especialidades de Informática en las Licenciaturas de Ciencias Físicas y Matemáticas.

En octubre de 1991 se crea en la UCM la Escuela Superior de Informática que después, en abril del año 2000, cambia su nombre por el de Facultad de Informática. En ella se han venido impartiendo las tres titulaciones oficiales de nivel universitario: Ingeniería en Informática, Ingeniería Técnica en Informática de Gestión e Ingeniería Técnica en Informática de Sistemas. Nuestra Facultad ha sido el centro universitario de la Comunidad de Madrid más demandado para las tres titulaciones impartidas. El edificio de la Facultad de Informática fue inaugurado en marzo de 2003 y alberga actualmente un total de 2200 alumnos.

A partir del curso 2010-2011 se pusieron en marcha tres nuevas titulaciones de grado dentro del Espacio Europeo de Educación Superior que además cumplen las especificaciones del Acuerdo del Consejo de Universidades (BOE 4 de agosto de 2009) sobre los títulos oficiales en el ámbito de la Ingeniería Técnica Informática. Las tres titulaciones dan acceso directo al Máster en Ingeniería Informática. Estas titulaciones son: Graduado en Ingeniería Informática, Graduado en Ingeniería de Computadores y Graduado en Ingeniería del Software. Adicionalmente, la Facultad de Informática también imparte junto con la Facultad de Ciencias Matemáticas el Doble Grado en Ingeniería Informática y en Matemáticas.

También se imparten en la Facultad de Informática estudios oficiales de posgrado, incluyendo el Máster en Investigación en Informática que se describe en esta guía, y doctorado (con mención de calidad ANECA), diversos títulos propios de especialización profesional y una extensa gama de actividades formativas complementarias, conferencias de informática básica e investigación, y cursos formativos gratuitos impartidos por empresas.

La Facultad de Informática de la UCM cuenta con las más modernas instalaciones, laboratorios de última generación de programación, diseño de circuitos, robótica, redes, sistemas operativos, etc., una biblioteca plenamente equipada y red WiFi en todo el edificio.

Toda la información referente a la Facultad puede encontrarse en la página web [www.fdi.ucm.es](http://www.fdi.ucm.es), clasificada tanto por temas de interés (alumnos, profesorado, posgrado, etc.) como por áreas de trabajo (biblioteca, laboratorios, departamentos, etc.).

### 1.1. Localización

La Facultad de Informática de la UCM se encuentra situada en el Campus de Moncloa (Ciudad Universitaria), en pleno centro de la ciudad de Madrid y a 500 metros de la estación de Metro "Ciudad Universitaria".

**Dirección:** C/ Profesor José García Santesmases, s/n  
28040 Madrid, España  
Teléfono: 91 394 7501  
Fax: 91 394 7510

#### Transporte público

**Metro:** Línea 6 (Ciudad Universitaria)

**Autobuses:** U (Autobús Universitario), F (Cuatro Caminos), G (Moncloa), 82 (Moncloa)

## **2. Información docente**

### **2.1. Objetivos formativos del Máster en Investigación en Informática**

El Máster Oficial en Investigación en Informática, regulado por el R.D. 1393/2007, es un máster de capacitación investigadora, que tiene por objeto ofrecer una formación de carácter avanzado en las facetas más relevantes de la Ingeniería Informática, tanto en aspectos centrales del área como en otros de naturaleza multidisciplinar.

El alumno, en función de los estudios que le dan acceso, debe cursar un mínimo de 60 ECTS y un máximo de 120 ECTS. Está dirigido tanto a los actuales Ingenieros en Informática (o equivalentes), que cursarían 60 ECTS, como a titulados en Ingeniería Técnica en Informática (o equivalentes), como a titulados técnicos o superiores en carreras científico-tecnológicas, que dependiendo de su formación previa podrían cursar hasta 120 ECTS.

El Máster ofrece un catálogo de asignaturas lo suficientemente amplio de manera que posibilite la especialización del estudiante en los diversos ámbitos de la investigación en el sector de las Tecnologías de la Información y las Comunicaciones (TIC). Bajo esta premisa los objetivos docentes del Máster en Investigación en Informática abarcan las siguientes líneas de investigación:

- Control, modelado, simulación y robótica,
- Computación de altas prestaciones y computación en grid,
- Automatización del diseño de sistemas digitales,
- Ingeniería del software e inteligencia artificial,
- Métodos formales para la programación,
- Programación declarativa multiparadigma.

Además de una especialización en las áreas de trabajo mencionadas, este máster ofrece una formación orientada a que los estudiantes sepan:

- Aplicar los conocimientos adquiridos a resolver problemas en entornos nuevos o poco conocidos dentro de contextos más amplios.
- Buscar y seleccionar la información útil necesaria para resolver problemas complejos, manejando con soltura las fuentes bibliográficas del campo.
- Integrar conocimientos y enfrentarse a la complejidad de formular juicios a partir de una información que, siendo incompleta o limitada, incluya reflexiones sobre las responsabilidades sociales y éticas vinculadas a la aplicación de sus conocimientos y juicios.
- Comunicar sus conclusiones, así como los conocimientos y razones últimas que las sustentan, a públicos especializados y no especializados de un modo claro y sin ambigüedades.
- Aplicar habilidades de aprendizaje que les permitan continuar estudiando de un modo que habrá de ser en gran medida autodirigido o autónomo.

### **2.2. Estructura de los estudios**

El alumno que acceda al Máster en Investigación en Informática con una titulación superior tiene que cursar 60 ECTS incluyendo un Trabajo de Fin de Máster de 30 ECTS y al menos 30 ECTS en asignaturas optativas. En este máster existen 4 especialidades; para obtener dicha mención, el alumno debe realizar el Trabajo de Fin de Máster en la especialidad correspondiente y cursar un mínimo de 20 ECTS en asignaturas optativas asociadas a la especialidad. Sin embargo, no es obligatorio hacer una especialidad.

El alumno que acceda al Máster con una titulación técnica tiene que cursar, además de los 60 ECTS en las condiciones descritas en el párrafo anterior, entre 0 y 60 créditos de Complementos de Formación según decida la Comisión de Posgrado al admitirlo.

## FORMACIÓN BASE

GEA	ASIGNATURA	TIPO	ESPECIALIDAD	ECTS
600884	<b>122013</b> Bioinformática	Optativa	Ingeniería de computadores	6
600885	<b>122014</b> Compiladores para Arquitecturas de Alto Rendimiento	Optativa	Ingeniería de computadores	6
600886	<b>122015</b> Computación en Red y Tecnología Grid	Optativa	Ingeniería de computadores	6
e600887	<b>122016</b> Hardware Dinámicamente Reconfigurable	Optativa	Ingeniería de computadores	6
600888	<b>122017</b> Metodología de Diseño de Sistemas Asíncronos	Optativa	Ingeniería de computadores	6
600889	<b>122018</b> Síntesis Arquitectónica de Alto Nivel	Optativa	Ingeniería de computadores	6
600890	<b>122019</b> Técnicas de Alto Rendimiento en el Diseño de Procesadores	Optativa	Ingeniería de computadores	6
600891	<b>122020</b> Tecnologías Avanzadas de Redes y Telecomunicaciones	Optativa	Ingeniería de computadores	6
600892	<b>122021</b> Control Inteligente	Optativa	Ingeniería informática para la industria	6
600893	<b>122022</b> Diseño de Sistemas Empotrados	Optativa	Ingeniería informática para la industria	6
600894	<b>122023</b> Optimización en el Control de Procesos	Optativa	Ingeniería informática para la industria	6
600895	<b>122024</b> Percepción Visual Artificial	Optativa	Ingeniería informática para la industria	6
600896	<b>122025</b> Redes Neuronales y sus Aplicaciones	Optativa	Ingeniería informática para la industria	6
600897	<b>122026</b> Análisis y Transformación de Programas	Optativa	Programación tecnología software	y 7,5
600898	<b>122027</b> Aplicación de Principios Económicos al Desarrollo de Sistemas	Optativa	Programación tecnología software	y 6

GEA	ASIGNATURA	TIPO	ESPECIALIDAD	ECTS
	Computacionales			
600899	<b>122028</b> Especificación y Validación del Software	Optativa	Programación tecnología software y	7,5
600900	<b>122029</b> Modelado de Sólidos y Técnicas de Visualización	Optativa	Programación tecnología software y	7,5
600901	<b>122030</b> Modelos Abstractos de Cómputo	Optativa	Programación tecnología software y	4,5
600902	<b>122031</b> Modelos de Sistemas Concurrentes y Distribuidos	Optativa	Programación tecnología software y	7,5
600903	<b>122032</b> Requisitos Software	Optativa	Programación tecnología software y	4,5
600904	<b>122033</b> Seguridad de Computadores	Optativa	Programación tecnología software y	6
600905	<b>122034</b> Aprendizaje Automático	Optativa	Sistemas inteligentes	6
600906	<b>122035</b> Agentes Inteligentes	Optativa	Sistemas inteligentes	6
600907	<b>122036</b> Deducción Automática	Optativa	Sistemas inteligentes	4,5
600908	<b>122037</b> E-Learning	Optativa	Sistemas inteligentes	6
600909	<b>122038</b> Extensiones de Programación Lógica	Optativa	Sistemas inteligentes	6
600910	<b>122039</b> Ingeniería Lingüística	Optativa	Sistemas inteligentes	6
600911	<b>122040</b> Razonamiento Aproximado y con Incertidumbre	Optativa	Sistemas inteligentes	4,5
600912	<b>122041</b> Trabajo fin de Master en Ingeniería de Computadores	Obligatoria (según especialidad)	Ingeniería de computadores	30
600913	<b>122042</b> Trabajo fin de Master en Ingeniería Informática para la Industria	Obligatoria (según especialidad)	Ingeniería informática para la industria	30
600914	<b>122043</b> Trabajo fin de Master en Programación y Tecnología Software	Obligatoria (según especialidad)	Programación tecnología software y	30
600915	<b>122044</b> Trabajo fin de Master en Sistemas Inteligentes	Obligatoria (según especialidad)	Sistemas inteligentes	30

## COMPLEMENTOS DE FORMACIÓN

GEA	ASIGNATURA	TIPO	ESPECIALIDAD	ECTS
601356	<b>122001</b> Estructura de Datos y de la Información	N/A	N/A	12
601350	<b>122002</b> Metodología y Tecnología de la Programación	N/A	N/A	12
601357	<b>122003</b> Teoría de Autómatas y Lenguajes Formales	N/A	N/A	9
601351	<b>122004</b> Procesadores de Lenguaje	N/A	N/A	9
601358	<b>122005</b> Estructura y Tecnología de Computadores	N/A	N/A	9
601359	<b>122006</b> Sistemas Operativos	N/A	N/A	6
601353	<b>122007</b> Redes	N/A	N/A	9
601354	<b>122008</b> Arquitectura e Ingeniería de Computadores	N/A	N/A	9
601355	<b>122009</b> Bases de Datos y Sistemas de Información	N/A	N/A	12
601363	<b>122010</b> Ingeniería del Software 1	N/A	N/A	9
601364	<b>122011</b> Ingeniería del Software 2	N/A	N/A	9
601352	<b>122012</b> Inteligencia Artificial e Ingeniería del Conocimiento	N/A	N/A	9



## 2.3. Horarios

Las asignaturas de Formación Base del Master se imparten en las Aulas 5, 6, 11 y 12 de la Facultad de Informática con los siguientes horarios. Para más información sobre las asignaturas de complementos de formación consultar la página Web de la facultad.

### Máster en Ingeniería Informática - Complementos de Formación - Curso 2011-12 - Horarios y Profesores

Abrev.	Asignatura	Grupo Asoc.	Horario 1Q	Horario 2Q	Profesores
AIC	ARQUITECTURA E INGENIERIA DE COMPUTADORES	Ing.Inf. 4ºB	L,M 16-17, X15-16	L16-17, M17-18	Martín Llorente, Ignacio Hermida Correa, Román
BDSI	BASES DE DATOS Y SISTEMAS DE LA INFORMACION	Ing.Inf. OptºC	L,M 19-21	L,M 19-21	Blanco Moreno, Diego
EDI	ESTRUCTURA DE DATOS Y DE LA INFORMACION	Ing.Inf.2º	L,M,X,J 19-20	L,M,X,J 19-20	Gómez-Zamalloa Gil, Miguel García Merayo, María de las Mercedes
ETC	ESTRUCTURA Y TECNOLOGÍA DE COMPUTADORES	Ing.Tec.Gest.2º	X,J,V 19-20	X,J,V 19-20	Vázquez Poletti, José Luis García Repetto, Pedro Luis
IS1	INGENIERÍA DEL SOFTWARE 1	Ing.Inf. 4º B	X,J,V 18-20		Blanco Rodríguez, Miguel ángel
IS2	INGENIERÍA DEL SOFTWARE 2	Ing.Inf. 4º B		X,J,V 18-20	Blanco Rodríguez, Miguel ángel
IAIC	INTELIGENCIA ARTIFICIAL E INGENIERIA DEL CONOCIMIENTO	Ing.Inf. 4º B	X,J,V 17-18	X,J,V 17-18	Gómez Gauchía, Héctor
MTP	METODOLOGÍA Y TECNOLOGÍA DE LA PROGRAMACIÓN	Ing.Inf. 3º C	L,M,X,J,V 10-11	L,J,V 10-11	Segura Díaz, Clara María
Plg	PROCESADORES DE LENGUAJE	Ing.Inf. 4º B	X,J,V 16-17	X,J,V 16-17	Pareja Lora, Antonio
RED	REDES	Ing.Inf. 4º B	L 17-19, M18-19	L 17-19, M18-19	Septién del Castillo, Julio
SO	SISTEMAS OPERATIVOS	Ing.Inf. 3º C	M,X,J,V 11-12		Higuera Toledano, María Teresa
TALF	TEORÍA DE AUTÓMATAS Y LENGUAJES FORMALES	Ing.Tec.Sist.2º	L,M,V 16-17	L,M,V 16-17	Rosa Velardo, Fernando

## Máster en Ingeniería Informática - Curso 2011-12 - Asignaturas Optativas - Horarios y Profesores

Abrev.	Asignatura	Cuatr.	Horario	Profesores
AA	APRENDIZAJE AUTOMATICO	1	L 16-18h	Gómez Albarrán, María de las Mercedes Díaz Agudo, María Belén Gonzalez Calero, Pedro Antonio
AI	AGENTES INTELIGENTES	1	J 14-16h	Fuentes Fernández, Rubén Pavón Mestras, Juan Luis
APEDS	APLICACION DE PRINCIPIOS ECONOMICOS AL DESARROLLO DE SISTEMAS COMPUTACIONALES	1	X 16-18h	Rubio Díez, Fernando
ATP	ANALISIS Y TRANSFORMACION DE PROGRAMAS	1, 2	J 16-18h	Segura Díaz, Clara María Peña Mari, Ricardo Vicente Albert Oriol, Elvira María
BIO	BIOINFORMATICA	2	X 16-18h	María Victoria López López
CAAR	COMPILADORES PARA ARQUITECTURAS DE ALTO RENDIMIENTO	2	M 18-20h	Tenllado Van Der Reijden, Christian Tomas Piñuel Moreno, Luis Gómez Pérez, José Ignacio
CI	CONTROL INTELIGENTE	1	J 16-18h	Santos Peña. Matilde
CRTG	COMPUTACION EN RED Y TECNOLOGIA GRID	1	L 16-18h	Santiago Moreno, Rubén Manuel Martín Llorente, Ignacio Huedo Cuesta, Eduardo
DA	DEDUCCION AUTOMATICA	1	J 14-16h	Nuñez García, Manuel García Merayo, María de las Mercedes Rodríguez Laguna, Ismael
DSE	DISEÑO DE SISTEMAS EMPOTRADOS	1	X 16-18h	Tenllado Van Der Reijden, Christian Tomas Gómez Pérez, José Ignacio
EL	E-LEARNING	1	M 18-20h	Moreno Ger, Pablo Martínez Ortiz, Iván Fernández Manjón, Baltasar Navarro Martín, Antonio Sierra Rodríguez, José Luis
EPL	EXTENSIONES DE PROGRAMACIÓN LÓGICA	1	L 18-20h	Caballero Roldán, Rafael López Fraguas, Francisco Javier Sánchez Hernández, Jaime
EVS	ESPECIFICACION Y VALIDACION DEL SOFTWARE	1, 2	X 14-16h	Verdejo López, José Alberto Martí Oliet, Narciso

				Palomino Tarjuelo, Miguel García Clavel, Manuel
HDR	HARDWARE DINAMICAMENTE RECONFIGURABLE	2	L 16-18h	Sánchez-Elez Martín, Marcos Mecha López, Hortensia
IL	INGENIERIA LINGÜÍSTICA	1	M 16-18h	Gervás Gómez-Navarro, Pedro Pablo Díaz Esteban, Alberto Francisco Gilmartín, Virginia
MAC	MODELOS ABSTRACTOS DE COMPUTO	1	L 16-18h	López Fraguas, Francisco Javier
MSCD	MODELOS DE SISTEMAS CONCURRENTES Y DISTRIBUIDOS	1, 2	M 16-18h	Llana Díaz, Luis Fernando Frutos Escrig, David de Ortega Mallén, Yolanda
MSTV	MODELADO DE SOLIDOS Y TECNICAS DE VISUALIZACION	1, 2	J 18-20h	Martín de la Calle, Pedro Jesús Verdejo López, José Alberto Gavilanes Franco, Antonio
OCP	OPTIMIZACION EN EL CONTROL DE PROCESOS	2	M 16-18h	Cruz García, Jesús Manuel de la
PVA	PERCEPCION VISUAL ARTIFICIAL	1	L 18-20h	Pajares Martinsanz, Gonzalo Guijarro Mata-García, María
RAI	RAZONAMIENTO APROXIMADO Y CON INCERTIDUMBRE	2	X 18-20h	Garmendia Salvador, Luis
RNA	REDES NEURONALES Y SUS APLICACIONES	2	L 18-20h	López Orozco, José Antonio Jiménez Castellanos, Juan Francisco
RS	REQUISITOS SOFTWARE	1	X 16-18h	Moreno Ger, Pablo Martínez Ortiz, Iván Navarro Martín, Antonio Fuentes Fernández, Rubén Sierra Rodríguez, José Luis
SC	SEGURIDAD DE COMPUTADORES	2	X 18-20h	Imaña Pascual, José Luis Botella Juan, Guillermo
SAAN	SINTESIS ARQUITECTONICA DE ALTO NIVEL	1	J 14-16h	Mendías Cuadros, José Manuel Molina Prego, María del Carmen Hermida Correa, Román
TARDP	TECNICAS DE ALTO RENDIMIENTO EN EL DISEÑO DE PROCESADORES	2	M16-18h	Prieto Matías, Manuel Chaver Martínez, Daniel Ángel Tirado Fernández, Francisco
TART	TECNOLOGÍAS AVANZADAS DE REDES Y TELECOMUNICACIONES	1	X 18-20h	García Villalba, Luis Javier

## 2.4. Fichas Docentes

---

### FORMACIÓN BASE

---

#### 122013- Bioinformática (B)

---

**Departamento:** Arquitectura de Computadores y Automática

**Duración:** Cuatrimestral (2º Cuatrimestre)

**Curso:** 2º

**Nivel:** Posgrado

**Tipo:** Optativa

**Profesores que imparten la asignatura:**

Victoria López - Da Ruan (pendiente de aprobación)

**Créditos:** 4,5

**Horas/semana:** 2 horas en aula (Miércoles de 16h a 18h), 0 en laboratorio (2º cuatrimestre)

**Objetivos:**

La Bioinformática aborda el tratamiento y análisis de información en biología y biomedicina mediante el desarrollo y uso de procedimientos, métodos y sistemas basados en las tecnologías de la información e informática. El objetivo de este curso es proporcionar una visión funcional de las diferentes áreas en que tiene presencia la Bioinformática, entrenarse en las herramientas de uso cotidiano y suministrar una perspectiva de las líneas de actuación futura en este campo.

