

<b>GUÍA DOCENTE</b>		
<b>EXPERIENCIA PILOTO DE CRÉDITOS EUROPEOS</b>		
<b>UNIVERSIDADES ANDALUZAS</b>		
<b>DATOS BÁSICOS DE LA ASIGNATURA</b>		
<b>NOMBRE: PROCESADORES DE LENGUAJES</b>		
<b>CÓDIGO: 3104004</b>	<b>AÑO DE PLAN DE ESTUDIO: 2005</b>	
<b>TIPO (troncal/obligatoria/optativa) : TRONCAL</b>		
<b>Créditos totales (LRU / ECTS): 9 / 7,2</b>	<b>Créditos LRU/ECTS teóricos: 6 / 4'8</b>	<b>Créditos LRU/ECTS prácticos: 3 / 2'4</b>
<b>CURSO: 1º</b>	<b>CUATRIMESTRE: 2º</b>	<b>CICLO: 2º</b>
<b>CURSO ACADÉMICO: 2013 – 2014</b>		
<b>TITULACIÓN: INGENIERÍA INFORMÁTICA</b>		
<b>DATOS BÁSICOS DE LOS PROFESORES</b>		
<b>NOMBRE:</b>		
<ul style="list-style-type: none"> <li>• <b>NICOLÁS LUIS FERNÁNDEZ GARCÍA</b> <ul style="list-style-type: none"> <li>○ Despacho: edificio Albert Einstein, tercera planta (Campus de Rabanales)</li> <li>○ Correo electrónico: ma1fegan@uco.es</li> <li>○ Teléfono: 957 21 83 47</li> </ul> </li> </ul>		
<b>CENTRO/DEPARTAMENTO:</b>		
<ul style="list-style-type: none"> <li>• <b>ESCUELA POLITÉCNICA SUPERIOR</b></li> <li>• <b>DEPARTAMENTO DE INFORMÁTICA Y ANÁLISIS NUMÉRICO</b></li> </ul>		
<b>ÁREA: CIENCIA DE LA COMPUTACIÓN E INTELIGENCIA ARTIFICIAL</b>		
<b>URL WEB: <a href="http://www.uco.es/users/ma1fegan/">http://www.uco.es/users/ma1fegan/</a></b>		
<b>DATOS ESPECÍFICOS DE LA ASIGNATURA</b>		
<b>1. DESCRIPTORES</b>		
<ul style="list-style-type: none"> <li>• <b>Compiladores</b></li> <li>• <b>Traductores e intérpretes</b></li> <li>• <b>Fases de compilación</b></li> <li>• <b>Optimización de código</b></li> <li>• <b>Macroprocesadores</b></li> </ul>		

## **2. SITUACIÓN**

### **2.1. PRERREQUISITOS**

- Se requieren conocimientos específicos de la materia de Teoría de Autómatas y Lenguajes Formales:
  - Expresiones regulares
  - Autómatas finitos
  - Gramáticas de contexto libre
  - Etc.

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

Esta materia está estrechamente relacionada con bastantes materias de la titulación:

- Metodología de la Programación
- Estructuras de Datos
- Teoría de Autómatas y Lenguajes Formales
- Etc.

### **2.3. RECOMENDACIONES**

- Responsabilidad en el estudio continuado de la asignatura: teoría, prácticas, ejercicios, trabajos en grupo, hojas de ejercicios, etc.
- Predisposición para plantear las dudas que se tengan sobre la asignatura en las tutorías individuales o por correo electrónico.

### 3. COMPETENCIAS

#### 3.1. COMPETENCIAS TRANSVERSALES/GENÉRICAS

- **Instrumentales**
  - Capacidad de análisis y síntesis
  - Capacidad de organización y planificación
  - Comunicación oral y escrita en la lengua nativa
  - Conocimiento de una lengua extranjera (inglés)
  - Conocimientos informáticos relativos al ámbito de estudio
  - Resolución de problemas
  - Toma de decisiones.
  
- **Personales**
  - Capacidad para aplicar los conocimientos teóricos a los casos prácticos.
  - Aprendizaje autónomo
  - Trabajo en grupo.
  
- **Sistémicas**
  - Razonamiento crítico
  - Creatividad
  - Motivación por la calidad.

