

**GUÍA DOCENTE  
EXPERIENCIA PILOTO DE CRÉDITOS EUROPEOS  
UNIVERSIDADES ANDALUZAS**

**DATOS BÁSICOS DE LA ASIGNATURA**

NOMBRE: Lenguajes de Inteligencia Artificial

CÓDIGO: **6130035**

AÑO DE PLAN DE ESTUDIO: 1999

TIPO (troncal/obligatoria/optativa) : optativa

|  |                            |                               |
|--|----------------------------|-------------------------------|
| Créditos totales (LRU / ECTS): 4,5 / 4 | Créditos teóricos: 3 / 2,5 | Créditos prácticos: 1'5 / 1'5 |
|--|----------------------------|-------------------------------|

CURSO: 2º

CUATRIMESTRE: 1º

CICLO: 1º

TITULACIÓN: INGENIERÍA TÉCNICA EN INFORMÁTICA DE GESTIÓN

CURSO ACADÉMICO: 2013 – 2014

**DATOS BÁSICOS DEL PROFESOR**

NOMBRE: Nicolás Luis Fernández García

CENTRO/DEPARTAMENTO: E.P.S. / Informática y Análisis Numérico

ÁREA: Ciencias de la Computación e Inteligencia Artificial

Nº DESPACHO:

Albert Einstein,  
Tercera planta  
Pasillo sur

Correo electrónico:

ma1fegan@uco.es

Teléfono:

957 21 83 47

URL WEB: [www.uco.es/users/ma1fegan](http://www.uco.es/users/ma1fegan)

**DATOS ESPECÍFICOS DE LA ASIGNATURA**

**1. DESCRIPTOR**

- Introducción a la inteligencia artificial
- Programación funcional
- Programación lógica

## **2. SITUACIÓN**

### **2.1. PRERREQUISITOS:**

- No se establecen prerrequisitos.

### **2.2. CONTEXTO DENTRO DE LA TITULACIÓN:**

- Esta asignatura está relacionada con:
  - Metodología y tecnología de la programación (primer curso)
  - Matemáticas I (primer curso)
  - Matemática II (primer curso)
  - Estructura de datos y la información (segundo curso)
  - Informática aplicada (segundo curso)

### **2.3. RECOMENDACIONES:**

- Se deben tener conocimientos sobre:
  - Diseño de algoritmos: explicados en la asignatura de primer curso denominada “Metodología y Tecnología de la Programación”.
  - Lógica matemática: explicados en la asignatura de primer curso denominada “Matemáticas II”.

## **3. COMPETENCIAS**

### **3.1. COMPETENCIAS TRANSVERSALES/GENÉRICAS:**

- Aprendizaje autónomo.
- Capacidad de aplicar los conocimientos a la práctica.
- Resolución de problemas

### **3.2. COMPETENCIAS ESPECÍFICAS:**

- **Cognitivas (Saber):**
  - Tipos de programación declarativa
  - Conocer las características de la programación funcional
  - Conocer las características de la programación lógica
- **Procedimentales/Instrumentales (Saber hacer):**
  - Utilizar un intérprete de Scheme
  - Utilizar un intérprete de Prolog
- **Actitudinales (Ser):**
  - Estudio
  - Interés por desarrollar las prácticas con calidad
  - Interés por investigar y buscar soluciones alternativas

#### **4. OBJETIVOS**

- Comprender los conceptos principales de la programación funcional y de la programación lógica
- Conocer las características básicas y fundamentales de un lenguaje de programación funcional (*Scheme*) y de un lenguaje de programación lógica (*Prolog*).

## 5. METODOLOGÍA

**NÚMERO DE HORAS DE TRABAJO DEL ALUMNO:**

**NÚMERO DE HORAS DE TRABAJO DEL ALUMNO: 102**

- Esta asignatura está en vías de extinción y, por tanto, no tiene clases presenciales, pero sí tiene tutorías individuales y evaluación.