**Conocimientos y destrezas que se requieren:**

**Contenidos:**

1. Introducción a la Bioinformática
2. Bases de datos en Biología
3. Análisis de Secuencias. Filogenias
4. Estructura de proteínas y ácidos nucleicos
5. Análisis de datos: técnicas de agrupamiento
6. Minería de datos en Bioinformática

**Conocimientos y destrezas que se adquieren:**

**Idioma en que se imparte:** Español

**Método docente:** Clases teóricas. Dinámicas de clase y presentación de trabajos.

**Exámenes:** Examen final (40% de la calificación)

**Método de evaluación:** 20% asistencia 40% trabajo 40% examen

**Bibliografía:**

- Arthur M. Lesk, *Introduction to Bioinformatics*; Oxford University Press. 2005;
- Neil C. Jones, Pavel A. Pevzner, *An Introduction to Bioinformatics Algorithms* 2004;
- Jean-Michel Claverie, Cedric Notredame. *Bioinformatics for Dummies*. 2003;
- Andreas D. Baxevanis, B. F. Francis Ouellette; *Bioinformatics: A Practical Guide to the Analysis of Genes and Proteins*; Second Edition. Wiley-Interscience. 2001;
- Stanley Letovsky; *Bioinformatics: Databases and Systems*; Kluwer Academic Publishers. 1999;

- Philip E. Bourne; *Structural Bioinformatics*; Helge Weissig. Wiley-Liss. 2003;

**Página web:**

## **122014 - Compiladores para Arquitecturas de Alto Rendimiento (CAAR)**

---

**Departamento:** Arquitectura de Computadores y Automática

**Duración:** Cuatrimestral (2º Cuatrimestre)

**Curso:** 2º

**Nivel:** Posgrado

**Tipo:** Optativa

**Profesores que imparten la asignatura:**

Luis Piñuel Moreno (coordinador);

**Créditos:** 6

**Horas/semana:** 2 horas en aula, 0 en laboratorio (2º cuatrimestre)

**Objetivos:** Conocer las optimizaciones más relevantes que realizan los compiladores para explotar las capacidades ofrecidas por las microarquitecturas actuales

**Conocimientos y destrezas que se requieren:** Interacción computador - compilador: nivel elemental; Entornos de desarrollo de lenguajes de programación: nivel elemental; Paralelismo a nivel de instrucción: nivel avanzado; Optimización de la jerarquía de memoria: nivel avanzado; Arquitecturas de paralelismo explícito: nivel elemental; Programación paralela con directivas: nivel medio; Lenguaje C++: nivel avanzado; Lenguaje C: nivel avanzado; Construcción de analizadores-traductores: nivel elemental;

**Contenidos:** Interacción entre el compilador y las arquitecturas (de alto rendimiento). Análisis de código. Generación y optimización de código. Profiling e instrumentación.

**Conocimientos y destrezas que se adquieren:** Implementación de técnicas de análisis, generación y optimización de código;

**Idioma en que se imparte:** Español

**Método docente:** Enseñanza Presencial Teórica; Realización (individual / en grupo) de (prácticas / trabajos / proyectos / ejercicios / problemas) no tutorizados.

**Exámenes:** Examen final en junio y septiembre.

**Método de evaluación:** Realización (individual / en grupo) de (prácticas / trabajos / proyectos / ejercicios / problemas):

**Bibliografía:**

Advanced Compiler Design and Implementation; *Steven S. Muchnick* ; Morgan Kaufmann; 1997;

Aart J. C. Bik, Aart J.C. Bik ; Software Vectorization Handbook, The: Applying Intel Multimedia Extensions for M; Intel Press; 2004;

Allen and Kennedy; *Optimizng Compilers for Modern Architectures*; Morgan Kaufmann;2002;

A. V. Aho, M. S. Cam, R. Sethi, I. D. Ullman; *Compilers: Principles, techniques and tools*; Addison Wesley, 2006;



**Página web:**www.ucm.es/campusvirtual

## **122015 - Computación en Red y Tecnología Grid (CRTG)**

---

**Departamento:** Arquitectura de Computadores y Automática

**Duración:** Cuatrimestral (1er cuatrimestre)

**Curso:** 2º

**Nivel:** Posgrado

**Tipo:** Optativa

**Profesores que imparten la asignatura:**

Rubén Manuel Santiago Montero (coordinador)

**Créditos:** 6

**Horas/semana:** 2 horas en aula, 0 en laboratorio (1er cuatrimestre)

**Objetivos:** Ofrecer al alumno una visión global de las últimas tendencias en computación en red y la tecnología Grid. Adicionalmente se realizan prácticas con el Globus Toolkit

**Conocimientos y destrezas que se requieren:** Manejo de sistemas Unix (como usuario): nivel medio; Administración de sistemas Unix: nivel elemental; Lenguaje de Programación C: nivel elemental; Lenguaje de Programación JAVA: nivel elemental;

**Contenidos:** 1. Últimas tendencias en Computación en Red y Tecnología Grid 2. Gestión de recursos distribuidos en la intra-net 3. Estructura del Globus Toolkit 4. Introducción a la Gestión de la Seguridad en el Grid. 5. Gestión de Recursos 6. Servicios de Información 7. Servicio de Transferencia de Ficheros 8. Preparación del entorno de Trabajo 9. Prácticas con el Globus Toolkit

**Conocimientos y destrezas que se adquieren:** Tecnología Grid: nivel medio;

**Idioma en que se imparte:** Español

**Método docente:** Clases teóricas en aula y prácticas en laboratorio

**Exámenes:** Examen final en febrero y septiembre.

**Método de evaluación:** Defensa individual de un trabajo. Asistencia a clase. Participación en clase

**Bibliografía:**

Ian Foster and Carl Kesselman; *The Grid, Blueprint for a New Computing Infrastructure*; Morgan Kaufmann Publishers, Inc., 1998.;

Borja Sotomayor and Lisa Childers; *Globus Toolkit 4. Programming Java Services*; Morgan Kaufmann 2006;

**Página web:**Campus Virtual

## **122016 - Hardware Dinámicamente Reconfigurable (HDR)**

---

**Departamento:** Arquitectura de Computadores y Automática

**Duración:** Cuatrimestral (2º Cuatrimestre)

**Curso:** 2º

**Nivel:** Posgrado

**Tipo:** Optativa

**Profesores que imparten la asignatura:**

Hortensia Mecha López (coordinador);  
Marcos Sánchez-Elez

**Créditos:** 6

**Horas/semana:** 1 horas en aula, 1 en laboratorio (2º cuatrimestre)

Créditos BOE: 6 Créditos ECTS: 6

**Objetivos:** Conocimiento de las arquitecturas de HW dinámicamente reconfigurable existentes, académicas y comerciales, y de los nuevos problemas que presenta su gestión. Diseño de Sistemas empujados con multitarea hardware autoreconfigurables

**Conocimientos y destrezas que se requieren:** Tecnología de hardware reconfigurable (FPGAs): nivel medio; Lenguaje de descripción de hardware (VHDL): nivel medio;

**Contenidos:** 1- Introducción al hardware reconfigurable y la reconfiguración dinámica. 2- Arquitecturas comerciales de grano fino y métodos de reconfiguración 3-Gestión de recursos en arquitecturas de grano fino 4-Otras arquitecturas reconfigurables 5-Prácticas de diseño con hardware dinámicamente reconfigurable

**Conocimientos y destrezas que se adquieren:** Tecnologías de hardware dinámicamente reconfigurable: nivel avanzado; Arquitecturas de hardware reconfigurable: nivel avanzado; Diseño de sistemas empujados con multitarea hardware autoreconfigurables: nivel medio

**Idioma en que se imparte:** Español

**Método docente:** Clases magistrales 40%. Defensa de casos prácticos 10%. Prácticas en laboratorio 50%

**Exámenes:** Evaluación final en junio y septiembre.

**Método de evaluación:** Asistencia y participación en clase 10%. Realización y defensa individuales de trabajos de investigación 20%. Realización de prácticas 30%. Proyecto final 40%

#### **Bibliografía:**

K. Compton, S. Hauck; "Reconfigurable Hardware: A Survey of Systems and Software"; ACM Computing Surveys, 2002;

O. Diessel, G. Wigley; Tech. report ACRC-99-018, 1999;

Stamatis Vassiliadis and Dimitrios Soudris "Fine and Coarse Grain Reconfigurable Computing" Editorial Springer, 2007

Manuales FPGAs de Xilinx <http://www.xilinx.com/support/documentation>

Manuales FPGAs Altera <http://www.altera.com/literature>

Partial Reconfiguratio Flow. Xilinx University Program <http://www.xilinx.com/university/>

### **122018 - Síntesis Arquitectónica de Alto Nivel (SAAN)**

---

**Departamento:** Arquitectura de Computadores y Automática

**Duración:** Cuatrimestral (1º Cuatrimestre)

**Curso:** 2º

**Nivel:** Posgrado

**Tipo:** Optativa

**Profesores que imparten la asignatura:**

M<sup>a</sup> Carmen Molina Prego (coordinador);

**Créditos:** 6

**Horas/semana:** 2 horas en aula, 0 en laboratorio (1er cuatrimestre)

**Objetivos:** Dotar al alumno de conocimientos y destrezas en las técnicas modernas de construcción de sistemas hardware a partir de especificaciones arquitectónicas de muy alto nivel de abstracción.

**Conocimientos y destrezas que se requieren:** Estructura de computadores: nivel medio; Lenguaje de descripción de hardware (VHDL): nivel medio;

**Contenidos:** Problemas clásicos de la síntesis arquitectónica: planificación temporal de operaciones, asignación de recursos hardware. Interacción entre planificación y asignación de recursos. Soluciones integradas. Síntesis para bajo consumo de energía. Síntesis orientada a tolerancia de fallos. Técnicas para independizar la calidad de los resultados del estilo descriptivo. Métodos formales en síntesis. Herramientas profesionales de síntesis arquitectónica.

**Conocimientos y destrezas que se adquieren:** Principios y fases del diseño arquitectónico: nivel avanzado; Análisis de la calidad de los circuitos obtenidos: nivel medio; Manejo de herramientas de diseño arquitectónico y lógico: nivel elemental; Técnicas de optimización de circuitos RT: nivel avanzado;

**Idioma en que se imparte:** Español

**Exámenes:** Examen final en febrero y septiembre.

**Método de evaluación:** Valoración continua de la participación en los debates (35%), y realización de un trabajo individual (65%)

**Bibliografía:**

López, J.C., Hermida, R., Geisselhardt, W.; *Advanced Techniques for Embedded System Design and Test*; Kluwer Academic Publishers, 1998;

Gajski, D., Dutt, N., Wu, A., Lin, S.; *High-Level Synthesis. Introduction to Chip and System Design*; Kluwer Academic Publishers, Dordrecht, The Netherlands, 1992.;

IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems; *Revista mensual de la IEEE Circuits and Systems Society*; ;

IEEE Transactions on Very Large Scale Integration (VLSI) Systems; *Revista bi-mensual de la IEEE Computer Society*; ;

**Página web:**<http://www.dacya.ucm.es/mendias/sa/sa.html>

**122019 - Técnicas de Alto Rendimiento en el Diseño de Procesadores (TARDP)**

---

**Departamento:** Arquitectura de Computadores y Automática

**Duración:** Cuatrimestral (2º Cuatrimestre)      **Curso:** 2º      **Nivel:** Posgrado

**Tipo:** Optativa

**Profesores que imparten la asignatura:**

Francisco Tirado Fernández (coordinador);

**Créditos:** 6



**Horas/semana:** 2 horas en aula, 0 en laboratorio (2º cuatrimestre)

**Objetivos:** Revisar las técnicas actuales de diseño de procesadores, dando una visión integrada de las interdependencias entre la evolución de la tecnología y la arquitectura de estos procesadores integrados.

**Conocimientos y destrezas que se requieren:** Arquitectura de Computadores: nivel avanzado;

**Contenidos:**

Introducción: Tendencias tecnológicas, costo/rendimiento. Procesadores superescalares. Tratamiento de dependencias de control. Acceso a memoria, límites del modelo. Predicción de datos. Diseño de bajo consumo. Procesadores Multicore y Manicore. Arquitecturas Heterogéneas

**Conocimientos y destrezas que se adquieren:** Técnicas avanzadas de diseño de procesadores: nivel avanzado;

**Idioma en que se imparte:** Español

**Método docente:** Enseñanza presencial teórica y actividades de los alumnos

**Exámenes:** Examen final en junio y septiembre.

**Método de evaluación:** Realización de un trabajo personal y evaluación continua

**Bibliografía:**

- Computer Architecture: A Quantitative Approach; J.L. Hennessy, D.A. Patterson; 4º Ed, Morgan Kaufmann, 2007;
- Modern Processor Design; J.P. Shen, M. H. Lipasti ; McGraw Hill, 2005;
- Chip multiprocessor architecture: techniques to improve throughput and latency; K. Olukotun, O. Olukotun, L. Hammond, J. Laudon; Morgan & Claypool Publishers, 2007;
- Transactional memory; J. R. Larus, J. Larus, R. Rajwar; Morgan & Claypool Publishers, 2006;
- The Memory System: You Can't Avoid It, You Can't Ignore It, You Can't Fake It; B. Jacob; Morgan & Claypool Publishers, 2009;
- Memory Systems: Cache, DRAM, Disk; B. Jacob; M. Kaufmann, 2007;

**Página web:** <http://www.dacya.ucm.es/tardp/>

**122020 - Tecnologías Avanzadas de Redes y Telecomunicaciones (TART)**

**Departamento:** Ingeniería del Software e Inteligencia Artificial

**Duración:** Cuatrimestral (2º Cuatrimestre)

**Curso:** 2º

**Nivel:** Posgrado

**Tipo:** Optativa

**Profesores que imparten la asignatura:**

Luis J. García Villalba (coordinador)

**Créditos:** 6

**Horas/semana:** 2 horas en aula, 0 en laboratorio (2º cuatrimestre)

**Profesorado:**; Luis J. García Villalba.

**Objetivos:** Conocimientos de las tecnologías más avanzadas en el área de las redes de computadores y de telecomunicaciones, haciendo especial énfasis en los protocolos y en las arquitecturas utilizados.

**Conocimientos y destrezas que se requieren:** Conocimientos de redes de computadores: nivel medio;

**Contenidos:** 1. Tecnologías básicas de comunicaciones: Fundamentos de las comunicaciones. Redes y movilidad. Servicios de seguridad y gestión de identidad. 2. Redes Inalámbricas: Introducción a las redes inalámbricas multisalto. Clasificación. Tipos (MANET, WSN, VANET, VMN). Protocolos de comunicaciones para redes ad hoc. Encaminamiento en redes ad hoc. Autoconfiguración en redes ad hoc. Redes de sensores. Redes malladas. Redes VANET. 3. Seguridad y confianza en sistemas distribuidos: Seguridad en redes. Análisis de riesgos y gestión de vulnerabilidades. Gestión de servicios de seguridad. Servicios avanzados de seguridad. 4. Integración de Redes: Mecanismos de gestión de la movilidad en redes heterogéneas. Seguridad en redes móviles y entornos inalámbricos. Servicios de autenticación y autorización de entornos móviles y heterogéneos. Servicios en redes heterogéneas.

**Conocimientos y destrezas que se adquieren:** Conocer la evolución histórica de las tecnologías de red: nivel medio; Adquirir los conceptos básicos de teoría de la comunicación: nivel medio; Conocer las características de los principales medios de transmisión: nivel medio; Adquirir los conocimientos básicos sobre las distintas arquitecturas de redes: nivel medio;

**Idioma en que se imparte:** Español

**Método docente:** Exposiciones teóricas. Realización de trabajos tutorizados.

**Exámenes:** Examen final en junio y septiembre.

**Método de evaluación:** Se valorará la asistencia y la participación en las discusiones. Es obligatoria la realización de un trabajo individual o en grupo que deberá ser expuesto en clase.

#### **Bibliografía:**

Michael Barbeau and Evangelos Kranakis; *Principles of Ad Hoc Networking*; Wiley, 2007;

Andrea Goldsmith; *Wireless Communications*; Cambridge University Press, 2005;

Amitava Mukherjee, Somprakash Bandyopadhyay and Debashis Saha; *Location Management and Routing in Mobile Wireless Networks*; Artech House, 2003;

C. Siva Ram Murthy and B. S. Manoj; *Ad Hoc Wireless Networks: Architectures and Protocols*; Prentice Hall PTR, 2008;

**Página web:** <http://www.fdi.ucm.es/profesor/javiervg/>

### **122022 - Diseño de Sistemas Empotrados (DSE)**

---

**Tipo:** Optativa **Nivel:** Posgrado

**Curso:** Segundo curso **Cuatrimestre:** 1º Cuatrimestre

Créditos BOE: 6 Créditos ECTS: 6

**Horas/semana:** 2 horas en aula, 0 en laboratorio (1º cuatrimestre)

**Profesorado:** José Ignacio Gómez Pérez (coordinador). Christian Tenllado van der Reijden, José Ignacio Gómez Pérez

**Departamento:** Arquitectura de Computadores y Automática

**Objetivos:** El objetivo básico de este curso es que alumno obtenga una visión clara sobre el diseño de sistemas empuotrados, en los que parte de la funcionalidad se implementará en software y parte en hardware.

**Conocimientos y destrezas que se requieren:** Diseño de sistemas digitales: nivel avanzado;Estructura de computadores: nivel avanzado;Sistemas operativos: nivel medio; Programación en C: nivel medio;

**Contenidos:**

1. Introducción a sistemas empuotrados
2. Procesadores en sistemas empuotrados
3. Tipos de arquitecturas e ISA
4. Tendencias actuales en microarquitectura embedded
5. Representación de la información. Punto fijo y punto flotante
6. Sistema de memoria
7. Tecnologías e memoria y fuentes de consumo (DRAM, SRAM, ReRAM...)
8. Arquitecturas de memoria en sistmeas empuotrados: cache vs. scratchpad.
9. Gestión del sistema de memoria
10. Optimizaciones de código
11. Transformaciones de bucles
12. Optimizaciones para localidad
13. Paralelismo a nivel de datos
14. Asignación y planificación de tareas
15. Asignación de tareas a recursos heterogéneos
16. Planificación de tareas
17. Sistemas operativos de tiempo real

**Conocimientos y destrezas que se adquieren:** Codiseño hardware/software: nivel medio; Programación en entornos hetereogéneos: nivel medio;

**Idioma en que se imparte:** Español

**Método docente:** Las clases consistirán en exposiciones teóricas por parte de los profesores y discusiones con los alumnos de los artículos seleccionados para la lectura. Habrá sesiones de laboratorio para realizar prácticas propuestas en clase.

**Exámenes:** Examen final en febrero y septiembre.

**Método de evaluación:**

- 40% examen o proyecto final
- 40% prácticas propuestas en clase
- 20% presentación de artículos en clase

**Bibliografía:**

Wayne Wolf; Computers as components. Principles of embedded computing system design; Morgan Kaufmann 2001;

D. Gajski, F. Vahid, S. Narayan, J. Gong; *Specification and design of embedded systems*; Prentice Hall 1994;

Michael Wolfe; High performance compilers for parallel computing.; Addison-Wesley 1996 2001; ACM Transactions on Embedded Computing Systems; ;

**Página web:** Se utilizará el campus virtual

## 122021 - Control Inteligente (CI)

---

**Tipo:** Optativa **Nivel:** Posgrado

**Curso:** Segundo curso **Cuatrimestre:** 1º Cuatrimestre

Créditos BOE: 6 Créditos ECTS: 6

**Horas/semana:** 2 horas en aula, 0 en laboratorio (1º cuatrimestre)

**Profesorado:** Matilde Santos Peñas

**Departamento:** Arquitectura de Computadores y Automática

### **Objetivos:**

Bajo el título de Control Inteligente se agrupan una serie de técnicas avanzadas de Control, basadas en la aplicación de la Inteligencia Artificial al control de procesos. Se pretende que los alumnos conozcan estas técnicas y su ámbito de aplicación.

**Conocimientos y destrezas que se requieren:** Control;

### **Contenidos:**

1. Control Inteligente
2. Sistemas Expertos: Control Experto
3. Redes Neuronales aplicadas al Control
4. Control Basado en la Lógica Fuzzy (Borrosa)
5. Algoritmos Genéticos
6. Aprendizaje Automático y medidas de confianza en la predicción
7. Estrategias neurofuzzy para reconocimiento de patrones

**Conocimientos y destrezas que se adquieren:**

**Idioma en que se imparte:** Español

### **Método docente:**

Enseñanza presencial teórica. Enseñanza presencial de ejercicios y supuestos practices. realización de prácticas no tutorizadas. Comentario y discusión de artículos científicos.

### **Exámenes:**

### **Método de evaluación:**

Entrega de una breve memoria escrita y presentación oral individual de un trabajo de aplicación de una de las técnicas de Inteligencia Artificial estudiadas a un problema del área del control (identificación, control, modelado, etc.)