#### 3.2. COMPETENCIAS ESPECÍFICAS

- **Cognitivas (Saber):**
  - Desarrollo de la capacidad de abstracción para comprender cuáles son los componentes básicos utilizados en el diseño de lenguajes
  - Conocimiento de los fundamentos teóricos del proceso de traducción de lenguajes
  - Utilización de los recursos disponibles para la generación automática de compiladores e intérpretes.
  - Integración de sistemas
  - Programación
  - Matemáticas
  
- **Procedimentales/Instrumentales (Saber hacer):**
  - Planificación, organización y estrategia.
  - Redacción e interpretación de documentación técnica
  
- **Actitudinales (Ser):**
  - Interés por la calidad
  - Toma de decisiones
  - Capacidad de iniciativa y participación

#### 4. OBJETIVOS

- Comprender los conceptos fundamentales de la traducción y la interpretación.
- Conocer las fases del proceso de compilación.
- Saber utilizar las herramientas para la generación automática de compiladores e intérpretes

## **5. METODOLOGÍA**

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

- **Número total de horas de trabajo del alumno: 192 horas**
- Esta asignatura está en vías de extinción y, por tanto, no tiene clases presenciales, pero sí tiene tutorías individuales y evaluación.

## 6. TÉCNICAS DOCENTES

Explicaciones de teoría	Trabajos de teoría X	Tutorías individuales X
Explicaciones de prácticas	Elaboración de prácticas X	Realización de ejercicios X

## DESARROLLO Y JUSTIFICACIÓN

- **Trabajos de teoría**
  - El profesor propondrá trabajos de teoría en la página de *moodle* de la asignatura que los alumnos deberán elaborar y entregar en la fecha indicada.
  - Los trabajos podrán hacerse en grupos de un máximo de tres alumnos.
- **Elaboración de las prácticas**
  - Se desarrollará un intérprete para comprender el funcionamiento de las fases del proceso de traducción de un lenguaje: análisis léxico, sintáctico, etc.
  - El intérprete se podrá hacer en grupos compuestos por un máximo de tres personas.
- **Realización de ejercicios**
  - Se publicarán hojas de ejercicios en la plataforma de moodle para que los alumnos puedan comprender y aplicar los conceptos teóricos.
  - Los alumnos procurarán hacer las hojas de ejercicios durante todo el desarrollo de la asignatura, facilitando de esta forma la comprensión de la asignatura.
  - Los alumnos preguntarán al profesor las dudas que tengan sobre los ejercicios durante las horas de tutorías.
- **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, el trabajo de teoría, el trabajo de prácticas, las hojas de ejercicios o el examen final.
  - Aunque las dudas más simples puedan plantearse mediante correo electrónico, es preferible que haya una reunión personal del profesor y el alumno para resolver las dudas más complejas.

## 7. BLOQUES TEMÁTICOS

### PROGRAMA DE TEORÍA

- **TEORÍA**
  1. INTRODUCCIÓN
  2. ANÁLISIS LÉXICO
  3. FUNDAMENTOS TEÓRICOS DEL ANÁLISIS SINTÁCTICO

4. ANÁLISIS SINTÁCTICO DESCENDENTE
5. ANÁLISIS SINTÁCTICO ASCENDENTE
6. TRADUCCIÓN BASADA EN LA SINTAXIS

- **PRÁCTICAS**

1. INTRODUCCIÓN A JAVA
2. ANTLR

## **8. BIBLIOGRAFÍA**

### **8.1 GENERAL**

- **BIBLIOGRAFÍA BÁSICA DE LA PARTE DE TEORÍA**

- Aho, A. V.; Sethi, R. y Ullman, J. D. “Compiladores: Principios, Técnicas y Herramientas”. Addison-Wesley Iberoamérica, 1990. ISBN: 0-201-62903-8.
- Aho, A. V. et al. “Compilers: principles, techniques and tools”, Addison-Wesley, 2007. ISBN: 978-0-321-49169-5.
- Appel, A. W. “Modern Compiler Implementation in Java (2nd edition)”. Cambridge University Press, 2002. ISBN: 0-521-82060-X.
- Garrido Alenda, A. et alii. “Diseño de compiladores”. Servicio de publicaciones de la Universidad de Alicante, 2002. ISBN: 84-7908-700-5.