**6. TÉCNICAS DOCENTES** (señale con una X las técnicas que va a utilizar en el desarrollo de su asignatura. Puede señalar más de una. También puede sustituirlas por otras):

|                            |                               |                            |
|----------------------------|-------------------------------|----------------------------|
| Explicaciones de teoría    | Trabajos de teoría            | Tutorías individuales<br>X |
| Explicaciones de prácticas | Elaboración de prácticas<br>X | Realización de ejercicios  |

### DESARROLLO Y JUSTIFICACIÓN:

- Elaboración de las prácticas:
  - Los alumnos realizarán cuatro prácticas de Scheme y una de Prolog.
  - En las horas de tutoría individual, se explicarán los recursos informáticos necesarios para el desarrollo de las prácticas: intérpretes de drscheme y swi-prolog.
  - Se describirán de los objetivos que se pretenden conseguir con la elaboración de cada una de las prácticas.
  - Los alumnos desarrollarán las prácticas codificando y documentando los programas.
- Corrección de las prácticas:
  - Se establecerá una fecha máxima para la entrega de cada práctica.
  - El profesor corregirá y evaluará, en presencia de cada alumno/a, los trabajos de prácticas que haya realizado.
  - El profesor indicará los posibles fallos y propondrá posibles soluciones alternativas.
- Tutorías individualizadas:
  - Los alumnos deben utilizar estas tutorías a lo largo de todo el curso y no sólo antes de la fecha del examen.
  - El profesor intentará resolver las dudas particulares que pueda tener cada alumno en relación con los temas de teoría y las prácticas.

**7. BLOQUES TEMÁTICOS** (dividir el temario en grandes bloques temáticos; no hay número mínimo ni máximo)

- Programación funcional: Scheme
- Programación lógica: Prolog

## 8. BIBLIOGRAFÍA

### 8.1 GENERAL

- **Lenguaje Scheme**

- Abelson, H., Sussman, G. J. y Sussman, J. “Structure and Interpretations of Computers Programs”. Second edition. The MIT Electrical Engineering and Computers Science Series, 1996. ISBN: 0-262-01153-0.
- Kelsey, R., Clinger, W, Rees, J. y otros: “Revised5 Report on the Algorithmic Language Scheme”, 1998.
  - <http://www.uco.es/users/malfeagan/Comunes/manuales/lia/r5rs.pdf>

- **Lenguaje Prolog**

- Mellish, C. S., Cloksin, W. F. “Programming in PROLOG”. Springer – Verlag, 2003, ISBN: 978-3-540-00678-7.
- Shapiro, E. y Sterling, L. “The Art of Prolog”. Second edition. MIT Press, 1994. ISBN: 0-262-19338-8.

### 8.2 ESPECÍFICA (

- **Lenguaje Scheme**

- Abelson, H., Sussman, G. J. y Sussman, J. “Structure and Interpretations of Computers Programs”. The MIT Electrical Engineering and Computers Science Series, 1993. ISBN: 0-262-01077-1.
- Dybvig, R. Kent. “The Scheme Programming Language” Second Edition. Ansi Scheme. Prentice Hall, 1996. ISBN: 0-13-454646-6.
- Friedman, D. P. y Felleisen, M. “The Seasoned Schemer”. MIT Press, 1996 ISBN: 0-262-56100-X.
- Grilmeyer, O. “Exploring Computer Science with Scheme”. Springer Verlag, 1999. ISBN: 0-387-94895-3.
- Harvey, B. y Wright, M. “Simply Scheme: Introducing Computer Science”. The MIT Press, 1994. ISBN: 0-262-08226-8.
- Pearce, Jon. “Programming and Meta-Programming in Scheme”. Springer –Verlag, 1998. ISBN: 0-387-98320-1.
- Waston, Mark. “Programming in Scheme. Learn Scheme Through Artificial Intelligence Programs”. Springer - Verlag, 1996. ISBN: 0-387-94681-0.

- **Lenguaje Prolog**

- Bramer, M. “Logic Programming with Prolog”. Springer, 2005. ISBN-10: 1-85233-938-2. ISBN-13: 978-1852-33938-8.
- Bratko, I. “PROLOG: Programming for Artificial Intelligence”, Segunda edición. Addison-Wesley, 1994. ISBN: 0-201-41606-9.
- Giannesini, F.; Kanoui, H; Pasero, R. y van Caneghem, M. “Prolog”. Addison Wesley Iberoamericana, 1989. ISBN: 84-7829-003-6.
- Mellish, C. S., Cloksin, W.F. “Programming in PROLOG”. Springer - Verlag, 1994. ISBN: 3-540-58350-5.
- Mellish, C. S., Cloksin, W.F. “Programación en PROLOG”. Editorial Gustavo Gili, S. A., 1987. ISBN: 84-252-1339-8.