### **Bibliografía:**

- G. Pajares, M. Santos; Inteligencia Artificial e Ingeniería del Conocimiento; RA-MA, 2005
- A. E. Ruano, Intelligent Control using Intelligent Computational Techniques, IEE Control Series, 2005
- Zilouchian, A., Jamshidi, M., Intelligent Control Systems Using Soft Computing Methodologies, CRC Press, 2001
- Cai, Si-Xing., Intelligent Control and Intelligent Automation. Principles, techniques and applications. World Scientific, 1997

- King, R.E., Computational Intelligence in Control Engineering. Control Engineering Series, Marcel Dekker, N.Y., 1999
- Dutton, K., Thompson, S., Barraclough, B. The art of control engineering. Prentice Hall, 1997

### **Complementaria**

- Harris, C.J, Moore, C.G., Brown, M., Intelligent Control: aspects of fuzzy logic and neural networks. World Scientific, 1993
- Driankov, D. Helledorn, H, y Reinfrank, M., J. An introduction to Fuzzy Control, Springer Verlag, 1993.
- Hilera, J.R, Martínez, V.J., Redes neuronales artificiales. RA-MA, 1995 Lectures Notes in Computer Science

## **122023 - Optimización en el Control de Procesos (OCP)**

---

**Departamento:** Arquitectura de Computadores y Automática

**Duración:** Cuatrimestral (2º Cuatrimestre)      **Curso:** 2º      **Nivel:** Posgrado

**Tipo:** Optativa

### **Profesores que imparten la asignatura:**

Jesús Manuel de la Cruz García (coordinador)

**Créditos:** 6

**Horas/semana:** 2 horas en aula, 0 en laboratorio (2º cuatrimestre)

**Objetivos:** Los problemas de optimización surgen en prácticamente todas las áreas de las ciencias aplicadas y de las ingenierías. Un problema de optimización es un modelo matemático que contiene variables, parámetros, relaciones matemáticas que ligan y restringen los posibles valores de aquellos, y uno o más funciones objetivo, también denominadas funciones de coste o índices de rendimiento. El objetivo es encontrar aquellas variables que dan el mejor valor de la función objetivo a la vez que cumplen todas las relaciones del modelo. La asignatura presenta los fundamentos de la optimización haciendo especial hincapié en los aspectos prácticos y algorítmicos, y su uso en el diseño de controladores para sistemas. Uso de Matlab para optimización. Lenguaje AMPL para modelar problemas de optimización

**Conocimientos y destrezas que se requieren:** Cálculo y Álgebra

### **Contenidos:**

1. INTRODUCCIÓN. Objetivo de la optimización. Breve reseña histórica. Áreas de aplicación. Algoritmos iterativos y convergencia.
2. MÉTODOS DE OPTIMIZACIÓN SIN RESTRICCIONES Introducción. Métodos Cuadráticos. Métodos para funciones multivaluadas. Método del gradiente conjugado. Método de Fletcher-Reeves. Método de Newton. Métodos Quasi-Newton. Optimización sin derivadas. Métodos para suma de cuadrados. Ecuaciones no lineales.
3. PROGRAMACIÓN LINEAL. Definiciones y Resultados Fundamentales. Método del Simplex. Método del simplex revisado. Dualidad y análisis de sensibilidad.
4. MÉTODOS DE OPTIMIZACIÓN CON RESTRICCIONES Condiciones para la optimización con restricciones. Métodos primal. Métodos de barrera. Métodos duales. Métodos de Lagrange.

5. OPTIMIZACIÓN MIXTA ENTERA Optimización lineal mixta entera. Método branch and bound. Optimización no lineal mixta entera. Descomposición generalizada de Benders. Aproximaciones externas.

6. INTRODUCCIÓN A LOS MÉTODOS DE OPTIMIZACIÓN HEURÍSTICA. Algoritmos genéticos. "Simulated annealing" (temple simulado). Búsqueda tabú.

7. OPTIMIZACIÓN EN EL CONTROL. Sintonía de controladores mediante optimización. Aplicaciones.

**Conocimientos y destrezas que se adquieren:** El concepto de Optimización: nivel avanzado; El concepto e importancia de la función objetivo: nivel avanzado; Optimización según el tipo de sistema: nivel avanzado; Distintos métodos de optimización de procesos industriales: nivel avanzado; Optimización Heurística. Algoritmos genéticos: nivel medio.

**Idioma en que se imparte:** Español

**Exámenes:** Examen final en junio y septiembre.

**Método de evaluación:** Trabajos realizados durante el curso.

**Bibliografía:**

Linear and Nonlinear programming

D.G. Luenberger. 2º Edition. Addison Wesley, 1989

Practical Methods of Optimization. R. Fletcher. 2º E. John Wiley & Sons, 1987.

Nonlinear Programming.

D.P. Bertsekas. 2º E. Athena Scientific, 1999.

Practical Optimization.

P.E. Gill. W. Murray and M. Wright. Academic Press, 1981.

Integer and Combinatorial Optimization.

G.L. Nemhauser, L.A. Wolsey. Wiley-Interscience 1999.

Combinatorial Optimization

C.H. Papadimitriou, K. Steiglitz. Dover, 1998.

Métodos Heurísticos

Intelligent Optimisation Techniques. Genetic Algorithms, Tabu Search, Simulated Annealing and Neural Networks. D.T. Pham, D. Karaboga. Springer, 2000.

---

## **122024 - Percepción Visual Artificial (PVA)**

**Departamento:** Ingeniería del Software e Inteligencia Artificial

**Duración:** Cuatrimestral (1º Cuatrimestre)

**Curso:** 2º

**Nivel:** Posgrado

**Tipo:** Optativa

**Profesores que imparten la asignatura:**

Gonzalo Pajares Martinsanz (coordinador)  
María Guijarro Mata-García

**Créditos:** 6

**Horas/semana:** 2 horas en aula, 0 en laboratorio (1er cuatrimestre)

**Objetivos:** Comprender el concepto, mecanismo y las etapas del proceso de percepción visual artificial, también conocido como Visión por Computador. Estudiar y aplicar las técnicas de procesamiento de bajo y alto nivel en visión artificial

**Conocimientos y destrezas que se requieren:** Tratamiento de matrices: nivel elemental; Significado geométrico y físico de la derivación: nivel elemental; Especificación de algoritmos iterativos y recursivos: nivel elemental; Variables aleatorias y estadísticas: nivel elemental; Matlab: nivel elemental;

**Contenidos:** 1) Introducción a la percepción visual artificial: recepción de imágenes de la escena y actuación en el medio. Captura de imágenes procedentes del entorno 2) Procesado de bajo nivel: 2.1) Preprocesado: eliminación de ruido, suavizado de imágenes, realzado de imágenes 2.2) Extracción de características: puntos de interés, bordes, regiones 2.3) Obtención de propiedades de las características: área, tamaño, excentricidad, color, etc. 3) Percepción: procesado de alto nivel. 3.1) Reconocimiento: aprendizaje supervisado y no supervisado, métodos de clasificación. 3.2) Obtención de la estructura tridimensional de la escena: visión estereoscópica 3.3) Detección del movimiento: obtención del flujo óptico 3.4) Fusión de Imágenes: multiresolución.

**Conocimientos y destrezas que se adquieren:** Procesamiento de imágenes digitales: nivel avanzado; Visión por computador: nivel avanzado; Reconocimiento de formas: nivel avanzado; Algoritmos de Aprendizaje Automático: nivel avanzado; Capacidad de comunicación oral y escrita en español e inglés; Capacidad de resolución cuantitativa de problemas: nivel avanzado; Capacidad de análisis; Capacidad de comprensión de documentación técnica: nivel avanzado; Capacidad de aprendizaje autónomo: nivel avanzado; Capacidad de obtención autónoma de información: nivel medio;

**Idioma en que se imparte:** Español

**Exámenes:** Examen final en febrero y septiembre.

**Método de evaluación:** Convocatoria de Junio y Septiembre: Es obligatoria la realización individual de al menos un trabajo (80%). Es necesaria la defensa oral de un trabajo (20%)

### **Bibliografía:**

Gonzalo Pajares y Jesús M. de la Cruz; *Visión por computador: imágenes digitales y aplicaciones*; 2ª edición, RA-MA, Madrid, 2007;

Gonzalo Pajares, J.M. de la Cruz; *Ejercicios Resueltos de Visión por Computador*; 1ª edición, RA-MA, Madrid, 2007;

R.O. Duda, P.E. Hart and D.G. Stork; *Pattern Classification*; Wiley, New York, 2001;

R.C. Gonzalez and R.E. Woods; *Digital Image Processing*; Prentice Hall, 2002;

**Página web:** <http://www-2.cs.cmu.edu/~cil/vision.html>; <http://www.dacya.ucm.es/Gonzalo/>

## **122025 - Redes Neuronales y sus Aplicaciones (RNA)**

**Departamento:** Arquitectura de Computadores y Automática

**Duración:** Cuatrimestral (2º Cuatrimestre)

**Curso:** 2º

**Nivel:** Posgrado

**Tipo:** Optativa

**Profesores que imparten la asignatura:**

José Antonio López Orozco (coordinador)  
Juan Jiménez Castellanos

**Créditos:** 6**Horas/semana:** 2 horas en aula, 0 en laboratorio (2º cuatrimestre)**Objetivos:** En este curso se estudian los fundamentos de los principales tipos de redes neuronales (feed-forward, redes dinámicas, competitivas, ...) analizando su estructura, funcionamiento, finalidad y características. Se completará con el estudio de aplicaciones donde se han utilizado estas redes.**Conocimientos y destrezas que se requieren:** Matlab: nivel medio; Programación: nivel elemental;**Contenidos:** En este curso se estudian los fundamentos de los principales tipos de redes neuronales (feed-forward, redes dinámicas, competitivas, ...) analizando su estructura, funcionamiento, finalidad y características, de modo que el estudiante sea capaz de entender y analizar otros tipos de redes. De este modo, se introduce en el campo de las redes neuronales y se le dota de la base necesaria para diseñar e implementar su propia red neuronal para aplicarla en su investigación. Se revisan las características comunes de las redes y las distintas clasificaciones y utilidades de éstas. Se estudiarán en detalle el Perceptrón, Adaline, el Perceptrón Multicapa y la Madelaine, el método de aprendizaje BackPropagation, Redes dinámicas como la red dinámica de Hopfield, y redes competitivas como la red ART y la SOM. También se analizan otras redes como las redes probabilísticas, las RBF... Para completar su formación y comprender mejor el funcionamiento de estas redes neuronales se estudian distintos ejemplos y aplicaciones donde se han utilizado los distintos tipos de redes analizando los resultados. Para la mejor comprensión del funcionamiento de cada modelo de red se realizarán diversas prácticas en Matlab.**Conocimientos y destrezas que se adquieren:** Programación de tipos de redes a problemas concretos: nivel avanzado; Planificación, especificación y diseño de aplicaciones: nivel medio; Evaluación de la calidad de una aplicación: nivel medio; Exposición y defensa de resultados y conocimientos: nivel medio;**Idioma en que se imparte:** Español**Exámenes:** Examen final en junio y septiembre.**Método de evaluación:** Asistencia y participación en clase. Realización individual de prácticas. Defensa individual de un trabajo o proyecto.**Bibliografía:**

Hilera J.R. Y Martínez V.J.; Redes Neuronales Artificiales. Fundamentos, modelos y aplicaciones; RA-MA., 1995;

Patterson D.W. ; Artificial Neural Networks. Theory and Applications; Prentice Hall, 1996;

Haykin, S; Neural Networks - A Comprehensive Foundatio; Macmillan, New York, 1994;

Durante el curso, además, se irán proponiendo una serie de lecturas complementarias.

---

**122026 - Análisis y Transformación de Programas (ATP)****Asignatura:** 122026 - Análisis y Transformación de Programas (ATP)**Departamento:** Sistemas Informáticos y Computación



Duración: Anual                      Curso: 2º                      Nivel: Posgrado

**Tipo:** Optativa

**Profesores que imparten la asignatura:**

Ricardo Peña Mari (coordinador)  
Clara Mª Segura Díaz  
Elvira María Albert Albiol

**Créditos:** 7,5

**Horas/semana:** 2 horas en aula, 0 en laboratorio

**Departamento:** Sistemas Informáticos y Computación

**Objetivos:** Análisis estático de programas; optimización y transformación de programas.

**Conocimientos y destrezas que se requieren:** Derivación de algoritmos recursivos: nivel medio; Procesadores de lenguaje: nivel avanzado;

**Contenidos:** Optimización y transformación de programas. Especialización de programas y evaluación parcial. Introducción al análisis estático: ejemplos, aplicaciones y técnicas. Análisis basado en el flujo de datos. Análisis basado en el flujo de control. Interpretación abstracta. Análisis basado en tipos.

**Conocimientos y destrezas que se adquieren:** Transformación de programas: nivel elemental; Optimización de programas: nivel elemental; Técnicas de análisis de programas: nivel medio; Aplicaciones de los análisis de programas: nivel medio;

**Idioma en que se imparte:** Español

**Método docente:** Exposiciones teóricas y discusión de artículos

**Exámenes:** Examen final en junio y septiembre.

**Método de evaluación:** Asistencia obligatoria a las clases. Entrega de ejercicios. Presentación de artículos.

**Bibliografía:**

A. W. Appel; *Modern Compiler Implementation in ML*; Cambridge University Press, 1998;

D. Jones and C.K. Gomard and P. Sestoft; *Partial Evaluation and Automatic Program Generation*; Prentice Hall, 1993;

F. Nielson, H. R. Nielson and C. Hankin; *Principles of Program Analysis*; Springer-Verlag, 1999;

A. L. M. Santos; *Compilation by Transformation in Non-Strict Functional Languages*; Phd. Thesis. Department of Computer Science, University of Glasgow, 1995;

**122027 - Aplicación de Principios Económicos al Desarrollo de Sistemas Computacionales (APEDS)**

---

**Departamento:** Sistemas Informáticos y Computación.

**Duración:** Cuatrimestral (1º Cuatrimestre)                      **Curso:** 2º                      **Nivel:** Posgrado

**Tipo:** Optativa

**Profesores que imparten la asignatura:**

Fernando Rubio (coordinador)  
Ismael Rodríguez Laguna  
Manuel Núñez

**Créditos:** 6

**Horas/semana:** 2 horas en aula, 0 en laboratorio (1er cuatrimestre)

**Departamento:** Sistemas Informáticos y Computación

**Objetivos:** Uso de principios económicos en el reparto de recursos computacionales en entornos distribuidos, en el comercio y subastas electrónicas, y en programación en general.

**Contenidos:** - Introducción a conceptos básicos de Microeconomía: Funciones de utilidad, Teoría general del equilibrio, Equilibrio de Pareto, etc. - Utilización de principios económicos para distribuir recursos computacionales en entornos distribuidos de manera eficiente. Algoritmos distribuidos óptimos y subóptimos para la distribución de recursos entre procesos. - Aspectos económicos del comercio electrónico y las subastas electrónicas. Algoritmos para la distribución computacional de recursos entre usuarios humanos. - Otras aplicaciones de la Economía a la Informática. La Economía como paradigma de programación.

**Conocimientos y destrezas que se adquieren:** Algoritmos económicos distribuidos de ubicación de recursos: nivel medio; Algoritmos de comercio y subastas electrónicas: nivel medio; Aplicación de la Economía a la programación: nivel medio;

**Idioma en que se imparte:** Español

**Método docente:** Exposiciones teóricas y discusión de artículos.

**Exámenes:** Examen final en febrero y septiembre.

**Método de evaluación:** Presentación de artículos.

**Bibliografía:**

A. Chavez y P. Maes; Kasbah: An agent marketplace for buying and selling goods; PAAM96, páginas 75-90, 1996;

T. Sandholm; Algorithm for optimal winner determination in combinatorial auctions; Artificial Intelligence, 135(1-2), páginas 1-54, 2002;

I. Rodríguez; Especificación de sistemas concurrentes usando conceptos de teoría económica; Universidad Complutense, 2004;

A. Mas-Colell, M.D. Whinston y J.R. Green; *Microeconomic Theory*; Oxford University Press, 1995;

## **122028 - Especificación y Validación del Software (EVS)**

---

**Departamento:** Sistemas Informáticos y Computación

Duración: Anual                      Curso: 2º                      Nivel: Posgrado

**Tipo:** Optativa

**Profesores que imparten la asignatura:**

Narciso Martí Oliet (coordinador)  
Manuel Clavel  
Miguel Palomino  
Alberto Verdejo

**Créditos:** 7,5

**Horas/semana:** 2 horas en aula, 0 en laboratorio

**Departamento:** Sistemas Informáticos y Computación

**Objetivos:** El objetivo del curso es la presentación de diferentes métodos formales para mejorar la fiabilidad del software, tanto de programas secuenciales como concurrentes, destacando el uso de especificaciones algebraicas, lógica de primer orden, razonamiento inductivo y demostración automática, lógicas temporales para sistemas concurrentes y aplicaciones del método de "model checking".

**Conocimientos y destrezas que se requieren:** Matemática discreta: teoría elemental de conjuntos, inducción: nivel elemental; Lógica: lógica proposicional y lógica de primer orden: nivel medio; Programación: programación declarativa, tipos abstractos de datos: nivel medio;

**Contenidos:** Lógica ecuacional y lógica de primer orden. Especificaciones algebraicas de tipos de datos. Especificación y verificación de programas secuenciales. Lógicas temporales. Especificación de sistemas orientados a objetos. Especificación y verificación de sistemas concurrentes. Abstracción de sistemas y "model checking".

**Conocimientos y destrezas que se adquieren:** Métodos generales de razonamiento sobre propiedades de sistemas: nivel avanzado; Herramientas informáticas para la validación y verificación de sistemas: nivel avanzado;

**Idioma en que se imparte:** Español

**Método docente:** Exposiciones teóricas, discusión de artículos y presentación de trabajos

**Exámenes:** Examen final en junio y septiembre.

**Método de evaluación:** Asistencia a clase (20%), participación en discusiones y resolución de problemas (20%), un trabajo individual (60%)

**Bibliografía:**

E. Astesiano, H.-J. Kreowski y B. Krieg-Bruckner; *Algebraic Foundations of Systems Specification*; Springer, 1999;

E. M. Clarke, O. Grumberg y D. A. Peled; *Model Checking*; The MIT Press, 2000;

M Clavel, F Durán, S Eker, P Lincoln, N Martí-Oliet, J Meseguer y C Talcott; *All About Maude: A High-Performance Logical Framework.*; Springer, 2007;

M Clavel, F Durán, S Eker, P Lincoln, N Martí-Oliet, J Meseguer y C Talcott; *Maude Manual (Version 2.6)*; maude.cs.uiuc.edu, 2011;

**Página web:** maude.cs.uiuc.edu

---

**122029 - Modelado de Sólidos y Técnicas de Visualización (MSTV)**

**Departamento:** Sistemas Informáticos y Computación

Duración: Anual                      Curso: 2º                      Nivel: Posgrado

**Tipo:** Optativa

**Profesores que imparten la asignatura:**

Pedro J. Martín de la Calle(coordinador)  
Antonio Gavilanes Franco

José Alberto Verdejo López

**Créditos:** 7,5

**Horas/semana:** 2 horas en aula, 0 en laboratorio

**Departamento:** Sistemas Informáticos y Computación

**Objetivos:** Técnicas avanzadas en informática gráfica

**Conocimientos y destrezas que se requieren:** Informática gráfica: nivel medio;

**Contenidos:** Modelos de iluminación global Estructuras de datos para la descomposición espacial Fundamentos físicos para la animación Modelado de la escena usando Java 3D Programación de GPUs.

**Conocimientos y destrezas que se adquieren:** Informática gráfica: nivel avanzado;

**Idioma en que se imparte:** Español

**Método docente:** Exposición de la materia fundamental del curso. Presentación de trabajos.

**Exámenes:** Examen final en junio y septiembre.

**Método de evaluación:** Realización individual de trabajos no tutorizados. La asistencia a clase es obligatoria.

**Bibliografía:**

P. Shirley; *Realistic Ray Tracing*; A K Peters, 2000;

D. B. Kirk, W. W. Hwu; *Programming massively parallel processors: a hands-on approach*; Morgan Kaufmann, 2010;

R. Parent; *Computer Animation Algorithms and Techniques*; Morgan Kaufmann Publishers, 2002;

E. Langetepe; G. Zachmann; *Geometric data structures for computer graphics*; A K Peters, 2006;

## **122030 - Modelos Abstractos de Cómputo (MAC)**

---

**Departamento:** Sistemas Informáticos y Computación

**Duración:** Cuatrimestral (1º Cuatrimestre)      **Curso:** 2º      **Nivel:** Posgrado

**Tipo:** Optativa

**Profesores que imparten la asignatura:**

Francisco Javier López Fraguas (coordinador)

**Créditos:** 4,5

**Horas/semana:** 2 horas en aula, 0 en laboratorio (1er cuatrimestre)

**Objetivos:** Presentar las técnicas de reescritura y el lambda-cálculo como modelos de cómputo para los lenguajes de programación, en particular de programación declarativa.

**Conocimientos y destrezas que se requieren:** Matemática discreta: nivel medio; Programación funcional: nivel medio; Lógica matemática: nivel medio; Programación lógica: nivel medio;

**Contenidos:** - Sistemas abstractos de reducción - Sistemas de reescritura de términos - Reescritura condicional - Estrechamiento - Aplicación a la programación lógico-funcional - Lambda-cálculo - Aplicación a la programación funcional

**Conocimientos y destrezas que se adquieren:** Técnicas de reescritura: nivel medio; Lambda-cálculo: nivel medio; Programación lógico-funcional: nivel elemental;

**Idioma en que se imparte:** Español

**Método docente:** Enseñanza presencial teórica y de ejercicios, tutorías colectivas con discusión de ejercicios y lecturas sugeridas y presentación de trabajos.

**Exámenes:** Examen final en febrero y septiembre.

**Método de evaluación:** Evaluación continua que tendrá en cuenta: asistencia y participación en clase, presentación de ejercicios y preparación de algún tema.

