



Guía docente				
Datos Identificativos				2014/15
Asignatura (*)	Procesamiento de Lenguajes		Código	614G01067
Titulación	Grao en Enxeñaría Informática			
Descriptorios				
Ciclo	Periodo	Curso	Tipo	Créditos
Grado	1º cuatrimestre	Cuarto	Obligatoria	6
Idioma	CastellanoGallego			
Prerrequisitos				
Departamento	Tecnoloxías da Información e as Comunicaciós			
Coordinador/a	Arcay Varela, Bernardino		Correo electrónico	bernardino.arcay@udc.es
Profesorado	Arcay Varela, Bernardino		Correo electrónico	bernardino.arcay@udc.es
	Dafonte Vazquez, Jose Carlos			carlos.dafonte@udc.es
	Gomez Garcia, Angel			angel.gomez@udc.es
	Martinez Perez, Maria			maria.martinez@udc.es
Web				
Descrición xeral	Compiladores; tradutores e intérpretes; etapas dun compilador; optimización de código; macroprocesadores. O obxectivo é familiarizar ó alumno co funcionamento dos reconecedores da linguaxe e os compiladores como un caso particular, o entorno no que traballan así coma algunhas ferramentas software para a construción dos mesmos. É preciso asumir a característica interdisciplinar da asignatura. Adquirir os coñecementos necesarios para deseñar e implementar as diferentes etapas necesarias para o desenvolvemento dun reconecedor da linguaxe: análise (léxico, sintáctico e semántico) e síntese (xeración de código intermedio, optimización de código e xeración de código obxeto).			

Competencias de la titulación	
Código	Competencias de la titulación
A28	Capacidad de identificar y analizar problemas, y diseñar, desarrollar, implementar, verificar y documentar soluciones software sobre la base de un conocimiento adecuado de las teorías, modelos y técnicas actuales.
A32	Capacidad de desarrollar procesadores específicos y sistemas empotrados, así como desarrollar y optimizar el software de dichos sistemas.
A34	Capacidad de diseñar e implementar software de sistemas y de comunicaciones.
A39	Capacidad para tener un conocimiento profundo de los principios fundamentales y modelos de la computación y saberlos aplicar para interpretar, seleccionar, valorar, modelar, y crear nuevos conceptos, teorías, usos y desarrollos tecnológicos relacionados con la informática.
A40	Capacidad para conocer los fundamentos teóricos de los lenguajes de programación y las técnicas de procesamiento léxico, sintáctico y semántico asociadas, y saber aplicarlas para la creación, diseño y procesamiento de lenguajes.
B1	Capacidad de resolución de problemas
B3	Capacidad de análisis y síntesis
C2	Dominar la expresión y la comprensión de forma oral y escrita de un idioma extranjero.
C4	Desarrollarse para el ejercicio de una ciudadanía abierta, culta, crítica, comprometida, democrática y solidaria, capaz de analizar la realidad, diagnosticar problemas, formular e implantar soluciones basadas en el conocimiento y orientadas al bien común.
C6	Valorar críticamente el conocimiento, la tecnología y la información disponible para resolver los problemas con los que deben enfrentarse.

Resultados de aprendizaje			
Competencias de materia (Resultados de aprendizaje)			Competencias de la titulación
Conocer los conceptos teóricos básicos en los que se basan los compiladores. Diseñar e implementar cada una de las fases precisas para la implementación de un compilador. Conocer las distintas herramientas disponibles para la implementación de compiladores y manejar algunas de las más habituales. El estudiante también sabrá verificar el funcionamiento del procesador del lenguaje.	A28		
	A40		



Partiendo de la base teórica de la computación, se plantean diferentes técnicas y herramientas (con una orientación muy práctica) para el procesamiento del lenguaje.	A39		
Saber optimizar el código objeto en función de la plataforma que lo va a ejecutar. Saber elegir el tipo de optimización según la máquina objeto y su arquitectura.	A32 A34		
Saber analizar y diagnosticar el caso concreto, el objetivo, para el que se precisa un reconocedor del lenguaje. El estudiante también será capaz de documentarse, sintetizar y adaptar la solución con un enfoque totalmente práctico pero basado en conocimiento.		B1 B3	C2 C4 C6

