

## DATOS BÁSICOS DE LA GUÍA DOCENTE:

<b>Materia:</b>	ANÁLISIS LÉXICO Y SINTÁCTICO		
<b>Identificador:</b>	30058		
<b>Titulación:</b>	GRADUADO EN INGENIERÍA INFORMÁTICA. PLAN 2008 (BOE 15/12/2008)		
<b>Módulo:</b>	TEORIA DE LA COMPUTACION		
<b>Tipo:</b>	OBLIGATORIA		
<b>Curso:</b>	2	<b>Periodo lectivo:</b>	Segundo Cuatrimestre
<b>Créditos:</b>	3	<b>Horas totales:</b>	75
<b>Actividades Presenciales:</b>	35	<b>Trabajo Autónomo:</b>	40
<b>Idioma Principal:</b>	Castellano	<b>Idioma Secundario:</b>	Inglés
<b>Profesor:</b>		<b>Correo electrónico:</b>	

## PRESENTACIÓN:

El objetivo del curso es el aprendizaje de soluciones a problemas encontrados en el diseño de intérpretes y compiladores de lenguajes. Desde una introducción general al estudio de compiladores e intérpretes, un estudio más detallado se enfoca al análisis del proceso de un sistema de compilación: las fases de escaneo y "parseado" del programa fuente pasando por la codificación de una representación intermedia. Las prácticas se orientan con énfasis especial en la obtención de ejemplos de analizadores léxicos y sintácticos a través de distintas técnicas y contextos. Durante las prácticas el estudiante hace uso de las competencias en programación, usando lenguaje Java y C.

## COMPETENCIAS PROFESIONALES A DESARROLLAR EN LA MATERIA:

<b>Competencias Generales de la titulación</b>	G02	Capacidad innovadora para proponer y encontrar formas nuevas y eficaces de realizar cualquier tarea y/ o función dentro de su entorno profesional con una elevada motivación por la calidad
	G03	Capacidad para trabajar dentro de equipos multidisciplinares para conseguir metas comunes, anteponiendo los intereses colectivos a los personales
	G04	Capacidad para trabajar siempre con responsabilidad y compromiso, creando un alto sentido del deber y el cumplimiento de las obligaciones
	G05	Capacidad para adaptarse a diferentes entornos con una actitud positiva y optimista y orientar su conducta a la consecución de metas
	G07	Capacidad para trabajar con flexibilidad y versatilidad para adaptarse a las necesidades y exigencias de su entorno profesional
	G13	Capacidad para utilizar estrategias de aprendizaje individuales orientadas a la mejora continua en el ejercicio profesional y para emprender estudios posteriores de forma autónoma
	G14	Capacidad de abstracción para manejar diferentes modelos complejos de conocimiento y aplicarlos al planteamiento y resolución de problemas
	G15	Capacidad para estructurar la realidad relacionando objetos, situaciones y conceptos a través del razonamiento lógico matemático
<b>Competencias Específicas de la titulación</b>	E03	Capacidad para reconocer los principios teóricos y aplicar satisfactoriamente los métodos prácticos apropiados para el análisis y la resolución de problemas de ingeniería
	E09	Capacidad para mantener las competencias profesionales mediante el aprendizaje autónomo y la mejora continua
	E11	Capacidad para mantenerse al día en el mundo tecnológico y empresarial en el ámbito de las tecnologías de la informática y comunicaciones
	E12	Capacidad para gestionar la complejidad a través de la abstracción, el modelado, las "best practices", los patrones, los estándares y el uso de herramientas apropiadas
	E27	Capacidad para elaborar y mantener documentación descriptiva de la génesis, producción y operatividad de los sistemas informáticos
<b>Resultados de Aprendizaje</b>	R01	Utilizar correctamente términos técnicos relacionados con la teoría y los procesos de compilación
	R02	Conocer y usar las herramientas más utilizadas para la construcción automática de compiladores
	R03	Entender, crear y modificar analizadores léxicos y semánticos con la herramienta
	R04	Conocer los principales ficheros de configuración de JavaCC
	R05	Definir gramáticas que permitan tratar un lenguaje dado y seguir los procedimientos necesarios para implementar un compilador que reconozca dicha gramática

R06	Reconocer y diferenciar los aspectos claves de los distintos tipos de analizadores léxicos y sintácticos analizados en la materia
-----	---

### REQUISITOS PREVIOS:

Es recomendable que los alumnos hayan cursado la materia de Lenguajes Formales.

### PROGRAMACIÓN DE LA MATERIA:

#### Contenidos de la materia:

<b>1 - Introducción.</b>
1.1 - Conceptos básicos: Compiladores, intérpretes, analizadores y traductores.
1.2 - Fases del compilador.
<b>2 - Análisis léxico.</b>
2.1 - Análisis léxico. Introducción a scanners. Expresiones regulares.
2.2 - Análisis léxico. Autómatas. Implementación.
<b>3 - Análisis sintáctico.</b>
3.1 - Introducción al Análisis Sintáctico.
3.2 - Gramáticas libres de contexto: Derivaciones y Árboles de Parseo
3.3 - Parsers LL y LR: Introducción.
3.4 - Parsers LL: Condición, tablas de parseo y algoritmos (LL1).
3.5 - Parsers LR: Análisis ascendente.
<b>4 - Java CC y proyectos.</b>
4.1 - Nociones básicas de JavaCC: lexer.
4.2 - Especificación de gramáticas en JavaCC: reglas sintácticas (parser).