### **Bibliografía:**

F. Baader, T. Nipkow ; *Term rewriting and all that*; Cambridge Univ. Press, 1998;

J.W. Klop; Term rewriting systems. In Handbook of Logic in Computer Science, Vol. 2.; Oxford Univ. Press, 1992;

Chris Hankin; An Introduction to Lambda Calculi for Computer Scientists.; King College Publications 2004;

H.P. Barendregt; Functional Programming and Lambda Calculus. In Handbook of Theor. Comp. Science ; Elsevier 1990;

## **122031 - Modelos de Sistemas Concurrentes y Distribuidos (MSCD)**

---

**Departamento:** Sistemas Informáticos y Computación

Duración: Anual                      Curso: 2º                      Nivel: Posgrado

**Tipo:** Optativa

### **Profesores que imparten la asignatura:**

David de Frutos Escrig (coordinador)  
Yolanda Ortega Mallén

**Créditos:** 7,5

**Horas/semana:** 2 horas en aula, 0 en laboratorio

**Objetivos:** Se presentarán los modelos formales para la especificación y verificación de sistemas concurrentes y distribuidos, englobados en dos grandes familias: las Algebras de Procesos y las Redes de Petri.

**Conocimientos y destrezas que se requieren:** Inducción: nivel avanzado; Programación concurrente: nivel elemental;

**Contenidos:** Problemas inherentes de la programación concurrente y de los sistemas distribuidos. Modelos formales para facilitar su desarrollo y análisis. - CCS: Sintaxis; Semántica de bisimulación; Especificaciones e implementaciones: pruebas de corrección. - CSP: Sintaxis; Semántica de fallos y divergencias; Caracterización axiomática; Pruebas de equivalencia. - Redes de Petri: Estructura y semántica; Métodos de análisis: matriciales y estructurales; Simulación; Redes de alto nivel. Sistemas en tiempo real: extensiones temporizadas de los modelos formales.

**Conocimientos y destrezas que se adquieren:** Desarrollo de sistemas concurrentes y distribuidos con modelos formales: nivel medio; Semántica de los modelos para la programación concurrente y distribuida: nivel medio; Especificación y prueba de propiedades de sistemas concurrentes y distribuidos: nivel medio; Modelización de sistemas en tiempo real: nivel elemental;

**Idioma en que se imparte:** Español

**Método docente:** Exposiciones teóricas comentando la bibliografía básica. Discusión de artículos especializados. Presentación de trabajos y resolución de ejercicios propuestos

**Exámenes:** Examen final en junio y septiembre.

**Método de evaluación:** Asistencia activa a las clases (30%) Resolución de ejercicios propuestos (30%) Lectura de artículos y presentación de alguno de ellos (40%)

#### **Bibliografía:**

J.C.M. BAETEN, T. BASTEN, M.A. RENIERS; *Process Algebra: Equational Theories of Communicating Processes*; Cambridge University Press, Cambridge, UK, 2010;

R. MILNER; *Communication and Concurrency*; Prentice Hall,1989.;

W. REISIG, G. ROZENBERG (Eds.) ; *Lectures on Petri Nets: Basic Models and Applications (LNCS 1491 y 1492 )*; Springer Verlag,1998;

L. ACETO, A. INGOLFSDOTTIR, K. LARSEN, J. SRBA; *Reactive Systems: Modelling, Specification and Verification*; Cambridge University Press,2007;

### **122032 - Requisitos Software (RS)**

---

**Departamento:** Ingeniería del Software e Inteligencia Artificial

**Duración:** Cuatrimestral (1º Cuatrimestre)      **Curso:** 2º      **Nivel:** Posgrado

**Tipo:** Optativa

#### **Profesores que imparten la asignatura:**

Antonio Navarro Martín (coordinador)  
Iván Martínez  
Pablo Moreno  
Rubén Fuentes

**Créditos:** 4,5

**Horas/semana:** 2 horas en aula, 0 en laboratorio (1er cuatrimestre)

**Objetivos:** Conocer el estado actual de la Ingeniería de Requisitos (IR) en ingeniería web. Conocer las técnicas de arquitectura dirigida por modelos y su aplicación a la IR. Aprender a gestionar la producción y mantenimiento de lenguajes software y su relación con la IR

**Contenidos:** 1. Ingeniería de requisitos en ingeniería web. 2. Arquitectura dirigida por modelos y su aplicación a la ingeniería de requisitos. 3. Producción y mantenimiento de lenguajes software y su aplicación a la ingeniería de requisitos.

**Idioma en que se imparte:** Español

**Exámenes:** Examen final en febrero y septiembre.

**Método de evaluación:** Asistencia a clase y participación en la discusión de cada uno de los temas. Realización de trabajos de análisis bibliográfico y/o desarrollos prácticos. Presentación de dichos trabajos en forma oral y escrita, evaluando especialmente su capacidad para situar en contexto y profundizar en el tema investigado. Presentación de un trabajo final que deberá tener la forma de una prepublicación científica.

### **Bibliografía:**

Lowe, D.; *Web System Requirements: an Overview*; Requirements Engineering 8 (2), 102-113, 2003;

Object Management Group; *Unified Modeling Language Infrastructure V2.1.2*; Object Management Group, 2007;

Kepple, A.; *Software Language Engineering*; Addison-Wesley, 2008;

**Página web:** Campus Virtual (<https://www.ucm.es/info/uatd/CVUCM/index.php>)

## **122033 - Seguridad de Computadores (SC)**

---

**Departamento:** Arquitectura de Computadores y Automática

**Tipo:** Optativa **Nivel:** Posgrado

**Curso:** Segundo curso **Cuatrimestre:** 2º Cuatrimestre

### **Profesores que imparten la asignatura:**

Imaña Pascual, José Luis  
Botella Juan, Guillermo

**Coordinador:** José Luís Imaña Pascual

Créditos BOE: 6 Créditos ECTS: 6

**Horas/semana:** 2 horas en aula, 0 en laboratorio (2º Cuatrimestre).

**Objetivos:** Introducción a la criptografía. Introducción a la seguridad informática. Revisión de los diferentes algoritmos y sistemas criptográficos. Estudio de la aritmética modular empleada en criptografía. Diseño e implementación de circuitos aritméticos eficientes utilizados en aplicaciones criptográficas. Ejemplos de uso de la criptografía.

Breve descriptor:

Fundamentos de criptografía. Fundamentos de seguridad informática. Sistemas de cifrado. Implementaciones de circuitos aritméticos utilizados en criptografía. Aplicaciones de la criptografía.

Requisitos

Criptografía: nivel básico; Aritmética modular: nivel elemental; Arquitectura y Estructura de computadores: nivel medio.

**Conocimientos y destrezas que se requieren:** Criptografía: nivel medio; Aritmética modular: nivel elemental; Arquitectura y Estructura de computadores: nivel medio;

### **Contenidos:**

Introducción. Fundamentos de criptografía. Fundamentos de seguridad informática. Sistemas de cifrado. Fundamentos de aritmética modular. Aritmética sobre cuerpos binarios.

Implementación eficiente de operaciones aritméticas con aplicaciones en criptografía. Aplicaciones de la criptografía.

**Conocimientos y destrezas que se adquieren:** Aritmética modular: nivel medio; Implementaciones hardware/software de circuitos aritméticos utilizados en cripto: nivel avanzado; Diseño básico de planes de seguridad;

**Idioma en que se imparte:** Español

**Método docente:** Enseñanza presencial teórica y presentación de trabajos.

**Método de evaluación:** Se valorará la asistencia a clase. Realización de un trabajo individual. Obligatoria su defensa individual (oral/escrita)

Enseñanza presencial teórica (90%). Presentación de trabajos (10%).

#### **Bibliografía:**

A.J. Menezes, P.C. van Oorschot, S.A. Vanstone; Handbook of Applied Cryptography. CRC Press, 1997.

A.J. Menezes, I. Blake, X. Gao, R. Mullin, S. Vanstone, T. Yaghoobian; Applications of Finite Fields. Kluwer Academic Publisher, 1993.

J.P. Deschamps, J.L. Imaña, G.D. Sutter; Hardware Implementation of Finite-Field Arithmetic. McGraw-Hill, 2009.

J. Ramío; Libro Electrónico de Seguridad Informática y Criptografía. Universidad Politécnica de Madrid, 2006.

### **122034 - Aprendizaje Automático (AA)**

---

**Departamento:** Ingeniería del Software e Inteligencia Artificial

**Duración:** Cuatrimestral (1º Cuatrimestre)      **Curso:** 2º      **Nivel:** Posgrado

**Tipo:** Optativa

#### **Profesores que imparten la asignatura:**

Pedro Antonio González Calero (coordinador)  
M<sup>a</sup> Belén Díaz Agudo  
Mercedes Gómez Albarrán

**Créditos:** 6

**Horas/semana:** 2 horas en aula, 0 en laboratorio (1er cuatrimestre)

**Objetivos:** Capacitar al estudiante para llevar a cabo investigación en razonamiento basado en casos dentro del marco del aprendizaje automático

**Conocimientos y destrezas que se requieren:** Inteligencia Artificial: nivel medio; Estadística: nivel elemental;

**Contenidos:** 1. Conceptos básicos de aprendizaje automático 2. Razonamiento basado en casos 3. Aprendizaje automático y razonamiento basado en casos 4. Aplicaciones

**Conocimientos y destrezas que se adquieren:** principios generales del aprendizaje automático: nivel medio; estado del arte en razonamiento basado en casos: nivel avanzado; capacidad de comunicación oral y escrita en español: nivel medio; capacidad de análisis y elaboración de documentos científicos: nivel medio;



**Idioma en que se imparte:** Español

**Método docente:** Enseñanza presencial teórica. Realización de trabajos tutorizados en grupo.

**Exámenes:** Examen final en febrero y septiembre.

**Método de evaluación:** Es obligatoria la realización de un trabajo en grupo. Es obligatoria la defensa en grupo (oral y escrita) de un trabajo en dos partes. El 30% corresponde a la parte teórica del trabajo. El 70% corresponde a una segunda parte teórica o práctica.

**Bibliografía:**

Tom M. Mitchell; *Machine Learning*; McGraw-Hill, 1997;

Leake, D. B., editor; *Case-Based Reasoning: Experiences, Lessons, and Future Directions*; Menlo Park, CA: AAAI Press/MIT Press 1996;

Ian H. Witten, Eibe Frank; *Data Mining: Practical Machine Learning Tools and Techniques with Java Implement*; Morgan Kaufmann, 1999;

Mario Lenz, Brigitte Bartsch-Spörl, Hans-Dieter Burkhard, Stefan Wess; *Case-Bases Reasoning Technology*; Springer, 1998;

**Página web:** Espacio habilitado en el Campus Virtual de la UCM

---

## 122035 - Agentes Inteligentes (AI)

**Departamento:** Ingeniería del Software e Inteligencia Artificial

**Duración:** Cuatrimestre (1º Cuatrimestre)      **Curso:** 2º      **Nivel:** Posgrado

**Tipo:** Optativa

**Profesores que imparten la asignatura:**

Jorge Jesús Gómez Sanz (coordinador)  
Rubén Fuentes Fernández

**Créditos:** 6

**Horas/semana:** 2 horas en aula, 0 en laboratorio (1er cuatrimestre)

**Objetivos:** Comprender la problemática del diseño de sistemas multi-agente

**Conocimientos y destrezas que se requieren:** Programación: nivel elemental; Inteligencia Artificial: nivel elemental; Ingeniería del software: nivel elemental;

**Contenidos:** 1. Conceptos generales. Sistemas de agente único. Sistemas Multi-agente. Características básicas de los agentes. 2. Funcionamiento interno de los agentes y los sistemas multi-agente. Estudio de agentes en diferentes dominios. 3. Técnicas para el diseño de agentes. Metodologías para el diseño de sistemas multi-agente. Metodología INGENIAS

**Conocimientos y destrezas que se adquieren:** Técnicas para especificar Sistemas Multi-Agente: nivel medio;

**Idioma en que se imparte:** Español

**Método docente:** Clases teóricas presenciales, videos y ejemplos de desarrollo de sistemas multi-agente

**Exámenes:** Examen final en febrero y septiembre.

**Método de evaluación:** Es obligatorio asistir a todas las clases. A lo largo del curso se pedirá a los participantes que colaboren en el seguimiento de su actividad dentro de la asignatura. Para ello se dispondrán de aplicaciones informáticas que tendrán que usarse de forma obligatoria. La nota vendrá determinada por trabajos de naturaleza eliminatoria (hay que aprobarlos para aprobar la asignatura) para aquellos que hayan asistido a todas las clases. Los trabajos, a elección de los alumnos, consistirán en la realización de un estudio crítico sobre alguno de los aspectos considerados en la asignatura usando artículos de investigación y libros de referencia. La elección del tema requerirá el visto bueno de los profesores. De forma alternativa, los alumnos podrán optar por hacer un desarrollo software sencillo con el entorno de desarrollo INGENIAS Development Kit. El tema del desarrollo también requerirá el visto bueno del profesor. Los desarrollos primarán sobre los trabajos en la nota por su complejidad, salvo excepciones.

### **Bibliografía:**

Ana Mas; Agentes software y sistemas multiagente. Conceptos, arquitecturas y aplicaciones; Pearson, 2004;

Brian Henderson-Sellers (ed.) y Paolo Giorgini (ed.); *Agent-Oriented Methodologies*; Idea Group. 2005;

Gerhard Weiss (ed.); *Multiagent systems*; The MIT Press, 1999;

M. Klusch (Ed.); Intelligent Information Agents. Agent-Based Information Discovery and Management; Springer Verlag, 1999;

**Página web:** <http://www.fdi.ucm.es/profesor/jpavon/doctorado/>

## **122036 - Deducción Automática (DA)**

---

**Departamento:** Sistemas Informáticos y Computación

**Duración:** Cuatrimestral (1º Cuatrimestre)      **Curso:** 2º      **Nivel:** Posgrado

**Tipo:** Optativa

### **Profesores que imparten la asignatura:**

M<sup>a</sup> de las Mercedes García Merayo (coordinadora)

Manuel Núñez García

Ismael Rodríguez Laguna

**Créditos:** 4,5

**Horas/semana:** 2 horas en aula, 0 en laboratorio (1er cuatrimestre)

**Objetivos:** Estudio de técnicas que permiten deducir de forma (semi-)automática la corrección de sistemas.

**Conocimientos y destrezas que se requieren:** Teoría de Autómatas; Matemática Discreta;

**Contenidos:** 1. Introducción a los métodos formales para la representación de sistemas. 2. Deducción automática de contraejemplos: Model checking. 3. Deducción automática de tests: Generación de tests a partir de requerimientos; generación de tests a partir de modelos con estados finitos.

**Conocimientos y destrezas que se adquieren:** Especificación formal de sistemas; Conceptos básicos de model checking; Métodos formales de testing; Algoritmos de derivación de tests;

**Idioma en que se imparte:** Español

**Método docente:** Enseñanza presencial teórica. Realización individual de trabajos no tutorizados.

**Exámenes:** Examen final en febrero y septiembre.

**Método de evaluación:** Convocatoria de Febrero: Realización obligatoria y defensa oral individual de un trabajo propuesto (80%). Participación en clase (20%). Convocatoria de Septiembre: Realización obligatoria y defensa oral individual de un trabajo propuesto (100%).

**Bibliografía:**

Aditya P. Mathur; *Foundations of Software Testing*; Addison-Wesley/Prentice Hall, 2008;

Mark Utting and Bruno Legeard; *Practical Model-Based Testing: A Tools Approach* ; Morgan-Kaufmann, 2007;

Edmund M. Clarke, Jr., Orna Grumberg and Doron A. Peled; *Model Checking*; MIT Press, 1999;

**122037 - E-Learning (EL)**

---

**Departamento:** Ingeniería del Software e Inteligencia Artificial

**Duración:** Cuatrimestral (1º Cuatrimestre)      **Curso:** 2º      **Nivel:** Posgrado

**Tipo:** Optativa

**Profesores que imparten la asignatura:**

Iván Martínez Ortiz (coordinador)  
Pablo Moreno Ger  
Antonio Navarro Martín  
José Luis Sierra Rodríguez  
Baltasar Fernández Manjón

**Créditos:** 6

**Horas/semana:** 2 horas en aula, 0 en laboratorio (1er cuatrimestre)

**Objetivos:** Aprender a diseñar y desarrollar aplicaciones que hacen un uso intensivo de la información utilizando tecnologías basadas en lenguajes de marcado y procesadores de lenguaje. Aplicación al dominio E-learning.

Contenidos:

Introducción al e-Learning.  
Técnicas de Gestión de la Información en e-Learning  
Ingeniería web. Componentes de la web 2.0.  
Estandarización y modelado educativo  
Adaptación al usuario: adquisición, mantenimiento y utilización de modelos de usuario.  
Interactividad y juegos educativos en e-Learning

**Idioma en que se imparte:** Español

**Método docente:** Enseñanza presencial teórica. Realización individual de prácticas no tutorizadas. Tutorías colectivas

**Exámenes:** Examen final en febrero y septiembre.

**Método de evaluación:** Asistencia a clase y participación en la discusión de cada uno de los temas. Realización de trabajos de análisis bibliográfico y/o desarrollos prácticos. Presentación de dichos trabajos en forma oral y escrita, evaluando especialmente su capacidad para situar en contexto y profundizar en el tema investigado. Escritura y presentación pública en clase de un trabajo final que deberá tener la forma de una prepublicación científica.

### **Bibliografía:**

Goldfard, C.; *The SGML Handbook*; Clarendon Press, Oxford. 1990;

Governor J., Hinchcliffe D., Nickull D.; *Web 2.0 Architectures*; O'Reilly, 2009;

Rossi, G., Pastor, O., Schwabe, D. and Olsina, L.; *Web Engineering: Modelling and Implementing Web Applications.*; Springer, 2007;

Aldrich, C.; San Francisco, CA: Pfeiffer, 2005;

**Página web:** Asignatura virtualizada en el Aula Virtual de la U. Complutense

## **122038 - Extensiones de Programación Lógica (EPL)**

---

**Departamento:** Sistemas Informáticos y Computación

**Duración:** Cuatrimestral (1º Cuatrimestre)      **Curso:** 2º      **Nivel:** Posgrado

**Tipo:** Optativa

**Profesores que imparten la asignatura:**

Jaime Sánchez Hernandez (coordinador)

Rafael Caballero Roldán

**Créditos:** 6

**Horas/semana:** 2 horas en aula, 0 en laboratorio (1er cuatrimestre)

**Objetivos:** Introducir diversas extensiones y aplicaciones de la programación lógica, en particular el paradigma lógico-funcional y las bases de datos deductivas.

**Conocimientos y destrezas que se requieren:** Programación lógica: nivel medio; Programación funcional: nivel elemental; Bases de datos: nivel elemental;

**Contenidos:** Programación lógico-funcional con funciones indeterministas, evaluación perezosa y restricciones. La lógica de reescritura CRWL. Modelos operacionales para programación lógico-funcional. Bases de datos deductivas. *Datalog Answer Set Programming*.

**Conocimientos y destrezas que se adquieren:** Programación lógico-funcional con restricciones: nivel medio; Modelos semánticos del paradigma lógico-funcional: nivel elemental; Modelos semánticos de las bases de datos deductivas: nivel medio; Programación en Answer Set Programming: nivel elemental;

**Idioma en que se imparte:** Español

**Método docente:** Enseñanza presencial teórica y de ejercicios, tutorías colectivas con discusión de ejercicios, lecturas sugeridas y presentación de trabajos.

**Exámenes:** Examen final en febrero y septiembre.

**Método de evaluación:** Evaluación continua que tendrá en cuenta asistencia y participación en clase, presentación de algún tema y realización de un examen.

## **Bibliografía:**

J.C. González Moreno, T. Hortalá González, F.J. López Fraguas, M. Rodríguez Artalejo; *An Approach to Declarative Programming based on a Rewriting Logic*; Journal of Logic Programming, 40(1) 47-87, 1999;

Hanus, M.; *Multi-paradigm Declarative Languages*; International Conference on Logic Programming (ICLP 2007), 2007;

J. Minker (Editor); *Foundations of deductive databases and logic programming*; Morgan Kaufmann Publishers Inc., 1987;

Chitta Baral; *Knowledge Representation, Reasoning and Declarative Problem Solving*; Cambridge University Press, 2003;

**Página web:** Páginas de los profesores de la asignatura

## **122039 - Ingeniería Lingüística (IL)**

---

**Departamento:** Ingeniería del Software e Inteligencia Artificial

**Duración:** Cuatrimestral (1º Cuatrimestre)      **Curso:** 2º      **Nivel:** Posgrado

**Tipo:** Optativa

**Profesores que imparten la asignatura:**

Pablo Gervás Gómez-Navarro (coordinador)  
Alberto Díaz Esteban  
Virginia Francisco Gilmartín

**Créditos:** 6

**Horas/semana:** 2 horas en aula, 0 en laboratorio (1er cuatrimestre)

**Objetivos:** Dar a conocer: - Las técnicas existentes para desarrollar interfaces en lenguaje natural para aplicaciones informáticas - Los paradigmas fundamentales de representación para información sintáctica y semántica - Las técnicas computacionales más representativas empleadas en los procesos de análisis y generación. - Las principales aplicaciones de estas herramientas.

