

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

**DATOS BÁSICOS DE LA ASIGNATURA**

NOMBRE: Teoría de Automatas y Lenguajes Formales

CÓDIGO: 6230012

AÑO DE PLAN DE ESTUDIO: 1999

TIPO (troncal/obligatoria/optativa) : Troncal

Créditos totales (LRU / ECTS): 7,5	Créditos teóricos: 4,5	Créditos prácticos: 3
------------------------------------	------------------------	-----------------------

CURSO: 2º

CUATRIMESTRE: 2º

CICLO: 1º

TITULACIÓN: INGENIERÍA TÉCNICA EN INFORMÁTICA DE SISTEMAS

**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**

- Máquinas secuenciales y autómatas finitos.
- Máquinas de Turing.
- Funciones recursivas.
- Gramáticas y lenguajes formales.
- Redes neuronales.

## **2. SITUACIÓN**

### **2.1. PRERREQUISITOS:**

- No se establecen prerrequisitos.

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

- Esta asignatura está relacionada con las siguientes:
  - Metodología y Tecnología de la Información (Primer curso).
  - Estructura de Datos y de la Información (Segundo curso).
  - Informática Aplicada (Segundo curso).

### **2.3. RECOMENDACIONES:**

- Se deben tener conocimientos sobre:
  - Fundamentos de lógica, grafos y árboles (Matemáticas II de primer curso).
  - Sistema operativo Unix y programación en C (Metodología y Tecnología de la Programación de 1<sup>er</sup> curso).

## **3. COMPETENCIAS**

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

- Aprendizaje autónomo.
- Capacidad de análisis y síntesis.
- Capacidad de aplicar los conocimientos a la práctica.
- Resolución de problemas.
- Trabajo individual o por parejas.
- Comunicación oral y escrita.

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

- **Cognitivas (Saber):**
  - Conocimientos sobre la teoría de lenguajes formales.
  - Conocimiento de los fundamentos de las redes neuronales artificiales como modelo de aprendizaje conexionista.
- **Procedimentales/Instrumentales (Saber hacer):**
  - Usar herramientas de generación de analizadores léxicos y sintácticos.
  - Especificar expresiones regulares para componentes léxicos.
  - Construir autómatas y máquinas de Turing
  - Diseñar gramáticas que generan lenguajes formales.
- **Actitudinales (Ser):**
  - Calidad
  - Toma de decisión
  - Capacidad de iniciativa y participación
  - Interés por investigar y buscar soluciones a nuevos problemas.

#### **4. OBJETIVOS**

- Describir los fundamentos matemáticos de la informática.
- Presentar el concepto de lenguaje formal.
- Conocer las equivalencias entre los lenguajes formales, las gramáticas formales, las expresiones regulares y las máquinas reconocedoras (máquinas de Turing y autómatas).
- Mostrar la relación entre los lenguajes formales y los lenguajes de programación.
- Diseñar un intérprete de un lenguaje de programación.
- Presentar la máquina de Turing como una máquina de cómputo universal.
- Definir el concepto de función recursiva y mostrar su relación con las máquinas de Turing.
- Introducir el concepto de red neuronal artificial como modelo de aprendizaje conexionista.

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

- **PROGRAMA DE TEORÍA.**

- PARTE I: Introducción.
- PARTE II: Lenguajes regulares.
- PARTE III: Lenguajes de contexto libre.
- PARTE IV: Máquinas de Turing.

- **PROGRAMA DE PRÁCTICAS.**

- Introducción a Lex (Flex): Análisis léxico.
- Introducción a Yacc (Bison): Análisis sintáctico y semántico.

**6. BIBLIOGRAFÍA**

**6.1 GENERAL**

- Alfonseca, M.; Sancho, J. y Martínez Orga, M. “Teoría de Lenguajes, Gramáticas y Autómatas”. Publicaciones R.A.E.C. Colección Textos de Cátedra, 1997. ISBN: 84-605-6092-9.
- Brookshear, J. G. “Teoría de la Computación: Lenguajes Formales, Autómatas y Complejidad”. Addison-Wesley Iberoamericana, 1993. ISBN: 0-201-60119-2.
- Hopcroft, J. E.; Motwani R. y Ullman, J. D. “Introducción a la teoría de autómatas, lenguajes y computación”. Segunda edición. Addison-Wesley, 2002. ISBN: 0-201-44124-1.
- Isasi, P.; Martínez, P. y Borrajo, D. “Lenguajes, gramáticas y autómatas. Un enfoque práctico”. Addison – Wesley, 1.997. 84-7829-014-1.
- Kelly, D. “Teoría de Autómatas y Lenguajes Formales”. Prentice Hall, 1995. ISBN: 0-13-497777-7.
- Martin, J. C. “Introduction to Languages and the Theory of Computation”. Tercera Edición. McGraw-Hill, 2.003. ISBN: 0-07-119854-7 (International ed.).