- **BIBLIOGRAFÍA BÁSICA DE LA PARTE DE PRÁCTICAS**

- Arnold, K. et al. El lenguaje de programación Java, Addison-Wesley, 2001. ISBN 8478290451.
- Bergsten, H. “Java Server Pages”. 2nd edition O’Reilly
- Carlson, D. “Eclipse distilled”. Addison-Wesley, 2005. ISBN 0321288157.
- Deitel, H. M. y Deitel, P. J. “Java: how to program”, 2nd edition, Prentice Hall, 1998, ISBN 0-13-899394-7.
- Eckel, B. “Piensa en Java”, Prentice Hall, 2002. ISBN: 84-205-3192-8
- Gallardo, D. et al. “Eclipse in action. A Guide for Java Developers”. Manning Publications Co., 2003. ISBN: 1-9300110-96-0.
- García Cota, E. J. y Troyano Jiménez, J.E. “Guía práctica de ANTLR 2.7.2.”. Disponible en
  - <http://www.lsi.us.es/~troyano/documentos/guia.pdf>
  - <http://www.uco.es/users/malfegan/manuales/pl/ANTLR/ANTLR-castellano.pdf>
- Parr, Terence. ANTLR, Reference Manual. Disponible en:
  - <http://wwwantlr.org/doc/index.html>
  - <http://www.uco.es/users/malfegan/manuales/pl/ANTLR/ANTLR-INGLES.pdf>

## 8.2 ESPECÍFICA

### • PROCESADORES DE LENGUAJES

- Alfonseca Moreno, M.; De La Cruz Echeandía, M; Ortega De La Puente, A. y Pulido Cañabate, E. “Compiladores e Intérpretes. Teoría y Práctica”. Pearson - Prentice Hall, 2006. ISBN: 84-205-5031-0
- Iñesta, J. M., García, P. y Gracia, I. “Técnicas básicas para el diseño de compiladores”. Servicio de Publicaciones de la Universitat Jaume I, 1998. ISBN: 84-8021-259-4.
- Jiménez Millán, J. A. “Compiladores y procesadores de lenguajes”. Servicio de Publicaciones de la Universidad de Cádiz. ISBN: 849627439X
- Fischer, C. N. y LeBlanc Jr., R. J. “Crafting a Compiler”. The Benjamin / Cummings Publishings Company, Inc., 1.988. ISBN: 0-8053-3201-4.
- Louden, K. C. “Construcción de compiladores. Principios y práctica”. Thomson editores, 2004. ISBN: 970-686-299-4.
- Pyster, A. B. Compilers Design and Construction. Tools and Techniques. With C and Pascal. 2nd Edittion. Van Nostrand Reinhold Company, Inc., 1988. ISBN: 0-442-27536-6.
- Teufel, B.; Schmidt, S y Teufel, T. “Compiladores. Conceptos fundamentales”. Addison-Wesley Iberoamericana, 1995. ISBN: 0-201-65365-6.
- Vivancos, E., Moreno, L., Gisbert, V y Benedí, J.M. “Compiladores I: una introducción a la fase de análisis”. Servicio de Publicaciones de la Universidad Politécnica de Valencia, 2000. ISBN: 84-7721-915-X.
- Waite, W. M. y Goos, G. “Compiler construction”. Editorial Springer - Verlag, 1985. ISBN: 0-387-90821-8

### • FUNDAMENTOS DE TEORÍA DE AUTÓMATAS Y LENGUAJES FORMALES

- Hopcroft, J. E.; Motwani R. y Ullman, J. D. “Introducción a la teoría de autómatas, lenguajes y computación”. Segunda edición. Addison-Wesley, 2.002. ISBN: 0-201-44124-1.
- Martin, J. C. “Lenguajes formales y teoría de la computación”. Tercera Edición. McGraw-Hill, 2.004. ISBN: 970-10-4594-7.

### • ENLACES A PÁGINAS WEB

- ANTLR: <http://www.antlr.org>
- Eclipse: <http://www.eclipse.org/>
- Instalación de ANTLR en Eclipse: <http://antlreclipse.sourceforge.net>
- Java: <http://www.java.com/es/>

## 9. TÉCNICAS DE EVALUACIÓN

- Se tendrán en cuenta los siguientes aspectos:
  - El conocimiento y dominio de los conceptos teóricos.
  - La elaboración del trabajo de teoría.
  - La capacidad para desarrollar aplicaciones prácticas específicas de la asignatura: diseño de intérpretes o compiladores.