**Conocimientos y destrezas que se requieren:** Teoría de Autómatas y Lenguajes formales: nivel medio; Programación declarativa: nivel medio; Programación en algún lenguaje de alto nivel (Java, C++, Prolog...): nivel medio;

**Contenidos:** Introducción al procesamiento de lenguaje natural (PLN); Análisis e interpretación del lenguaje: niveles léxico, sintáctico, semántico, pragmático; Generación de lenguaje; Tratamiento estadístico del lenguaje; Recursos lingüísticos; Procesamiento de voz; Sistemas de diálogo; Aplicaciones;

**Conocimientos y destrezas que se adquieren:** Fundamentos del análisis sintáctico: nivel medio; Fundamentos del análisis semántico: nivel medio; Fundamentos de la generación de LN: nivel medio; Saber utilizar recursos lingüísticos: nivel elemental; Saber utilizar herramientas de PLN: nivel elemental; Fundamentos del tratamiento estadístico del lenguaje: nivel medio; Utilización de tecnologías de procesamiento de voz: nivel elemental;

**Idioma en que se imparte:** Español

**Método docente:** Enseñanza presencial teórica, enseñanza presencial de prácticas de laboratorio, y realización individual de trabajos no tutorizados.

**Exámenes:** Examen final en febrero y septiembre.

**Método de evaluación:** Realización de prácticas, y realización y presentación individual de trabajo.

**Bibliografía:**

Allen, J. ; *Natural Language Understanding*. ; Ed. Benjamin Cummings Publishing Company. 1995.;

Jurafsky, D. & James H. Martin. ; *Speech and Language Processing: An Introduction to Natural Language Processing*, ; Prentice Hall, 2000.;

Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze; *Introduction to Information Retrieval*;; Cambridge University Press. 2008.;

Dale, R., Reiter, E.; *Building Applied Natural Language Generations Systems*; Cambridge University Press, 2000;

---

## **122040 - Razonamiento Aproximado y con Incertidumbre (RAI)**

**Departamento:** Ingeniería del Software e Inteligencia Artificial

**Duración:** Anual                      **Curso:** 2º            **Nivel:** Posgrado

**Tipo:** Obligatoria (según perfil de ingreso)

**Profesores que imparten la asignatura:**

Luis Garmendia Salvador (coordinador)

**Créditos:** 4,5

**Horas/semana:** 2 horas en aula, 0 en laboratorio (1er cuatrimestre)

**Objetivos:** Capacidad de trabajar en equipo; Capacidad de comunicación oral y escrita en castellano o inglés;

**Conocimientos y destrezas que se requieren:** Lógica: nivel elemental; Inteligencia Artificial e Ingeniería del Conocimiento: nivel elemental;

Breve Descriptor

Introducción a la Lógica Fuzzy (borrosa o difusa) generalizando la lógica clásica para hacer razonamiento con incertidumbre, falta de nitidez o de información. Fundamentos teóricos y aplicaciones.

**Contenidos:** 1.- Lógica borrosa: nuevas tendencias de la matemática 2.- Introducción a la Inteligencia Artificial 3.- Conjuntos difusos. 4.- Lógicas borrosas. 5.- Familias de t-normas, t-conormas y negaciones. 6.- Relaciones borrosas: similaridades y T-indistinguibilidades. 7.- Razonamiento aproximado. Regla composicional de inferencia. Operadores de Implicación. 8.- Aplicaciones. Control fuzzy

**Conocimientos y destrezas que se adquieren:** Capacidad de trabajar en equipo; Capacidad de comunicación oral y escrita en castellano o inglés;

**Idioma en que se imparte:** Español

**Método docente:**

Clases teóricas

Enseñanza presencial teórica .

Tutoría colectiva

Trabajos de campo

Realización voluntaria de trabajo sobre teoría de conjuntos borrosos o sobre aplicaciones de control fuzzy con XFuzzy en java (40-60% de la asignatura)

**Presentaciones**

Presentación oral voluntaria del trabajo realizado o presentación de divulgación científica de aspectos teóricos o de aplicaciones de la lógica borrosa (20% de la asignatura)

**Exámenes:** Examen final en febrero y septiembre.

**Método de evaluación:**

Asistencia a clase y participación: 20%

Realización de ejercicios propuestos: 0-20%

Trabajo: 40-60%

Comunicación oral o presentación oral del trabajo: 20%

**Bibliografía:**

H. T. Nguyen y E. A. Walker; A first course in Fuzzy Logic; CRC Press. (1996).;

Enric Trillas y otros; Introducción a la lógica borrosa; Ed. Ariel. 1995;

G. Klir y T. A. Folger; Fuzzy sets, uncertainty and information; Prentice-Hall, Singapore (1992).;

H.J. Zimmermann; Fuzzy Set Theory and its Applications ; Boston : Kluwer Academic Publishers, cop.1996;

**Página web:** [www.fdi.ucm.es/profesor/lgarmend](http://www.fdi.ucm.es/profesor/lgarmend)

---

**122041 - Proyecto fin de Máster en Ingeniería de Computadores (PFMIC)**

---

**Departamento:** Arquitectura de Computadores y Automática

Duración: Anual                      Curso: 2º                      Nivel: Posgrado

**Tipo:** Obligatoria (según especialidad)

**Profesores que imparten la asignatura:**

Francisco Tirado Fernández (coordinador);

**Créditos:** 30

**Horas/semana:** 0 horas en aula, 0 en laboratorio

**Objetivos:** Desarrollar capacidades de investigación básicas en el área de la Ingeniería de Computadores

**Contenidos:** Desarrollo de un proyecto de investigación bajo la dirección de un doctor perteneciente al programa de doctorado y especializado en Ingeniería de Computadores.

**Conocimientos y destrezas que se adquieren:** Autonomía investigadora: nivel medio;

**Idioma en que se imparte:** Español

**Método docente:** Seguimiento personalizado del trabajo de investigación.

**Exámenes:** Examen final en junio y septiembre.

**Método de evaluación:** Presentación del trabajo ante un tribunal de doctores.

### **122042 - Proyecto fin de Máster en Ingeniería Informática para la Industria (PFMIII)**

---

**Departamento:** Arquitectura de Computadores y Automática

**Duración:** Anual                      Curso: 2º                      Nivel: Posgrado

**Tipo:** Obligatoria (según especialidad)

**Profesores que imparten la asignatura:**

Francisco Tirado Fernández (coordinador);

**Créditos:** 30

**Horas/semana:** 0 horas en aula, 0 en laboratorio

**Objetivos:** Desarrollar capacidades de investigación básicas en el área de la Ingeniería Informática para la Industria

**Contenidos:** Desarrollo de un proyecto de investigación bajo la dirección de un doctor perteneciente al programa de doctorado y especializado en Ingeniería Informática para la Industria.

**Conocimientos y destrezas que se adquieren:** Autonomía investigadora: nivel medio;

**Idioma en que se imparte:** Español

**Método docente:** Seguimiento personalizado del trabajo de investigación.

**Exámenes:** Examen final en junio y septiembre.

**Método de evaluación:** Presentación del trabajo ante un tribunal de doctores.

### **122043 - Proyecto fin de Máster en Programación y Tecnología Software (PFMPTS)**

---

**Departamento:** Sistemas Informáticos y Computación

**Duración:** Anual                      Curso: 2º                      Nivel: Posgrado

**Tipo:** Obligatoria (según especialidad)

**Profesores que imparten la asignatura:**

Francisco Javier López Fraguas (coordinador);

**Créditos:** 30

**Horas/semana:** 0 horas en aula, 0 en laboratorio



**Objetivos:** Desarrollar capacidades de investigación básicas en el área de la Programación y Tecnologías Software

**Contenidos:** Desarrollo de un proyecto de investigación bajo la dirección de un doctor perteneciente al programa de doctorado y especializado en programación y tecnologías software.

**Conocimientos y destrezas que se adquieren:** Autonomía investigadora: nivel medio;

**Idioma en que se imparte:** Español

**Método docente:** Seguimiento personalizado del trabajo de investigación.

**Exámenes:** Examen final en junio y septiembre.

**Método de evaluación:** Presentación del trabajo ante un tribunal de doctores

---

### **122044 - Proyecto fin de Máster en Sistemas Inteligentes (PFMSI)**

**Departamento:** Ingeniería del Software e Inteligencia Artificial

**Duración:** Anual                      **Curso:** 2º                      **Nivel:** Posgrado

**Tipo:** Obligatoria (según especialidad)

**Profesores que imparten la asignatura:**

Juan Pavón Mestras (coordinador);

**Créditos:** 30

**Horas/semana:** 0 horas en aula, 0 en laboratorio

**Objetivos:** Desarrollar capacidades de investigación básicas en el área de los Sistemas Inteligentes

**Contenidos:** Desarrollo de un proyecto de investigación bajo la dirección de un doctor perteneciente al programa de doctorado y especializado en Sistemas Inteligentes.

**Conocimientos y destrezas que se adquieren:** Autonomía investigadora: nivel medio;

**Idioma en que se imparte:** Español

**Método docente:** Seguimiento personalizado del trabajo de investigación.

**Exámenes:** Examen final en junio y septiembre.

**Método de evaluación:** Presentación del trabajo ante un tribunal de doctores.

---

### **COMPLEMENTOS FORMATIVOS**

---

#### **122001 - Estructura de Datos y de la Información (EDI)**

**Departamento:** Sistemas Informáticos y Computación

**Duración:** Anual                      **Curso:** 1º                      **Nivel:** Posgrado

**Tipo:** Obligatoria (según perfil de ingreso)

**Profesores que imparten la asignatura:**

Miguel Gómez-Zamalloa Gil  
M<sup>a</sup> de las Mercedes García Merayo

**Créditos:** 12

**Horas/semana:** 5 horas en aula, 0 en laboratorio

**Objetivos:** Obtener la capacidad de especificar tipos abstractos de datos (TAD); implementarlos con la estructura de datos adecuada; analizar la eficiencia; utilizarlos en programas.

**Conocimientos y destrezas que se requieren:** Programación estructurada; Tipos de datos elementales; Recursión; Lógica de primer orden; Inducción;

**Contenidos:**

1. Análisis de la eficiencia de algoritmos
2. Diseño y análisis de algoritmos iterativos
3. Diseño y análisis de algoritmos recursivos
4. Especificación e implementación de tipos abstractos de datos
5. Tipos de datos lineales: pilas, colas, listas
6. Árboles
7. Tablas
8. Grafos

**Conocimientos y destrezas que se adquieren:** Diseño de algoritmos iterativos y recursivos; Análisis de la complejidad de algoritmos; Tipos abstractos de datos; Estructuras de datos lineales; Estructuras de datos arborescentes; Estructuras de datos funcionales; Estructuras de datos relacionales;

**Idioma en que se imparte:** Español

**Exámenes:** Primer cuatrimestre. Segundo cuatrimestre (conjunto con el final de junio). Examen final en junio y septiembre.

**Método de evaluación:** El examen parcial de febrero libera la materia correspondiente hasta septiembre. Los exámenes finales de junio y septiembre constan de dos partes, una correspondiente al primer cuatrimestre, que no tendrán que realizar quienes hayan aprobado en febrero, y otra correspondiente al segundo cuatrimestre. Para aprobar es necesario tener al menos un 4 en cada una de las partes y que la nota media sea al menos 5.

**Bibliografía:**

Peña, R. ; Diseño de programas. Formalismo y abstracción; Prentice Hall, 1998.;

Martí Oliet, N., Ortega Mallén, Y., Verdejo López, J. A.; *Estructuras de datos y métodos algorítmicos: Ejercicios resueltos*; Colección Prentice Practica, Pearson/Prentice Hall, 2003;

Larry Nyhoff ; C++: An Introduction to Data Structures ; Prentice Hall, 1998.;

**Página web:** [http://www.fdi.ucm.es/href.asp?http=/vista/asp/bbdd/Docu\\_Docente2.asp](http://www.fdi.ucm.es/href.asp?http=/vista/asp/bbdd/Docu_Docente2.asp)

**122002 - Metodología y Tecnología de la Programación (MTP)**

---

**Departamento:** Sistemas Informáticos y Computación

**Duración:** Anual                      **Curso:** 1º      **Nivel:** Posgrado

**Tipo:** Obligatoria (según perfil de ingreso)

**Profesores que imparten la asignatura:**

Clara Mª Segura Díaz

**Créditos:** 12

**Horas/semana:** 5 horas en aula, 0 en laboratorio (1er cuatrimestre); 3 horas en aula (2º cuatrimestre)

**Objetivos:** Conocer los distintos métodos de análisis y diseño de algoritmos y ser capaz de aplicarlos en el desarrollo de soluciones para problemas variados. Valorar cuál de los distintos métodos aplicables al desarrollo de una solución es el mejor para cada caso concreto desde el punto de vista de la complejidad.

**Conocimientos y destrezas que se requieren:** Métodos de diseño de algoritmos; Estructuras de datos; Análisis de la complejidad de algoritmos;

**Contenidos:** Aspectos avanzados de la complejidad de algoritmos. Transformación de algoritmos recursivos a iterativos. Divide y vencerás. Programación dinámica. Algoritmos voraces. Análisis amortizado. Exploración del espacio de soluciones: vuelta atrás, árboles de juego, ramificación y poda. Algoritmos probabilísticos. Precondicionamiento. Complejidad de problemas.

**Conocimientos y destrezas que se adquieren:** Análisis avanzado de la complejidad de algoritmos; Transformación de algoritmos recursivos a iterativos; Divide y vencerás; Programación dinámica; Análisis amortizado; Algoritmos voraces; Algoritmos de exploración del espacio de soluciones: vuelta atrás, árboles de ju; Algoritmos probabilísticos; Precondicionamiento; Complejidad de problemas;

**Idioma en que se imparte:** Español

**Método docente:** Enseñanza presencial teórica. Enseñanza presencial de (problemas / ejercicios).

**Exámenes:** Primer cuatrimestre. Examen final en junio y septiembre.

**Método de evaluación:** Febrero: Examen parcial necesario y liberatorio (60% de la nota final). Junio: Examen final necesario, no obligatorio y liberatorio (40% de la nota final). Septiembre: Examen final (60% + 40% de la nota final). Constará de dos partes. El alumno que haya aprobado únicamente el examen de febrero o junio podrá presentarse solamente a la otra parte.

**Bibliografía:**

E. Horowitz, S. Shani, S. Rajasekaran ; *Computer algorithms*; 3a edición, Computer Science Press, 1998;

R. Neapolitan, K. Naimipour; *Foundations of algorithms*; 2a o 3a edición, Jones and Bartlett Publishers,

1998 o 2003;

Narciso Martí, Yolanda Ortega, Alberto Verdejo; *Estructuras de datos y métodos algorítmicos: ejercicios resueltos*; 1a edición, Prentice Hall, 2003;

Levitin, A. ; Introduction to the design and analysis of algorithms; Addison-Wesley, 2003;

**Página web:** Campus Virtual

## **122003 - Teoría de Autómatas y Lenguajes Formales (TALF)**

---

**Departamento:** Sistemas Informáticos y Computación

**Duración:** Anual                    **Curso:** 1º            **Nivel:** Posgrado

**Tipo:** Obligatoria (según perfil de ingreso)

**Profesores que imparten la asignatura:**

Fernando Rosa Velardo

**Créditos:** 9

**Horas/semana:** 3 horas en aula, 0 en laboratorio

**Objetivos:** Primero, se trata de comprender el concepto de sistema informático abstracto, independiente de tecnologías, lo que conlleva el estudio de los autómatas y su jerarquía. Además, deben estudiarse los lenguajes formales como parte básica de la Informática.

**Conocimientos y destrezas que se requieren:** Inducción y recursión; Programación estructurada;

**Contenidos:** 1. Introducción a los Lenguajes Formales. 2. Lenguajes Regulares . 3. Lenguajes Independientes del Contexto. 4. Maquinas de Turing. 5. Problemas indecibles, intratables y otras clases de problemas.

**Conocimientos y destrezas que se adquieren:** Clasificación y procesamiento de lenguajes, autómatas y gramáticas: nivel medio; Computabilidad: nivel elemental;

**Idioma en que se imparte:** Español

**Método docente:** Enseñanza presencial teórica. Enseñanza presencial de problemas. Realización individual de ejercicios y problemas no tutorizados.

**Exámenes:** Primer cuatrimestre. Segundo cuatrimestre (conjunto con el final de junio). Examen final en junio y septiembre.

**Método de evaluación:**

Examen parcial de febrero que libera hasta junio la materia correspondiente. Examen parcial en junio. Para poder presentarse en junio al segundo parcial será necesario tener una nota de al menos 5 en el parcial de febrero. Para aprobar por parciales es necesario tener al menos un 5 en cada una de las partes. Exámenes finales en junio y septiembre.

**Bibliografía:**

J. E. Hopcroft, R. Motwani, J. Ullman; *Introducción a la Teoría de Autómatas, Lenguajes y Computación*; 3ª ed., Addison-Wesley, 2008;

John C. Martin; *Introduction to Languages and the Theory of Computation*; Mac. Graw-Hill, 1991;

D. Kelley; *Teoría de Autómatas y Lenguajes Formales*; Prentice Hall, 1995;

## **122004 - Procesadores de Lenguaje (PLg)**

---

**Departamento:** Sistemas Informáticos y Computación

**Duración:** Anual                    **Curso:** 1º      **Nivel:** Posgrado

**Tipo:** Obligatoria (según perfil de ingreso)

**Profesores que imparten la asignatura:**

Antonio Pareja Lora

**Créditos:** 9

**Horas/semana:** 3 horas en aula, 0 en laboratorio

**Objetivos:** El objetivo de esta asignatura es introducir al alumno en los principios y técnicas utilizadas en la definición de lenguajes de programación y en la construcción de sus procesadores (traductores, compiladores o intérpretes).

**Conocimientos y destrezas que se requieren:** Teoría de Autómatas y Lenguajes Formales: nivel medio; Estructuras de datos: nivel medio; Lenguajes: Prolog, Pascal, C , C++, Java: nivel medio;

**Contenidos:** 1.- Analizadores léxicos. 2.- Analizadores sintácticos descendentes y ascendentes y traductores asociados. 3.- Comprobación de la semántica estática y de las restricciones contextuales. 4.- Generación de código. 5.- Optimización. 6.- Herramientas para el prototipado y la construcción de procesadores de lenguaje. 7.- Aplicaciones de los procesadores de lenguaje.

**Conocimientos y destrezas que se adquieren:** Definición y especificación de lenguajes de programación y de sus procesadores: nivel medio; Implementación de procesadores mediante lenguajes de propósito general: nivel medio; Herramientas de prototipado y construcción de procesadores: nivel medio; Fundamentos de las herramientas de construcción de procesadores: nivel medio;

**Idioma en que se imparte:** Español

**Método docente:** Enseñanza presencial teórica. Enseñanza presencial de ejercicios y supuestos prácticos. Realización de prácticas no tutorizadas.

**Exámenes:** Examen final en junio y septiembre.

**Método de evaluación:** Los alumnos serán evaluación en base a una práctica y a un examen final en aula. - Realización de prácticas. Esta prueba es eliminatoria (30% de la nota final). Para superar las prácticas es necesario: (i) realizar las mismas en grupo y entregarlas y superarlas en los plazos establecidos durante el período de clases de la asignatura (de Octubre de 2010 a Junio de 2011), o bien (ii) realizar individualmente otras prácticas diferentes, durante el período no docente, y entregarlas y superarlas en Septiembre, en las fechas establecidas. - Realización de examen final en aula. Esta prueba es eliminatoria (70% de la nota final).

**Bibliografía:**

Aho, A.V. , Sethi, R., Ullman, J.D.; *Compilers. Principles, Techniques and Tools.*; Addison-Wesley. 1988 (existe una edición en español actualizada - 2008);

Wilhen, R. , Maurer, D.; *Compiler Design*; Addison-Wesley. 1995;

Fischer, C.N. , LeBlanc, R,J. Jr; *Crafting a Compiler with C.* ; Addison-Wesley. 1991;

Sethi, R; *Programming Languages, Concepts and Constructs*; 2nd. Edition. Addison-Wesley. 1996;

**Página web:** www.ucm.es -> Campus Virtual

## **122005 - Estructura y Tecnología de Computadores (ETC)**

---

**Departamento:** Arquitectura de Computadores y Automática

**Duración:** Anual                      Curso: 1º            Nivel: Posgrado

**Tipo:** Obligatoria (según perfil de ingreso)

**Profesores que imparten la asignatura:**

Pedro Luis García Repetto (coordinador)

José Luis Vázquez Poletti

**Créditos:** 9

**Horas/semana:** 5 horas en aula, 0 en laboratorio

**Objetivos:** Estudio de las tecnologías empleadas en la fabricación de circuitos digitales. Estudio de la estructura y organización de los computadores.