**6.2 ESPECÍFICA**

- Aho, A. V. et al. “Compiladores: Principios, Técnicas y Herramientas”. Segunda Edición, Pearson y Addison - Wesley, 2008. ISBN: 978-970-26-1133-2.
- Cutland, N. J. “Computability: An Introduction to Recursive Function Theory”. Cambridge University Press, 1.980. ISBN: 0-521-29-465-7.
- Du, Ding-Zhu y Ko, Ker. “Problem Solving in automata, languages and complexity”. John Wiley & Sons, 2.001. ISBN: 0-471-43960-6.
- Fernández, G y Sáez Vacas, F. “Fundamentos de Informática: Lógica, Autómatas, Algoritmos y Lenguajes”. Anaya Multimedia, 1.995. ISBN: 84-7614-792-9.
- Freeman, J. A y Skapura, D. M. “Redes Neuronales: Algoritmos, Aplicaciones y Técnicas de Programación”. Addison-Wesley/Díaz de Santos, 1.993. ISBN: 0-201-60115-X.
- Hamburger, Henry. “Logic and language models for computer science”. Prentice Hall, 2.002. ISBN: 0-13-065487-6.

- Harrison, Michael A. "Introduction to Formal Language Theory". Addison-Wesley, 1.978. ISBN: 0-201-02955-3.
- Hiler, J. R. y Martínez, V. J. "Redes Neuronales Artificiales: Fundamentos, Modelos y Aplicaciones". RA-MA Editorial, 1.995. ISBN: 84-7897-155-6.
- Hopcroft, J. E. y Ullman, D. U. "Introduction to Automata Theory, Languages and Computation". Addison-Wesley, 1.979. ISBN: 0-201-02988-X.
- Redes de neuronas artificiales: un enfoque práctico/ Pedro Isasi Viñuela, Inés M. Galván León. Publicación Madrid: Pearson Educacion, 2004 Descr. física XI, 229 p. : il. ; 24 cm ISBN 8420540250
- Kernighan, B. W. y Pike, R. "El entorno de programación Unix". Prentice-Hall, 1.994. Software Series, 1.984. ISBN: 0-3-937681-X
- Khoussainov, Bakhadyr. "Automata theory and its applications". Boston: Birkhäuser, cop. 2001 ISBN: 0-8176-4207-2.
- Kimber, Efim y Smith Carl. "Theory of computing. A gentle introduction". Prentice Hall, 2.001. ISBN: 0-13-027961-7.
- Levine, J. R.; Mason, T. y Brown, D. "Lex & Yacc". O'Reilly & Associates, 1.995. ISBN: 1565920007.
- Lewis, H. R y Papadimitriou, C. H. "Elements of the Theory of Computation". Prentice Hall, 1.981. ISBN: 0-13-273426-5.
- Sudkamp, Thomas A. "Languages and machines". Second Edition. Addison Wesley, 1998. ISBN: 0-201-82136-2.
- Wasserman, P. D. "Neural Computing: Theory and Practice". Van Nostrand Reinhold/Díaz de Santos, 1.989. ISBN: 0-442-20743-3

## 7. TÉCNICAS DE EVALUACIÓN

- La evaluación de la asignatura se realizará mediante:
  - Examen final de teoría: examen escrito compuesto por:
    - Preguntas de Teoría.
    - Ejercicios de problemas.
  - Trabajos de prácticas.
  - Trabajo de teoría.

### Criterios de evaluación y calificación:

- La evaluación de la asignatura estará basada en
  - Examen escrito de teoría
  - Trabajos de prácticas: trabajo de lex (flex) y trabajo de yacc (bison)
  - Trabajos de teoría
- La “calificación final” de la asignatura se obtendrá aplicando la siguiente expresión aritmética:  
**Calificación final** = ( 1 Trabajo de teoría  
+ 3 Examen escrito de teoría  
+ 2 Prácticas  
)/ 6
- Para poder aprobar la asignatura es necesario obtener “una calificación final” superior o igual a cinco (5) puntos, exigiéndose, además, que
  - La nota del examen escrito de teoría sea igual o superior a cuatro (4) puntos.
  - Se hayan realizado correctamente los trabajos de teoría y de prácticas
- La nota del trabajo de teoría se obtendrá al partir de la documentación elaborada.
- La nota de prácticas se obtendrá a partir de los trabajos correctamente realizados:
  - Trabajo de lex (flex)
  - Trabajo de YACC (bison)
- Se establecerá una fecha máxima para la entrega de las prácticas y del trabajo de teoría.