## CRITERIOS DE EVALUACIÓN Y CALIFICACIÓN

- La evaluación de la asignatura será realizada teniendo en cuenta los siguientes apartados:
  - **Teoría**
    - Examen escrito de teoría:
      - Se realizará un examen escrito al final del curso que estará compuesto por preguntas teóricas y ejercicios.
    - Trabajo de teoría:
      - Se desarrollará un trabajo sobre conceptos relacionados con la asignatura.
      - Los trabajos se podrán hacer en grupos compuestos por un máximo de tres personas
  - **Prácticas**
    - Se desarrollará un intérprete de un lenguaje de programación específico
    - El intérprete se podrá hacer en grupos compuestos por un máximo de tres personas
- La calificación final se obtendrá como resultado de la siguiente expresión:

$$\text{Calificación final} = ( 3 * \text{examen de teoría} \\ + 1 * \text{trabajo de teoría} \\ + 2 * \text{trabajo de prácticas} \\ ) / 6$$

- Importante: para poder aprobar la asignatura es imprescindible:
  - Realizar correctamente los trabajos de prácticas.
  - Realizar correctamente el trabajo de teoría.
  - Obtener una nota en el examen de teoría igual o superior a cuatro (4).
  - Obtener una calificación final igual o superior a cinco (5) puntos.



## 10. TEMARIO DESARROLLADO

### TEORÍA

#### 1. INTRODUCCIÓN

- Traducción e interpretación
- Tipos de traductores
- Programas relacionados con la traducción
- Estructura de un compilador: fases y pasos
- Herramientas para la construcción de compiladores
- Combinación de compiladores: “*Bootstrapping*”

#### 2. ANÁLISIS LÉXICO

- Introducción
- Especificación de los componentes léxicos: expresiones regulares
- Reconocimiento de los componentes léxicos: autómatas finitos
- Implementación de los analizadores léxicos
- Detección y recuperación de los errores léxicos

#### 3. FUNDAMENTOS TEÓRICOS DEL ANÁLISIS SINTÁCTICO

- Introducción
- Especificación sintáctica de los lenguajes: gramáticas de contexto libre
  - Definición y convenios de notación
  - Derivación y árbol sintáctico
  - Lenguaje generado por una gramática de contexto libre
  - Diseño de gramáticas de contexto libre
  - Ambigüedad
  - Operaciones de limpieza de las gramáticas de contexto libre
  - Recursividad por la izquierda
  - Factorización por la izquierda
  - Formas normales de Chomsky y Greibach
- Tipos de análisis sintácticos
- Detección y recuperación de los errores sintácticos
- Generadores de analizadores sintácticos

#### 4. ANÁLISIS SINTÁCTICO DESCENDENTE

- Introducción
- Análisis sintáctico mediante descenso recursivo
- Análisis sintáctico descendente predictivo
  - Caracterización de las gramáticas LL
  - Funciones Primero y Siguiente
  - Construcción de la Tabla Predictiva
  - Método recursivo sin retroceso
  - Método no recursivo
- Detección y recuperación de errores durante el análisis sintáctico descendente

## **5. ANÁLISIS SINTÁCTICO ASCENDENTE**

- Introducción
- Análisis SLR
- Análisis LR canónico
- Análisis LALR
- Detección y recuperación de errores durante el análisis sintáctico ascendente

## **6. TRADUCCIÓN BASADA EN LA SINTAXIS**

- Introducción
  - Definiciones basadas en la sintaxis
    - Descripción
    - Definiciones S-atribuidas
    - Definiciones L-atribuidas
    - Evaluación de los árboles anotados
  - Esquemas de traducción
    - Descripción
    - Equivalencia con las definiciones L-atribuidas
  - Evaluación descendente y predictiva de atributos heredados y sintetizados
  - Evaluación ascendente de atributos sintetizados
  - Evaluación ascendente de atributos heredados y sintetizados
- 

## **PRÁCTICAS**

- Información general
- Instalación de herramientas
- Java
  - Introducción
  - Características básicas
  - Clases y objetos
  - Herencia
  - Polimorfismo
  - Excepciones
- ANTLR
  - Introducción
  - Análisis Léxico
  - Análisis Sintáctico
  - Análisis Semántico