**Conocimientos y destrezas que se requieren:** Resolución de circuitos: nivel medio; Fundamentos de semiconductores: nivel medio; Diseño digital: nivel elemental; Programación en ensamblador: nivel elemental; Estructura de computadores: nivel elemental;

**Contenidos:** 1. Dispositivos básicos de conmutación. 2. Familias lógicas. 3. Circuitos regenerativos. 4. Sistemas combinacionales y secuenciales. 5. Introducción a la estructura de computadores. 6. Arquitectura del procesador. 7. Diseño del procesador. 8. Buses de sistema. 9. El subsistema de entrada/salida. 10. El subsistema de memoria.

**Conocimientos y destrezas que se adquieren:** Familias lógicas: nivel medio; Circuitos de electrónica digital: nivel medio; Diseño digital. Optimización de sistemas digitales: nivel medio; Programación en ensamblador: nivel medio; Estructura de computadores: nivel avanzado; Arquitectura de computadores: nivel elemental;

**Idioma en que se imparte:** Español

**Método docente:** Enseñanza presencial teórica. Enseñanza presencial de problemas.

**Exámenes:** Primer cuatrimestre. Segundo cuatrimestre (conjunto con el final de junio). Examen final en junio y septiembre.

**Método de evaluación:** Convocatoria de junio: examen final (presupone que es obligatorio, escrito, en aula) formado por cuestiones, desarrollo teórico y problemas. Examen cuatrimestral en febrero (presupone que es opcional, escrito y liberatorio) que incluye cuestiones, desarrollo teórico y problemas. Convocatoria de septiembre: examen final escrito que incluye cuestiones, desarrollo teórico y problemas.

**Bibliografía:**

N. R. Malik; Circuitos electrónicos. Análisis, simulación y diseño; Prentice Hall- 1996;

M. Ercegovac, T. Lang and J. Moreno; *Introduction to Digital Systems*; John Wiley & Sons, 1999;

D. A. Patterson, J. L. Hennessy; *Estructura y diseño de computadores*; Editorial Reverté. 2000;

W. Stallings; *Organización y arquitectura de computadores*; Prentice Hall, 4ª Edición, 2000;

**Página web:** Campus virtual de la UCM

## 122006 - Sistemas Operativos (SO)

---

**Departamento:** Arquitectura de Computadores y Automática

**Duración:** Cuatrimestral      **Curso:** 1º      **Nivel:** Posgrado

**Tipo:** Obligatoria (según perfil de ingreso)

**Profesores que imparten la asignatura:**

M<sup>a</sup> Teresa Higuera Toledano

**Créditos:** 6

**Horas/semana:** 4 horas en aula, 0 en laboratorio (1er cuatrimestre)

**Objetivos:** Presentar los conceptos, estructura y mecanismos que fundamentan el comportamiento de los sistemas operativos, con referencias prácticas concretas a UNIX y Windows-NT

**Conocimientos y destrezas que se requieren:** Lenguaje máquina: nivel elemental; Estructura básica de un computador: nivel medio; Diseño de programas; Implementación de programas; Depuración de programas; Especificación de algoritmos iterativos y recursivos; Estructuras de datos lineales: nivel medio; Estructuras de datos arborescentes: nivel medio;

**Contenidos:** Repaso de estructura de Computadores Introducción a los sistemas operativos: visión general de su función y estructura. Llamadas al sistema. Gestión de procesos e hilos: creación, seguimiento, planificación y finalización Sincronización y Comunicación de procesos e hilos: semáforos, monitores y mensajes. Interbloqueos: prevención, evitación y detección. Gestión de memoria virtual. Gestión de entrada/salida: rutinas de dispositivos. Terminales, discos y red Gestión de sistemas de ficheros: volúmenes, directorios y ficheros.

**Conocimientos y destrezas que se adquieren:** Funcionalidad de un sistema operativo: nivel avanzado; Estructura de un sistema operativo: nivel avanzado; Servicios de un sistema operativo: nivel avanzado; Concurrencia y gestión de procesos: nivel medio; Gestión de memoria: nivel avanzado; Gestión de dispositivos: nivel medio; Gestión de ficheros: nivel avanzado;

**Idioma en que se imparte:** Español

**Método docente:** Enseñanza presencial teórica. Enseñanza presencial de problemas

**Exámenes:** Examen final en febrero y septiembre.

**Método de evaluación:** Convocatoria de febrero: Examen final obligatorio y escrito, formado por teoría y problemas. Se exigirá una puntuación mínima en cada parte del examen (teoría y problemas) Convocatoria de septiembre: Examen final escrito que incluye teoría y problemas. Se exigirá una puntuación mínima en cada parte del examen (teoría y problemas)

**Bibliografía:**

J.Carretero y otros; *Sistemas Operativos: una visión aplicada*; McGraw-Hill, 2007;

F. Pérez Costoya y otros ; *Problemas de Sistemas Operativos: de la Base al Diseño*; McGraw-Hill, 2002;

W.Stallings; *Sistemas Operativos* ; 4ª edición. Prentice-Hall, 2002 ;

A.Tanenbaum ; *Sistemas Operativos Modernos*; 2ª edición, Prentice-Hall, 2003;

**Página web:** Campus Virtual

## 122007 - Redes (RED)

---

**Departamento:** Arquitectura de Computadores y Automática

**Duración:** Anual      **Curso:** 1º      **Nivel:** Posgrado

**Tipo:** Obligatoria (según perfil de ingreso)

**Profesores que imparten la asignatura:**

Julio Septién del Castillo (coordinador)

**Créditos:** 9

**Horas/semana:** 3 horas en aula, 0 en laboratorio (1er cuatrimestre); 1 hora en aula, 2 en laboratorio (2º cuatrimestre)

**Objetivos:** Conseguir la comprensión de los problemas de cada uno de los distintos niveles de una arquitectura de red, las técnicas usadas para resolverlos y su funcionamiento integrado

**Conocimientos y destrezas que se requieren:**

**Contenidos:** Introducción a las redes. Arquitecturas de red. La capa física: conceptos de transmisión de datos y medios de transmisión. La capa física en redes públicas: Red telefónica, RDSI, GSM. Tecnologías de acceso residencial: xDSL, Cable módem, PLC. La capa de enlace de datos: protocolos de gestión. HDLC, PPP, LLC. La capa de control de acceso al medio: redes locales y metropolitanas. Ethernet, Token Ring, FDDI, redes de alta velocidad, redes inalámbricas. La capa de red: redes de área extensa. Redes de paquetes, Frame Relay, ATM. Interconexión de redes: el protocolo IP. ICMP, ARP y DNS. Routing en Internet: RIP, OSPF. IP sobre redes Ethernet y ATM. IP versión 6. La capa de transporte: TCP, UDP. Modelo cliente-servidor e interfaz de sockets. La capa de aplicación: aplicaciones representativas sobre la arquitectura TCP/IP.

**Conocimientos y destrezas que se adquieren:** Arquitectura de redes: nivel medio; Medios físicos de transmisión: nivel medio; Técnicas de transmisión de datos: nivel medio; Redes locales y metropolitanas: nivel medio; Redes de área extensa : nivel medio; Protocolos TCP/IP: nivel medio; Aplicaciones sobre TCP/IP: nivel medio;

**Idioma en que se imparte:** Español

**Método docente:** Enseñanza presencial teórica. Enseñanza presencial de problemas.

**Exámenes:** Primer cuatrimestre. Segundo cuatrimestre (conjunto con el final de junio). Examen final en junio y septiembre.

**Método de evaluación:** Convocatorias de Junio y Septiembre: examen final formado por desarrollos teóricos, cuestiones, tests y/o problemas. Examen cuatrimestral en Febrero, por desarrollos teóricos, cuestiones, tests y/o problemas, laboratorio para ambas convocatorias.

**Bibliografía:**

W. Stallings; *Comunicaciones y Redes de Computadores*; Prentice-Hall, 7ª Edición, 2004;

A.Tannenbaum; *Redes de Computadoras*; Prentice-Hall, 4ª Edición, 2004;

D. E. Comer; *Redes globales de información con Internet y TCP/IP Vol. 1.*; Prentice-Hall, 3ª edición, 1996 (4ª edición en inglés);

F. Halsall; *Redes de Computadoras e Internet*; Pearson/Prentice-Hall, 5ª ed. , 2006;



## **122008 - Arquitectura e Ingeniería de Computadores (AIC)**

---

**Departamento:** Arquitectura de Computadores y Automática

**Duración:** Anual      **Curso:** 1º      **Nivel:** Posgrado

**Tipo:** Obligatoria (según perfil de ingreso)

**Profesores que imparten la asignatura:**

Román Hermida Correa (coordinador)  
Ignacio Martín Llorente

**Créditos:** 9

**Horas/semana:** 4 horas en aula, 0 en laboratorio (1er cuatrimestre); 2 horas en aula, 0 en laboratorio (2º cuatrimestre)

**Objetivos:** Estudiar conceptos avanzados de arquitectura de sistemas digitales con especial énfasis en paralelismo a nivel de instrucciones y arquitecturas paralelas.

**Conocimientos y destrezas que se requieren:** Diseño de sistemas digitales síncronos: nivel avanzado; Estructura de Computadores: nivel avanzado; Programación en lenguaje de alto nivel: nivel medio; Sistemas Operativos: nivel medio; Programación en lenguaje ensamblador: nivel medio;

**Contenidos:** 1. Introducción: tendencias tecnológicas, coste y rendimiento. 2. Aspectos básicos de la segmentación. 3. Paralelismo a nivel de instrucción: planificación dinámica de instrucciones, técnicas de predicción de saltos, ejecución especulativa. 4. Lanzamiento múltiple de instrucciones, límite de ILP y multithreading. 5. Caches: reducción de fallos, ocultación de latencia. Memoria principal y memoria virtual: ancho de banda y latencia. 6. Procesamiento paralelo: Introducción, memoria compartida, memoria distribuida. 7. Redes de interconexión de multiprocesadores. 8. Sincronización. Coherencia. Consistencia. 9. Aspectos de la programación de multiprocesadores, programas paralelos, proceso de paralelización, explotación del paralelismo en bucles.

**Conocimientos y destrezas que se adquieren:** Relación tecnología-coste-rendimiento: nivel avanzado; Segmentación: nivel avanzado; Paralelismo a nivel de instrucción: nivel avanzado; Especulación: nivel avanzado; Optimización de la jerarquía de memoria: nivel avanzado; Multiprocesadores: nivel medio; Redes de interconexión de procesadores: nivel avanzado; Coherencia y consistencia: nivel avanzado; Programación paralela: nivel medio;

**Idioma en que se imparte:** Español

**Método docente:** Enseñanza presencial teórica. Enseñanza presencial de problemas.

**Exámenes:** Primer cuatrimestre. Segundo cuatrimestre (conjunto con el final de junio). Examen final en junio y septiembre.

**Método de evaluación:** Examen cuatrimestral en febrero: opcional, escrito y liberatorio, que incluye teoría y problemas (Peso en la nota final: 65%). Plazo de validez: hasta la convocatoria de septiembre. Nota: A mitad del primer cuatrimestre habrá una prueba de clase opcional tipo test, cuyo peso en la nota del primer cuatrimestre será del 20% para los alumnos que deseen realizarla. Convocatoria de junio: Examen final obligatorio y escrito, formado por teoría y problemas. Convocatoria de septiembre: Examen final escrito que incluye teoría y problemas. En los exámenes finales, los contenidos del primer cuatrimestre tendrán un peso del 65%, y los del segundo del 35%, en la calificación final. Mismo examen (prácticas y otros elementos de

evaluación, en su caso) en todos los grupos y criterios detallados de puntuación comunes. Mismo examen en todos los grupos corregido de forma horizontal entre todos los profesores de la asignatura.

### **Bibliografía:**

J. Hennessy, D. Patterson; *Computer Architecture: A Quantitative Approach*; (4th edition), Morgan Kaufmann Publishers, 2007;

David E. Culler, Jaswinder P. Singh; *Parallel Computer Architecture: A hardware/software approach*; Morgan Kaufmann Publishers, 1999. ;

D. Sima, T. Fountain, P. Kasuc; *Advanced computer Architecture: A design space approach*; Addison-Wesley, 1997;

**Página web:**<http://www.fdi.ucm.es/profesor/rhermida/aic.htm>

## **122009 - Bases de Datos y Sistemas de Información (BDSI)**

---

**Departamento:** Sistemas Informáticos y Computación

**Duración:** Anual                      **Curso:** 1º                      **Nivel:** Posgrado

**Tipo:** Obligatoria (según perfil de ingreso)

**Profesores que imparten la asignatura:**

Diego Blanco Moreno (coordinador)

**Créditos:** 12

**Horas/semana:** 4 horas en aula, 0 en laboratorio (1er cuatrimestre); 2 horas en aula, 2 en laboratorio (2º cuatrimestre)

**Objetivos:** Adquirir conocimientos teóricos y destrezas para ser capaz de analizar un problema, crear un diseño e implementar una Bases de Datos de tamaño medio.

**Conocimientos y destrezas que se requieren:** Programación orientada a objetos: nivel elemental; Sistemas operativos: nivel elemental; Programación lógica: nivel elemental;

**Contenidos:** 1. Introducción a los Sistemas de Gestión de Bases de Datos (SGBD) y Modelos de bases de datos, 2. Lenguajes de definición y manipulación de bases de datos, 3. Diseño de bases de datos relacionales, 4. Organización física de los datos. Gestión de archivos, 5. Procesamiento de consultas, 6. Transacciones y control de la concurrencia, 7. Temas avanzados

**Conocimientos y destrezas que se adquieren:** Modelo relacional: nivel avanzado; Modelo orientado a objetos: nivel elemental; Diseño de bases de datos relacionales: nivel avanzado; Gestión de transacciones y control de la concurrencia: nivel elemental; Sistema gestor de bases de datos: nivel elemental; Utilización del lenguaje de definición y manipulación de datos: SQL: nivel medio; Utilización de lenguaje Procedimental PL/SQL: nivel medio; Capacidad de análisis de problemas: nivel medio;

**Idioma en que se imparte:** Español

**Método docente:** Enseñanza presencial teórica de problemas y de prácticas de laboratorio. Realización en grupo de un trabajo tutorizado.

**Exámenes:** Primer cuatrimestre. Segundo cuatrimestre (conjunto con el final de junio). Examen final en junio y septiembre.

**Método de evaluación:** Examen escrito de conocimientos teóricos y prácticos cuatrimestral Febrero (liberatorio hasta Junio). Examen Final en Junio y Septiembre. Trabajo obligatorio. Prácticas obligatorias en laboratorio.

**Bibliografía:**

SILBERSCHATZ, A., KORTH, H.F., SUDARSHAN, S.; *Database System Concepts (Fundamentos de bases de datos)*; 5ª edición, McGraw-Hill, 2006;

ELMASRI, R., NAVATHE, S.B. ; *Fundamentals of Database Systems(Fundamentos de sistemas de bases de datos)*; 4ª edición, Addison-Wesley, 2004;

ULLMAN, J.D. ; *Principles of Databases and Knowledge Base Systems*; Computer Science Press, 1998;

PONS, O., ACIO, S., MARÍN, N., MEDINA, J.M., VILA, Mª A. ; *Introducción a los Sistemas de Bases de Datos*; Paraninfo, 2008;

**Página web:** Ver página de docencia de cada profesor

**122010 - Ingeniería del Software 1 (IS1)**

---

**Departamento:** Sistemas Informáticos y Computación

**Duración:** Cuatrimestral (1er cuatrimestre)

**Curso:** 1º

**Nivel:** Posgrado

**Tipo:** Obligatoria (según perfil de ingreso)

**Profesores que imparten la asignatura:**

Miguel Ángel Blanco Rodríguez (coordinador);

**Créditos:** 9

**Horas/semana:** 4 horas en aula, 2 en laboratorio (1er cuatrimestre)

**Departamento:** Sistemas Informáticos y Computación

**Objetivos:** Que el alumno sea capaz de gestionar proyectos software de dimensión industrial

**Conocimientos y destrezas que se requieren:** Capacidad de comunicación oral y escrita en español: nivel avanzado; Capacidad de resolución cualitativa de problemas: nivel medio; Capacidad de trabajar en equipo: nivel medio; Capacidad de aprendizaje autónomo: nivel medio; Capacidad de obtención autónoma de información: nivel medio; Capacidad de comprensión de documentación técnica: nivel avanzado; Capacidad de comunicación oral y escrita en inglés: nivel medio;

**Contenidos:** Introducción a la Ingeniería de Software, Planificación y gestión de proyectos, El proceso de desarrollo de software, Análisis y especificación de Requisitos, Introducción a UML y el modelado de aplicaciones

**Conocimientos y destrezas que se adquieren:** Especificación de requisitos para aplicaciones: nivel elemental; Planificación y gestión de proyectos: nivel elemental; Ingeniería de requisitos: nivel elemental; Documentación de proyectos software: nivel elemental; Modelado de aplicaciones: nivel elemental;

**Idioma en que se imparte:** Español

**Exámenes:** Examen final en febrero y septiembre.

**Método de evaluación:** Convocatoria de Febrero. Realización obligatoria de un proyecto, de temática libre, en equipos formados de acuerdo con las instrucciones del profesor. El proyecto supondrá el 60% de la nota. La nota del proyecto se descompondrá en una nota global del proyecto (50%) y una individual (50%). La nota individual se obtendrá como resultado del seguimiento de la actividad del estudiante en el proyecto. Este seguimiento se realizará en el laboratorio y con el uso de aplicaciones a determinar por el profesor. La nota del proyecto se obtendrá tras una serie de entregas. Las entregas se harán en Noviembre y Enero. El resto de la nota de la asignatura (40%) se obtendrá de un examen sobre la realización del proyecto. Es necesario aprobar examen y proyecto para aprobar la asignatura. Convocatoria de Septiembre. Para los que hayan hecho el proyecto, un examen en los mismos términos que en Junio. Para los que no hayan hecho el proyecto en el curso actual, además del examen, se pedirá un trabajo equivalente en forma de proyecto individual a entregar con antelación al examen. En Septiembre, el examen será un 60% de la nota y el proyecto un 40%.

### **Bibliografía:**

G. Booch, J. Rumbaugh, I. Jacobson; *UML. El lenguaje de modelado unificado. Guía del usuario. 2ª Edición*; Ed. Addison Wesley Iberoamericana, 2006;

Pressman, R.S.; *Ingeniería del Software: Un Enfoque Práctico. (6ª Edición)*; Mc Graw Hill. 2006;

Sommerville, I.; *Software Engineering (8ª Edition)*; Addison - Wesley. 2007;

## **122011 - Ingeniería del Software 2 (IS2)**

---

**Departamento:** Sistemas Informáticos y Computación

**Duración:** Cuatrimestral (2º cuatrimestre)      **Curso:** 1º      **Nivel:** Posgrado

**Tipo:** Obligatoria (según perfil de ingreso)

**Profesores que imparten la asignatura:**

Miguel Ángel Blanco Rodríguez (coordinador);

**Créditos:** 9

**Horas/semana:** 2 horas en aula, 4 en laboratorio (2º cuatrimestre)

Competencias Específicas

Especificación, diseño, modelado y validación de aplicaciones: nivel medio; Planificación y gestión de proyectos software: nivel medio; Documentación de proyectos software: nivel medio; Patrones de diseño: nivel medio; Desarrollo en equipo: nivel medio; Integración de código a gran escala: nivel medio; Herramientas CASE: nivel medio; UML: nivel medio; Diseño de pruebas: nivel medio;

### **ACTIVIDADES DOCENTES**

Clases teóricas

Clases magistrales donde se explicarán la teoría de ingeniería del software

Laboratorios

Desarrollo de un proyecto

### **Breve descriptor:**

La ingeniería del software es una material pluridisciplinar donde se intenta hacer de la producción del software algo controlado y predecible. En esta asignatura se estudian la parte de

diseño e implementación de un desarrollo. Esta implementación y diseño se deben ejecutar en el marco teórico proporcionado por la asignatura IS1, esto es, dentro de un proceso de desarrollo y de acuerdo con un plan de proyecto.

### **Requisitos**

Capacidad de comunicación oral y escrita en español: nivel medio; Capacidad de resolución cualitativa de problemas: nivel medio; Capacidad de trabajar en equipo: nivel medio; Capacidad de aprendizaje autónomo: nivel medio; Capacidad de obtención autónoma de información: nivel medio; Laboratorio de programación 1, 2 y 3: nivel medio; Programación orientada a objetos: nivel medio; Capacidad de comprensión de documentación técnica: nivel medio; Capacidad de comunicación oral y escrita en inglés: nivel medio;

### **Objetivos**

Que el alumno sea capaz de: gestionar proyectos software de dimensión industrial; y diseñar y construir sistemas de dicha dimensión usando técnicas actuales de desarrollo de software.

### **Contenidos:**

Introducción a la Ingeniería de Software, Planificación y gestión de proyectos, El proceso de desarrollo de software, Análisis y especificación de Requisitos, Diseño y modelado de Software, Implementación y validación, Mantenimiento de aplicaciones, Temas avanzados.