## 9. TÉCNICAS DE EVALUACIÓN

- Herramientas de evaluación
  - Prácticas de la asignatura
    - Los alumnos realizarán cuatro prácticas de Scheme y una de Prolog
    - Se establecerá una fecha máxima para la entrega de cada práctica.
  - Examen final
    - Consistirá en un examen escrito en el que deberá codificar ejercicios de Scheme y Prolog

### **Criterios de evaluación y calificación** *(referidos a las competencias trabajadas durante el curso):*

- En la convocatoria de febrero, la calificación final se calculará mediante la siguiente expresión aritmética

$$\text{Calificación final} = \text{Prácticas (60 \%)} + \text{Examen final (40 \%)}$$

- Observación: será necesario obtener una puntuación igual o superior a 5,0 puntos en el examen final para poder aprobar la asignatura
- En las convocatorias extraordinarias de septiembre, diciembre o enero, la evaluación de la asignatura se realizará exclusivamente mediante un examen final.

## 10. TEMARIO DESARROLLADO

### Primera parte: Scheme

1. Introducción al lenguaje Scheme
  - Características fundamentales de la programación funcional
  - Reseña histórica de Scheme
    - Lisp
    - Compilación e interpretación
    - Ámbito dinámico y ámbito léxico o estático
    - Origen de Scheme
2. Expresiones y Funciones
  - Elementos básicos de Scheme
    - Caracteres
    - Cadenas de caracteres
    - Números
    - Identificadores: variables y palabras reservadas
  - Expresiones
    - Literales
    - Variables
    - Operadores aritméticos
  - Definición de función
  - Reglas de ámbito léxico
    - Definiciones internas
    - Estructuras de bloques
  - Funciones anónimas: la forma especial “lambda”
  - Las formas especiales “let” y “let\*”
3. Predicados y sentencias condicionales
  - Operadores relacionales
  - Operadores lógicos
  - Predicados primitivos
    - Predicados simbólicos
    - Predicados numéricos
    - Predicados de equivalencia
  - Formas especiales condicionales:
    - Forma especial “if”
    - Forma especial “cond”
    - Forma especial “case”
4. Iteración y recursión
  - Forma especial iterativa “do”
  - Recursión simple
  - Recursión múltiple
  - Recursión de cola
  - Forma especial “let con nombre”
  - Funciones utilizadas como parámetros

- Funciones devueltas como resultados
5. Tipos compuestos de datos
    - Vector
    - Operaciones con vectores
    - Pares
    - Listas
    - Operaciones con listas
    - Conversión de listas y vectores
    - Funciones con argumentos opcionales
  6. Abstracción de datos
    - Definición de tipo abstracto de datos
    - Representación de tipos abstractos mediante vectores
    - Representación de tipos abstractos mediante listas
    - Representación de tipos abstractos mediante listas de asociación
  7. Lectura y escritura
    - Interacción con el sistema
    - Apertura y cierre de un fichero
    - Lectura
    - Escritura
    - Interacción entre funciones y ficheros

## **Segunda parte: Prolog**

8. Introducción al lenguaje Prolog
  - Características fundamentales de la programación lógica
  - Hechos
  - Preguntas
  - Variables
  - Conjunción, disyunción y negación
  - Reglas
9. Elementos básicos del Prolog
  - Elementos simples y compuestos
  - Aritmética
  - Igualdad y coincidencia
  - Operadores relacionales
10. Listas
  - Descripción
  - Operaciones con listas
11. La reevaluación y “el corte”
  - Generación de soluciones múltiples
  - Descripción del “corte”
  - Aplicaciones y problemas del corte



## 12. Entrada y salida

- Lectura y escritura de términos
- Lectura y escritura de caracteres
- Lectura y escritura con ficheros