La planificación de la asignatura podrá verse modificada por motivos imprevistos (rendimiento del grupo, disponibilidad de recursos, modificaciones en el calendario académico, etc.) y por tanto no deberá considerarse como definitiva y cerrada.

### METODOLOGÍAS Y ACTIVIDADES DE ENSEÑANZA Y APRENDIZAJE:

#### Metodologías de enseñanza-aprendizaje a desarrollar:

- **Sesiones teóricas.** El profesor expondrá la parte teórica de la asignatura. Las sesiones teóricas serán acompañadas por ejercicios para aplicar los conceptos teóricos. Se valorará la participación, debate, preguntas e inquietudes mostradas por los alumnos relacionadas con la asignatura.

- **Sesiones prácticas.** Aprendizaje basado en resolución de ejercicios y problemas. Los contenidos teóricos se apoyan en la resolución de problemas que serán propuestos por el profesor y resueltos por los alumnos. Algunos de estos problemas serán solucionados por los alumnos como parte de su trabajo autónomo en horario fuera de las clases lectivas. Otros problemas se resolverán en el tiempo dedicado a clases prácticas. La participación de los alumnos en este proceso formará parte de la evaluación, esto incluye la resolución de problemas de manera voluntaria y la participación cuando otro compañero está desarrollando esta tarea.

- **Sesiones de tutoría.** Las sesiones de tutoría se anunciarán a través de la PDU. Se fomentará en estas sesiones el uso de tecnologías no presenciales.

- **Trabajo autónomo.** Los alumnos deberán estudiar el material presentado, tratar de resolver los ejercicios y problemas propuestos, que serán similares a los que formarán parte de los exámenes parciales y desarrollar los trabajos propuestos.

#### Volumen de trabajo del alumno:

Modalidad organizativa	Métodos de enseñanza	Horas estimadas
Actividades Presenciales	Clase magistral	10

	Otras actividades teóricas	4
	Casos prácticos	6
	Resolución de prácticas, problemas, ejercicios etc.	8
	Talleres	2
	Prácticas de laboratorio	1
	Actividades de evaluación	4
<b>Trabajo Autónomo</b>	Asistencia a tutorías	2
	Estudio individual	18
	Preparación de trabajos individuales	12
	Preparación de trabajos en equipo	4
	Tareas de investigación y búsqueda de información	4
	<b>Horas totales:</b>	75

### SISTEMA DE EVALUACIÓN:

#### Obtención de la nota final:

Pruebas escritas:	50	%
Trabajos individuales:	35	%
Trabajos en equipo:	10	%
Participación:	5	%
<b>TOTAL</b>	<b>100</b>	<b>%</b>

\*Las observaciones específicas sobre el sistema de evaluación serán comunicadas por escrito a los alumnos al inicio de la materia.

### BIBLIOGRAFÍA Y DOCUMENTACIÓN:

#### Bibliografía básica:

AHO, A.V., LAM M., SETHI, R., ULLMAN, J.D. (2008). Compiladores: Principios, técnicas y herramientas. México: Prentice Hall. (ISBN 9789702611332)

CAMPBELL, B., IYER, S., AKBAL-DELIBAS, B. (2012). Introduction to Compiler Construction in a Java World. Chapman & Hall/ CRC (ISBN 9781439860885)

HOLUB, A. I. (1991). Compiler design in C. Prentice-Hall. (ISBN 9780131550452)

SÁNCHEZ, G., VALVERDE J.A. (1989). Compiladores e Intérpretes. Madrid: Ed. Díaz de Santos. (ISBN 9788487189067)

#### Bibliografía recomendada:

AHO, A.V., ULLMAN, J.D. (1972). The theory of parsing, translation and compiling (Vol. I y II). Prentice-Hall. (ISBN 9780139145568)

ALFONSECA, M., SANCHO, J., MARTÍNEZ ORGA, M. (1997). Teoría de lenguajes, gramáticas y autómatas. Madrid: Universidad y Cultura, D.L. (ISBN 8460560929)

GRUNE, D., VAN REEUWIJK, K., BAL, H.E., JACOBS, C. J. H., LANGEDOEN, K. (2012). Modern Compiler Design. Springer. (ISBN 9781461446989)

HOPCROFT, J. E., MOTWANI, R., ULLMAN, J. D. (2002). Introducción a la teoría de autómatas, lenguajes y computación, 2ª edición. Madrid: Pearson Educación. (ISBN 9788478290567)

#### Páginas web recomendadas:

Java CC	<a href="https://java.net/projects/javacc">https://java.net/projects/javacc</a>
---------	---

\* Guía Docente sujeta a modificaciones