**Idioma en que se imparte:** Español

**Exámenes:** Examen final en junio y septiembre.

### **Método de evaluación:**

Convocatoria de Junio. Realización obligatoria de un proyecto, de temática libre, en equipos formados de acuerdo con las instrucciones del profesor. El proyecto supondrá el 60% de la nota. La nota del proyecto se descompondrá en una nota global del proyecto (50%) y una individual (50%). La nota individual se obtendrá como resultado del seguimiento de la actividad del estudiante en el proyecto. Este seguimiento se realizará en el laboratorio y con el uso de aplicaciones a determinar por el profesor. La nota del proyecto se obtendrá tras una serie de entregas. Las entregas se harán en , Abril y Mayo. El resto de la nota de la asignatura (40%) se obtendrá de un examen sobre la realización del proyecto. Es necesario aprobar examen y proyecto para aprobar la asignatura. Convocatoria de Septiembre. Para los que hayan hecho el proyecto, un examen en los mismos términos que en Junio. Para los que no hayan hecho el proyecto en el curso actual, además del examen, se pedirá un trabajo equivalente en forma de proyecto individual a entregar con antelación al examen. En Septiembre, el examen será un 60% de la nota y el proyecto un 40%.

### **Bibliografía:**

R. Pressman; Ingeniería del Software. Un enfoque práctico; 6ª edición, Ed. McGraw-Hill, 2005;  
I. Sommerville; Ingeniería del Software; 8 edición, Ed. Addison Wesley, 2007;  
G. Booch, J. Rumbaugh, I. Jacobson; UML. El lenguaje de modelado unificado. Guía del usuario. 2ª edición; Ed. Addison Wesley Iberoamericana, 2006.;  
E. Gamma, R. Helm, R. Johnson, y J. Vlissides; Patrones de diseño; Addison Wesley, 2003;

## **122012 - Inteligencia Artificial e Ingeniería del Conocimiento (IAIC)**

---

**Departamento:** Ingeniería del Software e Inteligencia Artificial

Duración: Anual                      Curso: 1º                      Nivel: Posgrado

**Tipo:** Obligatoria (según perfil de ingreso)

## **Profesores que imparten la asignatura:**

Héctor Gómez Gauchía

**Créditos:** 9

**Horas/semana:** 3 horas en aula, 0 en laboratorio

**Objetivos:** Introducir al alumno en los problemas y técnicas de la Inteligencia Artificial, proporcionándole una visión unificada de la misma.

**Conocimientos y destrezas que se requieren:** Lenguaje Prolog, Lenguajes de propósito general (JAVA, C++): nivel elemental; Lógica: nivel elemental; Estructuras de datos arborescentes: nivel medio; Algoritmos de exploración del espacio de estados: nivel elemental;

**Contenidos:** 1. Evolución histórica. Aspectos fundamentales. Aplicaciones. Breves nociones de Ingeniería del Conocimiento. 2. Resolución de problemas y espacio de búsqueda: Métodos ciegos. Métodos heurísticos. Búsqueda óptima. Mejora iterativa. Búsqueda con adversario. 3. Técnicas de Representación del Conocimiento. Tipos de conocimiento y tipos de representación. Lógica de predicados. Redes semánticas. Sistemas de producción. Marcos. Guiones. Estudio comparativo de las técnicas de representación. 4. Aprendizaje automático: Conceptos de aprendizaje, Aprendizaje de naturaleza inductiva, deductiva y mixta. 5. Procesamiento del lenguaje natural: Análisis léxico, sintáctico, semántico y pragmático.

**Conocimientos y destrezas que se adquieren:** Técnicas y herramientas de representación del conocimiento: nivel elemental; Métodos de búsqueda heurística y su aplicación a problemas reales: nivel medio; Técnicas de procesamiento de lenguaje natural; Algoritmos de aprendizaje automático: nivel elemental;

**Idioma en que se imparte:** Español

**Exámenes:** Primer cuatrimestre. Segundo cuatrimestre (conjunto con el final de junio). Examen final en junio y septiembre.

**Método de evaluación:** Exámenes escritos compuestos por problemas y/o cuestiones. Además, es necesaria la realización en grupo de prácticas y/o la realización individual de trabajos (no contabilizan en la nota final). Algunos trabajos que se indicarán expresamente, sí contarán, en conjunto, un 10% en la nota final (una vez aprobados los exámenes). Examen cuatrimestral en febrero. Exámenes finales en junio y septiembre. Compensable a partir de 4 (sobre 10) entre los exámenes de cuatrimestre 1 y 2.

## **Bibliografía:**

Russell, S. y Norvig, P. ; *Artificial Intelligence: A Modern Approach* ; Prentice Hall, New Jersey, Edición del 2004 en español;

Rich, E. y Knight, K.; *Artificial Intelligence*; McGraw-Hill, New York, 2ª edición, 1991 (edición en español, 1994);

S.Fdez, J. Glez, J.Mira; Problemas resueltos de Inteligencia Artificial aplicada; UNED Pearson 1998 (2005);

Gonzalo Pajares y Matilde Santos; *Inteligencia Artificial e Ingeniería del Conocimiento*; RA-MA, 2005 (Primera Edición en español) ISBN: 84-7897-676-0;

**Página web:** Campus Virtual

## **3. Recursos y Servicios de la Facultad de Informática**

### 3.1. Aulas

La Facultad de Informática dispone de 20 aulas completamente equipadas, así como una sala de grados, sala de juntas, salón de actos y sala de reuniones. Además de las aulas del edificio de la facultad durante este curso 2011-12 se dispone de 4 aulas adicionales, las 1208, 1210, 1218 y 1220 del Edificio Multiusos que se destinan a primer curso de los Estudios de Grado.

### 3.2. Laboratorios

La Facultad de Informática dispone de 12 laboratorios docentes, con un total de más de 300 ordenadores conectados a Internet, y abiertos de 9 de la mañana a 9 de la tarde los cinco días de la semana. Para proveer los diferentes servicios prestados a profesores y alumnos se dispone de 15 servidores.

Algunos de estos laboratorios albergan además entrenadores para la realización de prácticas de electrónica y diseño de circuitos, entornos de trabajo con microprocesadores, placas de diseño con FPGAs, robots, etc. En el curso 2006-07 se pusieron en servicio las pantallas informativas de los laboratorios donde se muestra información en tiempo real de uso y ocupación de los laboratorios, tanto para clases como para acceso libre de alumnos.

Se dispone de 30 portátiles y 300 maletines de componentes electrónicos para préstamo entre los alumnos. Como novedad en el curso 2011/12 se han adquirido 3 pizarras digitales interactivas portátiles eBeam que estarán a disposición de los profesores que quieran utilizarlas en los laboratorios o en las aulas.

Los laboratorios están integrados en redes de topología variable, reconfigurables por software a demanda del profesorado y adaptables de forma inmediata a las necesidades docentes. Además existe otra red aislada e independiente de máxima seguridad para entornos de prueba y formación.

Servicios a los alumnos
Cuenta de acceso a laboratorios (Windows, Linux)
Espacio de almacenamiento permanente en disco
Servicio de impresión gratuito
Servicio de mensajería on-line
Visualización de disponibilidad de recursos
Préstamo de portátiles
Préstamo de componentes electrónicos

### 3.3. Biblioteca

La Biblioteca de la Facultad de Informática forma parte de la Biblioteca Complutense de Madrid, la mayor biblioteca universitaria española y una de las más dinámicas, como lo demuestra el hecho de ser la primera fuera del ámbito anglosajón en sumarse al *Google Library Project* mediante la digitalización de sus fondos libres de derechos de autor.

Nuestra biblioteca ocupa 1.215 m<sup>2</sup> distribuidos en cuatro plantas y tiene 257 puestos de lectura. Dispone de dos salas de lectura, una mediateca con 47 puestos de consulta y tres salas de trabajo en grupo, y un depósito. Cuenta con una plantilla de 11 funcionarios y 2 becarios y el horario de apertura es de 9 a 21 horas.

Dispone de una colección de más de 22.000 volúmenes, incluyendo el fondo histórico del Centro de Cálculo de la Universidad Complutense, la biblioteca personal del profesor José García Santesmases y una biblioteca de Ciencia Ficción, así como 371 publicaciones periódicas y 4.000 CD y DVD.

Además, los fondos relacionados con Informática ocupan un lugar destacado dentro de las colecciones electrónicas de la UCM. En el campo de las publicaciones periódicas, destacan las suscripciones a la *ACM Digital Library*, a *IEEE Xplore*, a *Springer Book Series*, *Wiley InterScience* y *Elsevier Science Direct*, recursos que reúnen a las revistas de mayor impacto



académico dentro de esta especialización. En cuanto a libros electrónicos, la Biblioteca aporta a las colecciones digitales la *Computer Science Collection* de Springer, recientemente adquirida, que complementa a *Safari Books Online* (12.000 libros de contenido tecnológico).

La Biblioteca participa activamente en la *Colección Digital Complutense* mediante el volcado de los proyectos de Sistemas Informáticos, trabajos de fin de máster y de las tesis en el Archivo Institucional de *E-prints Complutense*.

El volumen de trabajo de la Biblioteca es significativo con 33.414 préstamos anuales, incluyendo material bibliográfico, audiovisual e informático (20 portátiles y 7 PC).

Entre las novedades cabe destacar la creación de sendas colecciones de películas de ciencia ficción y videojuegos y la adquisición de varios lectores de libros electrónicos (e-readers) y tabletas.

Entre las actividades realizadas por la biblioteca destacamos:

- Semana de Bienvenida a los nuevos alumnos, con una visita guiada e impartiendo un curso de formación.
- Celebración del Día del Libro con maratones de lectura en e-reader, concursos de vídeos de corta duración, etc.
- Publicación del Blog *Estado del Arte*.  
<http://www.ucm.es/BUJCM/blogs/estadodelarte/>
- Mantenimiento de una cuenta institucional en twitter: @BibliomaticaUCM  
<http://twitter.com/#!/bibliomaticaucm>
- Cursos de apoyo a la presentación de Proyectos de Sistemas Informáticos y Trabajos de Fin del Máster.
- Seguimiento mensual de los artículos publicados por personal docente e investigador de la Facultad en las principales plataformas bibliográficas.

### 3.4. Organización académica

Las tareas de gobierno y representación de las Facultades recaen, según está legislado en la Ley Orgánica de Universidades (LOU), sobre el Decano y la Junta de Facultad. El Decano es ayudado por un equipo decanal compuesto en nuestro centro por cinco vicedecanos y una secretaria académica.

<b>Decano</b>	<b>D. Daniel Mozos Muñoz</b>
<b>Vicedecana de Asuntos Económicos e Infraestructura</b>	<b>D.ª Hortensia Mecha López</b>
<b>Vicedecano de Estudios y Calidad</b>	<b>D.ª Belén Díaz Agudo</b>
<b>Vicedecano de Ordenación Académica</b>	<b>D. Rafael Caballero Roldán</b>
<b>Vicedecano de Relaciones Externas e Investigación</b>	<b>D. Manuel Prieto Matias</b>
<b>Vicedecano de Posgrado y Formación Continua</b>	<b>D. Narciso Martí Oliet</b>
<b>Secretaria de la Facultad</b>	<b>D.ª Mª Mercedes Gómez Albarrán</b>

Todas las decisiones que afectan al funcionamiento de la Facultad se toman en la Junta de Facultad. En ella participan representantes de todos los colectivos: profesores, personal de administración y servicios, y estudiantes. La Junta de la Facultad de Informática consta de 54 miembros repartidos entre los diferentes estamentos del siguiente modo:

<b>AUTORIDADES ACADÉMICAS</b>	<b>7</b>
<b>DIRECTORES DE DEPARTAMENTOS</b>	<b>3</b>
<b>PROFESORES CON VINCULACIÓN PERMANENTE A LA UNIVERSIDAD:</b>	<b>28</b>



<b>RESTO DE PERSONAL DOCENTE E INVESTIGADOR</b>	<b>4</b>
<b>PERSONAL DE ADMINISTRACIÓN Y SERVICIOS</b>	<b>4</b>
<b>ESTUDIANTES</b>	<b>8</b>

Muchas de las decisiones de la Facultad son estudiadas previamente en profundidad por diversas comisiones que presentan los resultados de sus deliberaciones a la Junta de Facultad. En la Facultad de Informática de la UCM existen las siguientes comisiones:

Comisión Permanente  
 Comisión de Ordenación Académica  
 Comisión de Estudios y Calidad  
 Comisión de Posgrado  
 Comisión Económica  
 Comisión de Investigación  
 Comisión de Biblioteca  
 Comisión de Calidad del Posgrado  
 Comisión de Calidad de los Grados  
 Comisión de Planes de Estudio del Master

### 3.5. Departamentos

---

Dpto. [Ingeniería del Software e Inteligencia Artificial](#)

Director: **D. Luis Hernández Yáñez**  
 Secretario: **D. Fernando Saenz Pérez**  
 Secretario Admvo.: **D<sup>a</sup>. Lourdes Zardaín Álvarez**  
 Despacho 402-A 4<sup>a</sup> planta  
 Teléfono: 91 394 7576 Fax: 91 394 7547  
[dp211@ucm.es](mailto:dp211@ucm.es)

Dpto. [Sistemas Informáticos y Computación](#)

Director: **D. Francisco López Fraguas**  
 Secretario: **D. Alberto Verdejo López**  
 Secretario Admvo.: **D<sup>a</sup>. Margarita Sánchez Martín**  
 Despacho 405 4<sup>a</sup> planta  
 Teléfono: 91 394 7528 Fax: 91 394 7529  
[dp210@ucm.es](mailto:dp210@ucm.es)

Dpto. [Arquitectura de Computadores y Automática](#)

Director: **D. Francisco Tirado Fernández**  
 Secretario: **D. Daniel Chaver Martínez**  
 Secretario Admvo.: **D<sup>a</sup>. Ana M<sup>a</sup> Cuenca Fernández**  
 Despacho 403 4<sup>a</sup> Planta  
 Teléfono: 91 394 7526 Fax: 91 394 7527  
[dp196@ucm.es](mailto:dp196@ucm.es)

### 3.6. Asociaciones

---

**Asociaciones de Alumnos de la Facultad de Informática:**

- **ASCII**
  - 1ª planta, Despacho 110
  - Teléfono: 91 394 7597
  - [asciifdi@gmail.com](mailto:asciifdi@gmail.com)
- **DISKÓBOLO**
  - 1ª planta, Despacho 111
  - Teléfono: 91 394 7597
  - [diskobolo.fdi@gmail.com](mailto:diskobolo.fdi@gmail.com)
- **ARCOPOLI (<http://www.arcopoli.org/>)**
  - 1ª planta, Despacho 111
  - Teléfono: 91 394 7597
  - [buzon@arcopoli.org](mailto:buzon@arcopoli.org)
- **GUEIM (<http://www.gueim.org/>)**
  - 1ª planta, Despacho 110
  - Teléfono: 91 394 7597
  - [secretario@gueim.org](mailto:secretario@gueim.org)

#### Otras asociaciones

- **IEEESB**
  - [sb.complutense@ieee.org](mailto:sb.complutense@ieee.org)
- **CLUB DEPORTIVO**
  - 1ª planta, Despacho 111
  - Teléfono: 91 394 7597
- **Asociación de Antiguos Alumnos**
  - [redantal@fdi.ucm.es](mailto:redantal@fdi.ucm.es)

### 3.7. Delegación de alumnos

---

**Delegación de alumnos de la Facultad** (<http://delegacion.fdi.ucm.es/>)

- Planta baja, Despacho 005
- Teléfono: 91 394 7597
- [delegacionfdi@gmail.com](mailto:delegacionfdi@gmail.com)

Desde Delegación de Alumnos, de forma cotidiana, se proporciona asesoramiento a los alumnos, sobre todo enfocado a la información y consejo acerca de problemas o dudas que se les puedan plantear referidos a la Facultad, a los profesores, las asignaturas o la carrera que estudien.

### 3.8. Teléfonos y direcciones de interés

---

Facultad de Informática  
 Universidad Complutense de Madrid  
 C/ Profesor José García Santesmases, s/n  
 28040 Madrid (España)  
 Teléfono: 91 394 7501  
 Fax: 91 394 7510

#### **Secretaría de Alumnos - Jefe de Sección:**

Horario: Periodo lectivo (16/9-14/6):

**L - V de 9.30 a 14 h.**

**M y J de 15:30 a 17:30 h.**

Agosto: **Cerrado**

---

D. José Antonio Macarrón Andrés  
 Teléfono: 91 394 7504-7505 Fax: 91 394 7510  
 Despacho B-001B Planta Baja

Otros períodos no lectivos:  
**L - V** de **9.30** a **14** h.

[fisalum@fdi.ucm](mailto:fisalum@fdi.ucm)

**Conserjería:**

---

D<sup>a</sup> Myriam Cortes de León  
Teléfono: 91 394 7502 Fax: 91 394 7510  
Despacho B-022A Planta Baja  
[ficonser@fdi.ucm.es](mailto:ficonser@fdi.ucm.es)

**Laboratorios - Responsable:**

---

D. Rafael Ruiz Gallego-Largo  
Teléfono/Fax: 91 394 7519  
Otros teléfonos: 91 394 7535-7520  
Despacho 202 2<sup>a</sup> Planta  
[filabor@fdi.ucm.es](mailto:filabor@fdi.ucm.es)

**Biblioteca - Director:**

---

D. Manuel Antonio Martín Mota  
Teléfono: 91 394 7523  
[mmartin@buc.ucm.es](mailto:mmartin@buc.ucm.es)

### **3.9. MIGS: Museo de Informática García Santesmases**

El Museo de Informática García Santesmases se encuentra situado en los pasillos de la Facultad de Informática de la Universidad Complutense de Madrid. Se denomina así en memoria del Profesor José García Santesmases, catedrático de esta Universidad que fue pionero en la investigación y docencia de la Informática en España.

En él se exponen máquinas desarrolladas en la UCM entre los años 1950 y 1975, así como computadoras comerciales que desde 1968 estuvieron en uso en el Centro de Cálculo de esta Universidad y equipos donados por Departamentos, particulares y otras entidades. Además del museo real, existe otro virtual, accesible vía Web, que contiene fotografías, vídeos y documentos, correspondientes a los primeros años de la Informática en nuestro país.

Durante el curso 2008-09, el museo ha ampliado sus fondos incorporando algunas nuevas colecciones que en años sucesivos irán completándose. En primer lugar, una colección de "home computers" que incluye computadores de 8 bits datados entre finales de los años 70 y mediados de los 80. En esta colección están representados sistemas tan emblemáticos como el Sinclair ZX Spectrum o el Commodore 64. En segundo lugar, una colección de sistemas Apple Macintosh que permite ver la evolución de esta familia de computadores personales desde mediados de los años 80 hasta nuestros días. Finalmente, se han iniciado colecciones temáticas de microprocesadores, memorias, sistemas y soportes de almacenamiento de datos (discos duros, floppies, etc.) y periféricos.

### **3.10. Otros servicios**

#### **Servicio de comedor y cafetería**

La facultad dispone de servicio de comedor y cafetería organizado en dos salones, uno para personal y otro para alumnos. La cafetería de alumnos está abierta en horario de 8 de la mañana a 19:30 de la tarde y tiene capacidad para 300 comensales.

#### **Espacio WIFI**

La Facultad dispone de conexión inalámbrica en todo el edificio, lo que permite a los estudiantes el uso de sus propios equipos informáticos en cualquiera de los espacios de nuestra facultad.

## Servicio de reprografía

Se dispone de un servicio de reprografía donde se pueden realizar fotocopias, impresiones de documentos digitales y encuadernaciones.

## Cajero automático

La Facultad de Informática tiene servicio de cajero automático las 24 horas del día.

## 4. Normas académicas

### 4.1. Calendario Académico del curso 2011/12

Aprobado en junta de Facultad (29/06/2011)

Primer cuatrimestre:	Del 3 de octubre de 2011 al 26 de enero de 2012, ambos inclusive
Segundo cuatrimestre:	Del 13 de febrero de 2012 al 1 de junio de 2012, ambos inclusive
Exámenes de febrero:	Del 28 de enero de 2012 al 11 de febrero de 2012, ambos inclusive
Exámenes de junio:	Del 4 de junio de 2012 al 4 de julio de 2012, ambos inclusive
Exámenes de septiembre:	Del 3 de septiembre de 2012 al 19 de septiembre de 2012, ambos inclusive

#### OBSERVACIONES:

- Vacaciones de navidad: del 22 de diciembre al 8 de enero, ambos inclusive.
- Vacaciones de semana santa: del 30 de marzo al 9 de abril, ambos inclusive.
- Festividades académicas:
  - 30 de septiembre de 2011, apertura del curso
  - 14 de noviembre de 2011, San Alberto Magno
  - 27 de enero de 2012. Santo Tomás de Aquino
- Días festivos ya establecidos por el Estado y la Comunidad Autónoma:
  - Miércoles 12 de octubre de 2011, Fiesta Nacional
  - Martes 1 de noviembre de 2011, día de todos los santos
  - Miércoles 9 de noviembre de 2011, fiesta local del municipio de Madrid
  - Martes 6 de diciembre de 2011, día de la Constitución
  - Jueves 8 de diciembre de 2011, Inmaculada Concepción
- Una vez que se publiquen en el B.O.E. y B.O.C.M. las correspondientes normas sobre el resto de días festivos, tanto de ámbito nacional como local, se comunicarán oportunamente.
- Se incorporará al calendario oficial, como día no lectivo, el puente oficial que coincida con el calendario escolar previsto para el próximo curso.