## **8. TEMARIO DESARROLLADO**

### **PROGRAMA DE TEORÍA**

#### **Parte I: INTRODUCCIÓN**

##### 1. Introducción

- Lenguajes y gramáticas formales
- Autómatas finitos y máquinas de Turing
- Funciones recursivas
- Redes neuronales

##### 2. Lenguajes formales

- Definiciones: palabras y lenguajes formales
- Operaciones con palabras
- Operaciones con lenguajes

##### 3. Gramáticas formales

- Introducción
- Definición de gramática formal
- Tipos de Gramáticas: jerarquía de Chomsky
- Generación de palabras: derivaciones
- Lenguaje generado por una gramática

#### **Parte II: LENGUAJES REGULARES**

##### 4. Gramáticas regulares

- Definiciones
- Equivalencia entre gramáticas regulares y lineales
- Árbol de derivación
- Operaciones con gramáticas regulares

##### 5. Expresiones regulares

- Definición
- Lenguaje denotado por una expresión regular
- Propiedades de las expresiones regulares
- Derivada de una expresión regular
- Gramática regular equivalente a una expresión regular

##### 6. Autómatas finitos y máquinas secuenciales

- Autómata finito determinista (AFD)
  - Definición
  - Representación gráfica
  - Lenguaje reconocido por un AFD
  - Autómata conexo: estados accesibles
  - Estados que no finalizan
  - Minimización de autómatas: autómata cociente
- Autómata finito no determinista (AFN):

- Definición
- Representación gráfica
- Lenguaje reconocido por un AFN
- Autómata finito determinista equivalente a un autómata finito no determinista
- Equivalencia entre autómatas finitos y gramáticas regulares
- Equivalencia entre autómatas finitos y expresiones regulares:
  - El problema de síntesis
  - El problema de análisis
- Máquinas secuenciales:
  - Máquina secuencial de Mealy
  - Máquina secuencial de Moore
  - Equivalencia entre máquinas secuenciales
  - Máquina secuencial generalizada

#### 7. Lenguajes Regulares

- Definición
- Lema de bombeo de los lenguajes regulares
- Propiedades de los lenguajes regulares
- Algoritmos de decisión de los lenguajes regulares
- Aplicaciones de los lenguajes regulares

### **Parte III: LENGUAJES DE CONTEXTO LIBRE**

#### 8. Gramáticas de contexto libre

- Definición
- Derivación en una gramática de contexto libre
- Árbol de derivación
- Lenguaje generado por una gramática de contexto libre
- Ambigüedad
- Vacuidad de una gramática de contexto libre
- Símbolos útiles e inútiles
- Reglas épsilon y reglas unitarias
- Recursividad por la izquierda
- Factorización por la izquierda
- Formas normales de Chomsky y Greibach

#### 9. Autómatas con pila

- Definición
- Representación gráfica
- Configuraciones de un autómata con pila
- Lenguaje reconocido por un autómata con pila
- Equivalencia entre autómatas con pila
- Equivalencia entre autómatas con pila y gramáticas de contexto libre

#### 10. Lenguajes de contexto libre

- Definición
- El lema de bombeo de los lenguajes de contexto libre



- Los lenguajes de contexto libre en la jerarquía de los lenguajes formales
- Propiedades de los lenguajes de contexto libre
- Algoritmos de decisión
- Aplicaciones de los lenguajes de contexto libre

#### **Parte IV: MÁQUINAS DE TURING**

##### 11. Máquinas de Turing

- Definición
- Construcción de máquinas de Turing
- Computación de una función con una máquina de Turing
- Función característica de un lenguaje
- Variantes de la máquina de Turing:
  - Extensiones
  - Restricciones
  - Máquina universal de Turing

#### **PROGRAMA DE PRÁCTICAS**

Se utilizarán un generador de analizadores léxicos y un generador de analizadores sintácticos que permitirán mostrar la relación existente entre los conceptos teóricos de la asignatura y el diseño de intérpretes y compiladores.

El programa de prácticas está compuesto por los siguientes temas:

- Lex (flex): generador de analizadores léxicos
- YACC (bison): generador de los analizadores sintácticos