Contenidos	
Tema	Subtema
Tema I. Introducción	1.1 Estructura de un compilador. 1.2 Ejemplo de las fases de un compilador
Tema II. Lenguajes y gramáticas	2.1 Notación y clasificación de Chomsky. 2.2 Gramáticas de contexto libre (GCL) y notación BNF. 2.3 Mejora y simplificación de gramáticas.
Tema III. Análisis Léxico	3.1 Tipos de máquinas reconocedoras o autómatas. 3.2 Autómatas Finitos. 3.3 Conversión entre gramática regular, expresión regular y autómata finito. 3.4 Implementación de autómatas.
Tema IV. Análisis Sintáctico (Parsing)	4.1 Máquinas teóricas, mecanismos con retroceso 4.2 Algoritmos sin retroceso. 4.2.1 Análisis sintáctico ascendente por precedencia simple. 4.2.2 Analizadores descendentes LL(K). 4.2.3 Analizadores ascendentes LR(k). 4.2.4 Generador de analizadores sintácticos YACC.
Tema V. Análisis Semántico	5.1 Definiciones dirigidas por la sintaxis. 5.2 Esquema de traducción. 5.3 Comprobaciones en tiempo de compilación.
Tema VI. Generación de Código	6.1 Lenguajes intermedios. 6.2 Generación de código intermedio. 6.3 Generación de código desde lenguaje intermedio.
Tema VII. Optimización de Código	7.1 Algoritmo de Nakata. 7.2 Lazos en los grafos de flujo. 7.3 Análisis global del flujo de datos. 7.4 Solución iterativa de las ecuaciones de flujo de datos.
Tema VIII. Errores	8.1 Tipos de errores 8.2 Recuperación de errores léxico-gráficos.
Tema IX. Intérpretes y Complementos	9.1 Estructura de un intérprete actual. 9.2 Arquitectura neutral de Java.

Planificación			
Metodologías / pruebas	Horas presenciales	Horas no presenciales / trabajo autónomo	Horas totales
Sesión magistral	21	31.5	52.5
Prueba objetiva	3	18	21
Trabajos tutelados	7	17.5	24.5
Prácticas de laboratorio	14	14	28
Atención personalizada	24	0	24



(*)Los datos que aparecen en la tabla de planificación són de carácter orientativo, considerando la heterogeneidad de los alumnos

Metodologías	
Metodologías	Descripción
Sesión magistral	En estas sesiones se analizarán en profundidad los conceptos teóricos y principales técnicas para la implementación de un reconocedor del lenguaje.
Prueba objetiva	Se realizará un examen al final del cuatrimestre y su correspondiente 2ª oportunidad. Este examen se centrará principalmente en los aspectos teóricos y ejercicios asociados. Sin embargo, podrá preguntarse alguna cuestión relacionada con las prácticas de implementación realizadas.
Trabajos tutelados	Se llevará a cabo un proyecto a propuesta de un grupo de 2 estudiantes del que se realizará un seguimiento en el horario de trabajo en grupo.
Prácticas de laboratorio	Se llevarán a cabo pequeñas prácticas sobre los diferentes temas de la asignatura que se irán desarrollando en las sesiones magistrales. Algunas de ellas serán ejercicios de desarrollo y otras de implementación.

Atención personalizada	
Metodologías	Descripción
Sesión magistral Prácticas de laboratorio Trabajos tutelados	Especialmente en el caso del proyecto a desarrollar, organizado por grupos de alumnos, se realizará un seguimiento. A ello se dedicarán principalmente las horas de trabajo en grupo reducido.

Evaluación		
Metodologías	Descripción	Calificación
Prácticas de laboratorio	Se llevarán a cabo pequeñas prácticas sobre los diferentes temas de la asignatura que se irán desarrollando en las sesiones magistrales. Algunas de ellas serán ejercicios de desarrollo y otras de implementación. El porcentaje por asistencia y entrega de estas pequeñas prácticas es testimonial (10%). El 20% restante corresponde al desarrollo del proyecto.	30
Prueba objetiva	Se realizará un examen al final del cuatrimestre y su correspondiente 2ª oportunidad. Este examen se centrará principalmente en los aspectos teóricos y ejercicios asociados. Sin embargo, podrá preguntarse alguna cuestión relacionada con las prácticas de implementación realizadas.	50
Trabajos tutelados	Se llevará a cabo un proyecto a propuesta de un grupo de 2 estudiantes del que se realizará un seguimiento en el horario de trabajo en grupo.	20
Otros		

Observaciones evaluación
En cualquier caso, es preciso aprobar las dos partes (prueba objetiva y prácticas-trabajos). En caso contrario, la máxima nota que se podrá conseguir es un 4.5. IMPORTANTE: El 40% de la nota lo aporta el proyecto que propondrán los grupos, el 20% corresponde al seguimiento y el 20% a la entrega final. ESTUDIANTES CON MATRÍCULA A TIEMPO PARCIAL: Deberán ponerse en contacto con los profesores de la asignatura para posibilitar la realización de las tareas fuera de la organización habitual de la materia.

Fuentes de información	
Básica	
Complementaria	

Recomendaciones
Asignaturas que se recomienda haber cursado previamente
Diseño de los Lenguajes de Programación/614G01065
Asignaturas que se recomienda cursar simultáneamente



Asignaturas que continúan el temario
Teoría de la computación/614G01039
Otros comentarios
Aunque tiene relación con otras asignaturas, está fuertemente relacionada con "Diseño de Lenguajes de Programación", que la complementa, y "Teoría de la Computación", que es la base inicial para poder afrontar los contenidos.

(*) La Guía Docente es el documento donde se visualiza la propuesta académica de la UDC. Este documento es público y no se puede modificar, salvo cosas excepcionales bajo la revisión del órgano competente de acuerdo a la normativa vigente que establece el proceso de elaboración de guías