### 4.2. Normas y periodos de matrícula 2011/12

#### Proceso de admisión

La información completa sobre el proceso de admisión general de la UCM para masteres universitarios se puede consultar a través de la **Página web:**

<http://www.ucm.es/pags.php?a=menu&d=0021497>

**Requisitos:** estar en posesión de alguno de los siguientes títulos

Título Universitario **Oficial Español** o estar en condiciones de acreditarlo antes del **15 de Noviembre de 2011**.

Título expedido por una **institución del Espacio Europeo de Enseñanza Superior** (EEES) y que faculte en el país de expedición el acceso a enseñanza de master.

Título de **sistemas educativos ajenos** al EEES: **autorización o permiso de acceso** por esta Universidad o **título homologado** por el **Ministerio de Educación**.

Plazos 2011

**Primer plazo de matrícula:** del 11 al 22 de julio de 2011. Admitidos del primer plazo de inscripción que hayan abonado la reserva de plaza y los admitidos del segundo plazo de inscripción.

**Admitidos de lista de espera de julio:** 25 a 29 de julio.

**Segundo plazo de matrícula:** del 19 a 23 de septiembre. Admitidos del tercer plazo de inscripción.

**Admitidos de lista de espera de septiembre:** 26 a 30 de septiembre.

Alumnos matriculados en cursos anteriores que no hayan completado el máster: **19 a 29 de septiembre**.

- La matrícula se formalizará **preferentemente por Internet\*** siguiendo las instrucciones que figuran más abajo. Los alumnos que no tengan acceso a Internet pueden realizar la matrícula en las aulas [generales de informática](#) .

- **Matrícula presencial:** Se imprimirá el [impreso](#) y se entregará en la secretaría de alumnos del centro en que se estén cursando los estudios.

- Todos los matriculados deberán acreditar documentalmente su condición de titulados y el resto de requisitos de acceso en la Secretaría del Centro correspondiente , **antes del día 31 de octubre de 2011**.

**Se puede consultar el Procedimiento para la formalización de matrícula a través de la página web.**

### **4.3. Trabajos de Fin de Máster**

#### **Normativa para los Trabajos de Fin de Master**

Aprobada por la Comisión de Posgrado, delegada de la Junta de la Facultad de Informática, en su sesión del 1 de febrero de 2011

De acuerdo con el Real Decreto 1393/2007, de 29 de octubre, por el que se establece la ordenación de las enseñanzas universitarias oficiales, las enseñanzas del Máster de Investigación en Informática concluirán con la elaboración y defensa pública de un trabajo de fin de máster que tendrá 30 créditos.

Teniendo en cuenta la estructura del máster alrededor de cuatro especialidades, los trabajos de fin de máster se organizan en las cuatro asignaturas siguientes:

<b>Trabajo Fin de Máster</b>	<b>Departamento</b>	<b>Líneas de investigación</b>
en Ingeniería de	Arquitectura de Computadores y	Automatización del diseño de sistemas digitales, computación

Computadores	Automática	de altas prestaciones y computación en grid
en Ingeniería Informática para la Industria	Arquitectura de Computadores y Automática	Control, modelado, simulación y robótica
en Programación y Tecnología Software	Sistemas Informáticos y Computación	Métodos formales para la programación Programación declarativa multiparadigma Informática gráfica
en Sistemas Inteligentes	Ingeniería del Software e Inteligencia Artificial	Ingeniería del software e inteligencia Artificial

Según sus intereses, cada alumno admitido debe decidir en qué asignatura de estas cuatro se matricula para la realización del trabajo de fin de máster. En caso de duda, puede solicitar más información al comité de tutelaje o al coordinador del máster.

El trabajo de fin de máster consistirá en la realización de un trabajo individual y original de iniciación a la investigación bajo la dirección de un profesor del máster de la asignatura elegida. El director del trabajo de fin de máster puede ser ayudado en su labor por otro profesor del máster en calidad de codirector o por otro doctor en calidad de colaborador externo.

Para la asignación del tema de investigación, el alumno se pondrá en contacto con los diferentes profesores de la asignatura matriculada en función de sus intereses de investigación hasta llegar a un acuerdo con alguno de ellos. En la lista de profesores publicada en la página web del máster, aparece para cada profesor su departamento y su línea de investigación. Si necesita ayuda en este proceso también puede consultar al comité de tutelaje o al coordinador del máster. En el caso excepcional de no llegar a un acuerdo, se le asignaría director mediante un sorteo. Si el alumno comenzó la realización del trabajo en el curso anterior y no pudo terminarlo, puede continuar con el mismo director.

Si el alumno llega a un acuerdo para la realización del trabajo de fin de máster con un profesor de una asignatura diferente a la que se ha matriculado, puede solicitar a la Comisión de Posgrado el cambio de asignatura de trabajo de fin de máster usando la instancia disponible a este efecto en la página web del máster. El cambio requiere el visto bueno del profesor que acepta dirigir el trabajo en la nueva asignatura y, en caso de que hubiera sido asignado, el del profesor de la antigua asignatura, así como la firma de los correspondientes directores de departamento. Dichos cambios podrán solicitarse hasta el último día hábil del mes de enero.

Al comienzo del segundo cuatrimestre se completará la lista de alumnos matriculados en las cuatro asignaturas correspondientes a trabajos de fin de máster con sus directores y el título provisional del trabajo de investigación elegido.

El alumno elaborará una memoria descriptiva del proyecto realizado, con una extensión mínima recomendada de 50 páginas. La memoria se puede redactar en castellano o en inglés.

Además del cuerpo principal describiendo el proyecto realizado, la memoria contendrá:

- un resumen en inglés de media página,
- ese mismo resumen en castellano,
- una lista de no más de 10 palabras clave en inglés,
- esa misma lista en castellano,
- un índice de contenidos,
- una bibliografía, y

- o una página con la siguiente autorización de difusión, firmada por el alumno:

El/la abajo firmante, matriculado/a en el Máster en Investigación en Informática de la Facultad de Informática, autoriza a la Universidad Complutense de Madrid (UCM) a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a su autor el presente Trabajo Fin de Máster: "TÍTULO", realizado durante el curso académico 20XX-20XX bajo la dirección de XXXX [y con la colaboración externa de dirección de YYYY] en el Departamento de XXXX, y a la Biblioteca de la UCM a depositarlo en el Archivo Institucional E-Prints Complutense con el objeto de incrementar la difusión, uso e impacto del trabajo en Internet y garantizar su preservación y acceso a largo plazo.

La portada de la memoria deberá contener la siguiente información:

- o Nombre de la asignatura, que coincidirá con alguno de los siguientes:
  - Trabajo Fin de Máster en Ingeniería de Computadores
  - Trabajo Fin de Máster en Ingeniería Informática para la Industria
  - Trabajo Fin de Máster en Programación y Tecnología Software
  - Trabajo Fin de Máster en Sistemas Inteligentes
- o "Máster en Investigación en Informática, Facultad de Informática, Universidad Complutense de Madrid"
- o Título
- o Autor
- o Director(es)
- o Colaborador externo de dirección, si lo hay
- o Curso académico

El trabajo de fin de máster se podrá presentar en la convocatoria de junio o la de septiembre. En el caso de un alumno que solo tenga pendiente para el curso actual el trabajo de fin de máster y se haya matriculado en la asignatura correspondiente por segunda vez, se podrá también presentar en la convocatoria extraordinaria de febrero.

El calendario para trabajos de fin de máster establece la fecha de entrega de la memoria para cada convocatoria (junio, septiembre y la convocatoria extraordinario de febrero). Antes de dicha fecha, el profesor o uno de los profesores directores del trabajo (dando de esta forma el visto bueno a la presentación del mismo) enviará por correo electrónico, con copia al otro director del trabajo si lo hubiera, el fichero pdf de la memoria al secretario académico del departamento al que pertenezca la asignatura matriculada.

<b>Trabajo Fin de Máster</b>	<b>Departamento</b>	<b>Secretario académico</b>
...		
... en Ingeniería de Computadores	Arquitectura de Computadores y Automática	Daniel Chaver Martínez dani02@dacya.ucm.es
... en Ingeniería Informática para la Industria		
... en Programación y Tecnología Software	Sistemas Informáticos y Computación	José Alberto Verdejo López alberto@sip.ucm.es
... en Sistemas Inteligentes	Ingeniería del Software e Inteligencia Artificial	Carlos Cervigón Ruckauer ccervigon@fdi.ucm.es

En la correspondiente fecha de dicho calendario, cada departamento enviará al coordinador del máster los tribunales evaluadores de los proyectos presentados en cada convocatoria e indicará

la fecha concreta de la exposición pública oral asignada a cada trabajo, de forma que el coordinador pueda hacer pública esta información con la debida antelación.

Los tribunales de cada asignatura deberán decidir de común acuerdo si algún trabajo es merecedor de la calificación de "Matrícula de Honor", teniendo siempre en cuenta el cupo de matrículas de honor correspondiente al número de alumnos matriculados en la asignatura.

En cada convocatoria, al finalizar las defensas de todos los trabajos de cada asignatura, el director del departamento encargado de esa asignatura firmará la lista de trabajos correspondientes, con sus notas, y la entregará al coordinador del máster.

Tras la entrega de actas de cada convocatoria, en el plazo de una semana, el alumno entregará en la secretaría del departamento encargado de la correspondiente asignatura de trabajo de fin de máster un ejemplar en papel de la memoria, en tamaño A4 y encuadernado en rústica, y un ejemplar electrónico de la memoria en pdf en un cederrón. En la portada de ambos ejemplares deberán figurar, además de los datos listados anteriormente, la convocatoria y la calificación obtenida; en la versión en papel figurará asimismo la firma del director del trabajo.

Cada departamento depositará en la Biblioteca de la Facultad los ejemplares en papel y en cederrón de las memorias de los proyectos evaluados en ese departamento.

### **Calendario para trabajos de fin de máster 2011 y 2012**

---

Aprobados por la Comisión de Posgrado, delegada de la Junta de la Facultad de Informática, en su sesión del 1 de febrero de 2011

#### **Convocatoria extraordinaria de febrero 2012**

Entrega de la memoria del trabajo: hasta el 9 de febrero a las 14 horas

Publicación de los tribunales evaluadores y fechas de defensa: 10 de febrero

Defensa pública del trabajo: del 27 al 29 de febrero

Entrega de actas: 29 de febrero

Entrega del trabajo en rústica y cederrón: hasta el 7 de marzo

#### **Convocatoria de junio 2012**

Entrega de la memoria del trabajo: hasta el 21 de junio a las 14 horas

Publicación de los tribunales evaluadores y fechas de defensa: 22 de junio

Defensa pública del trabajo: del 9 al 11 de julio

Entrega de actas: 11 de julio

Entrega del trabajo en rústica y cederrón: hasta el 18 de julio

#### **Convocatoria de septiembre 2012**

Entrega de la memoria del trabajo: hasta el 10 de septiembre a las 14 horas

Publicación de los tribunales evaluadores y fechas de defensa: 11 de septiembre

Defensa pública del trabajo: del 26 al 28 de septiembre

Entrega de actas: 28 de septiembre

Entrega del trabajo en rústica y cederrón: hasta el 5 de octubre



#### 4.4. Reglamento de Laboratorios

---

La Facultad de Informática dispone de 12 laboratorios (11 situados en la segunda planta del edificio principal y 1 situado en la primera planta de la biblioteca) que deberán usarse según las siguientes condiciones:

1. Los laboratorios pueden ser usados únicamente por alumnos de la Facultad de Informática, **con fines exclusivamente académicos** y para la realización de turnos de prácticas regladas o turnos de prácticas no tuteladas.
2. Los **turnos de prácticas regladas** son actividades docentes coordinadas y supervisadas por un profesor. Tienen un horario prefijado y una duración de al menos **2 horas**. Durante estos turnos el **responsable del laboratorio es el profesor**, y **el acceso al laboratorio está restringido** a aquellos alumnos que tengan asignado el turno.
3. Los **turnos de prácticas no tuteladas** son periodos durante los que el alumno puede hacer uso de un puesto de laboratorio sin supervisión de un profesor. Tienen una **duración máxima de 1 hora, prorrogable si no hay demanda**. No tienen una hora de comienzo determinada pero **finalizan a las horas en punto**. Durante estos turnos el **responsable del laboratorio es el personal técnico** y **el acceso a los laboratorios es libre**.
4. Todo alumno que desee hacer uso de un turno de prácticas, incluso cuando no requiera el uso del ordenador, tiene la obligación de identificarse abriendo sesión en el puesto que elija. En caso de trabajo en grupo bastará con la identificación de uno de los miembros. Para ello dispone de una cuenta personal por sistema operativo, accesible desde cualquier puesto de la red de laboratorios. Además deberá tener disponible en todo momento algún documento que le identifique como alumno del Centro y que le podrá ser requerido por el personal técnico.
5. Cada laboratorio tiene **20/40 puestos** conectados en red y cada puesto puede ser ocupado por un **máximo de 2 alumnos**.
6. El horario de apertura de los laboratorios, salvo en periodos no lectivos, es de **9:00h a 21:00h de lunes a viernes**.

Como normas generales se recuerda que:

- Los laboratorios son un lugar de estudio, por lo que se ruega silencio.
- Está prohibido introducir comidas o bebidas en los laboratorios.
- Está prohibido instalar software en los puestos así como modificar su configuración.
- Está prohibido manipular las impresoras.
- Está prohibido imprimir material no académico, y durante los turnos de prácticas regladas está prohibido imprimir material no relacionado con la asignatura en curso.
- Cualquier material entregado al alumno debe ser devuelto al finalizar el turno. Si detecta en él cualquier deficiencia debe comunicarlo al personal técnico
- El personal técnico de los laboratorios podrá monitorizar toda actividad realizada dentro de sesiones abiertas en los laboratorios para garantizar el máximo aprovechamiento académico de los recursos disponibles.

En caso de incumplimiento de estas normas, el personal técnico podrá impedir al infractor el uso de su cuenta en turnos de prácticas no tuteladas y lo pondrá en conocimiento del Decanato para que tome las medidas disciplinarias oportunas.

#### 4.5. Reglamento de Biblioteca

---

La Biblioteca de la Facultad de Informática se rige por el [Reglamento de la BUCM](#) y las [Normas de Sala y Préstamo de la BUCM](#). El horario de apertura de la Biblioteca de la Facultad durante los periodos lectivos es de **9:00h a 21:00h de lunes a viernes**. La Biblioteca dispone de una Mediateca y 3 Salas para Trabajo en grupo y 2 Salas de Lectura. Existen una serie de **normas generales** para todas las salas:

- Todas las salas de la Biblioteca pueden ser usadas por alumnos de la Facultad de Informática, **con fines exclusivamente académicos**.

- Se trata de lugares de estudio, por lo que se ruega silencio. Asimismo se ruega desconectar cualquier dispositivo que pueda producir ruido.
- Está prohibido fumar, comer o beber.
- Está **prohibido instalar software** en los puestos así como modificar su configuración.

### Mediateca y Salas de Trabajo en Grupo

La Mediateca y Salas de Trabajo en Grupo están situadas en la primera planta de la Biblioteca y comparten ubicación con el Laboratorio 12. Hay **12 puestos de lectura** (de acceso libre), **7 puestos informáticos individuales** y **3 salas** para trabajo en grupo con **6 puestos** cada una. Cada sala de trabajo puede ser usada de forma exclusiva por un grupo de entre 3 y 6 alumnos. También puede ser usada por un grupo de 2 alumnos pero, en este caso, podrá incorporarse posteriormente otro grupo de 2 ó 3 alumnos para que compartan la misma sala. El uso de los puestos de Mediateca y Salas de Trabajo en Grupo se rige por las siguientes normas:

El uso de los puestos que no son de libre acceso se organiza por **turnos** que tienen una duración **máxima de 1 h** (salas) y **1/2 h** (puestos informáticos), **prorrogable si no hay demanda**. No tienen una hora de comienzo determinada pero **finalizan a las horas y a las medias en punto**.

Para hacer **uso de un turno**, el alumno debe **entregar su carné de estudiante** al personal de la biblioteca para que éste le asigne un puesto. El carné le será devuelto al finalizar el turno.

Se podrán **reservar turnos de uso no consecutivos con 1 día lectivo de antelación** como máximo. Se anularán aquellas reservas no hechas efectivas en los primeros **10 min** del turno.

### Salas de Lectura

Las Salas de Lectura están situadas en la segunda y tercera planta de la Biblioteca. Estas Salas de Lectura podrán usarse según las siguientes condiciones:

La Sala de Lectura de la segunda planta dispone de **92 puestos individuales**. La Sala de Lectura de la tercera planta dispone de **88 puestos individuales**. Por motivos de seguridad, una vez completado el aforo no se permitirá el acceso de más usuarios.

El usuario **no deberá ausentarse** de su puesto de lectura por un periodo superior a **20 min**. Transcurrido este tiempo, el personal de la biblioteca podrá retirar sus efectos personales para liberar el puesto.

Todos los libros son de **libre acceso en la Sala**. Una vez consultados, deben ser **devueltos a los carritos o mesas auxiliares**, nunca a las estanterías.

### 4.6. Normas básicas de convivencia en la Facultad de Informática

La Facultad de Informática es un espacio de trabajo y convivencia entre personas con diferentes funciones. El objetivo de la Facultad es el desarrollo de tareas docentes e investigadoras en el mejor entorno posible. Por ello las diferentes personas implicadas deberán observar unas normas básicas para el buen desarrollo de estas tareas. Entre estas normas básicas de convivencia nos gustaría destacar las siguientes, sin obviar cualquier otra que el sentido común pueda determinar:

- Normas básicas de conducta en clase (aula y laboratorio)
  - El comportamiento tanto en aulas, laboratorios, como en el resto de la Facultad debe facilitar la convivencia y el respeto mutuo entre profesores, alumnos y resto de personal de la Facultad.
  - No se puede comer ni beber en clase.

- Los teléfonos móviles deberán estar desconectados o mantenidos en silencio, no estando permitido atender llamadas ni mensajes en las aulas, ni realizar fotografías o grabaciones.
- En las aulas, laboratorios y pasillos se debe mantener silencio durante las horas de clase.
- Todos los usuarios deben respetar y velar por el buen uso y cuidado del material del aula o laboratorio.
- No se puede fumar en ninguna zona de la Facultad.
- Las horas de clase están dedicadas a la docencia por lo que no se deben utilizar para otras actividades como leer el periódico, realizar pasatiempos, etc.
- Asistencia a clase y puntualidad
  - Se deben respetar los horarios de las clases y asistir con puntualidad a las mismas. Las clases comienzan a las horas en punto (excepto en el doble grado que lo hacen a las medias) y tienen una duración de 50 minutos. El profesor tiene potestad para impedir el acceso al aula a los alumnos que no lleguen a tiempo a clase.
  - No se puede abandonar el aula durante la clase salvo motivo de causa mayor.
  - El control de la asistencia de los alumnos lo determinará en cada caso el profesor de la asignatura. La asistencia se considera fundamental para obtener buenos resultados y en muchos casos puede formar parte del método de evaluación.
- Evaluaciones
  - Los diversos métodos de evaluación realizados en las asignaturas implican el desarrollo de algún trabajo o examen de carácter individual o en grupo. En todos los casos, el trabajo deberá ser realizado por el alumno o grupo de alumnos de modo original. Por tanto, cuando se descubra que un alumno ha copiado se le aplicarán las medidas de disciplina académica fijadas por la legislación vigente.

#### 4.7. Defensor del universitario

---

El [Estatuto del Estudiante](#) es el texto que regula los derechos y deberes de los estudiantes de la Universidad Complutense de Madrid. En particular este texto incluye un Título que regula la figura del Defensor del Universitario.

El Defensor Universitario es el órgano institucional, elegido por el Claustro, para velar por el respeto a los derechos y las libertades de todos los miembros de la Comunidad Universitaria: **profesorado, alumnado, y personal de administración y servicios**, con la finalidad de contribuir a la mejora de la calidad y el buen funcionamiento de la Universidad. Su actuación se desarrolla bajo los principios de **independencia, autonomía y confidencialidad**, no estando sometido a mandato imperativo alguno en el desarrollo de sus competencias.

La actuación del Defensor Universitario está dirigida hacia la mejora de la convivencia y calidad universitaria en todos sus ámbitos, **investigando**, ya sea de oficio o a instancia de parte, las reclamaciones, quejas y consultas que se registren en la Oficina del Defensor. Así mismo es competencia del Defensor, siempre que así lo acepten las partes, actuar como **mediador** para dirimir los desacuerdos que se produzcan, sobre temas universitarios, entre los miembros de la Comunidad Universitaria.

**Oficina del Defensor Universitario**  
**Universidad Complutense de Madrid**  
**Calle Isaac Peral, nº 1 Planta 7ª B**  
**(Residencia de Profesores) 28015 Madrid**