



Guía Docente				
Datos Identificativos			2013/14	
Asignatura (*)	Procesamento de Linguaxes	Código	614G01067	
Titulación				
Descritores				
Ciclo	Período	Curso	Tipo	Créditos
Grao	1º cuatrimestre	Cuarto	Obrigatoria	6
Idioma	CastelánGalego			
Prerrequisitos				
Departamento	Tecnoloxías da Información e as Comunicaciós			
Coordinación	Arcay Varela, Bernardino	Correo electrónico	bernardino.arcay@udc.es	
Profesorado	Arcay Varela, Bernardino Dafonte Vazquez, Jose Carlos Gomez Garcia, Angel Martinez Perez, Maria Rodríguez Fernández, Alejandra	Correo electrónico	bernardino.arcay@udc.es carlos.dafonte@udc.es angel.gomez@udc.es maria.martinez@udc.es alejandra.rodriquez@udc.es	
Web				
Descrición xeral	<p>Compiladores; tradutores e intérpretes; etapas dun compilador; optimización de código; macroprocesadores.</p> <p>O obxectivo é familiarizar ó alumno co funcionamento dos recoñecedores 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 recoñecedor da linguaxe: análise (léxico, sintáctico e semántico) e síntese (xeración de código intermedio, optimización de código e xeración de código obxeto).</p>			

Competencias da titulación	
Código	Competencias da titulación

Resultados da aprendizaxe			
Competencias de materia (Resultados de aprendizaxe)	Competencias da titulación		
Coñecer os conceptos teóricos básicos nos que se basean os compiladores. Deseñar e implementar cada unha das fases precisas para a implementación dun compilador. Coñecer as distintas ferramentas dispoñibles para a implementación de compiladores e manexar algunhas das máis habituais. O estudante tamén saberá verificar o funcionamento do procesador da linguaxe.	A28		
	A40		
Partindo da base teórica da computación, plantéxanse diferentes técnicas e ferramentas (cunha orientación moi práctica) para o procesamento da linguaxe.	A39		
Saber optimizar o código obxecto en función da plataforma que o vai a executar. Saber elixir o tipo de optimización segundo a máquina obxecto e a súa arquitectura.	A32		
	A34		
Saber analizar e diagnosticar o caso concreto, o obxectivo, para o que se precisa un recoñecedor da linguaxe. O estudante tamén será capaz de documentarse, sintetizar e adaptar a solución cun enfoque totalmente práctico pero baseado no coñecemento.		B1	C2
		B3	C4
			C6

Contidos	
Temas	Subtemas
Tema I. Introducción	1.1 Estructura dun compilador. 1.2 Exemplo das fases dun compilador.
Tema II. Linguaxes e Gramáticas	2.1 Notación e clasificación de Chomsky. 2.2 Gramáticas de contexto libre (GCL) e notación BNF. 2.3 Mellora e simplificación de gramáticas.



Tema III. Análise Léxica	<p>3.1 Tipos de máquinas recoñecedoras ou autómatas.</p> <p>3.2 Autómatas Finitos.</p> <p>3.3 Conversión entre gramática regular, expresión regular e autómatata finito.</p> <p>3.4 Implementación de autómatas.</p>
Tema IV. Análise Sintáctica (Parsing)	<p>4.1 Máquinas teóricas, mecanismos con retroceso</p> <p>4.2 Algoritmos sen retroceso.</p> <p>4.2.1 Análise sintáctica ascendente por precedencia simple.</p> <p>4.2.2 Analizadores descendentes LL(K).</p> <p>4.2.3 Analizadores ascendentes LR(k).</p> <p>4.2.4 Xenerador de analizadores sintácticos YACC.</p>
Tema V. Análise Semántica	<p>5.1 Definicións dirixidas pola sintáxis.</p> <p>5.2 Esquema de tradución.</p> <p>5.3 Comprobacións en tempo de compilación.</p>
Tema VI. Xeneración de Código	<p>6.1 Linguaxes intermedios.</p> <p>6.2 Xeneración de código intermedio.</p> <p>6.3 Xeneración de código dende linguaxe intermedio.</p>
Tema VII. Optimización de Código	<p>7.1 Algoritmo de Nakata.</p> <p>7.2 Lazos nos grafos de fluxo.</p> <p>7.3 Análise global do fluxo de datos.</p> <p>7.4 Solución iterativa das ecuacións de fluxo de datos.</p>
Tema VIII. Erros	<p>8.1 Tipos de erros.</p> <p>8.2 Recuperación de erros léxico-gráficos.</p>
Tema IX. Intérpretes e Complementos	<p>9.1 Estructura dun intérprete actual.</p> <p>9.2 Arquitectura neutral de Java.</p>

Planificación			
Metodoloxías / probas	Horas presenciais	Horas non presenciais / traballo autónomo	Horas totais
Sesión maxistral	21	31.5	52.5
Proba obxectiva	3	18	21
Traballos tutelados	7	17.5	24.5
Prácticas de laboratorio	14	14	28
Atención personalizada	24	0	24

\*Os datos que aparecen na táboa de planificación son de carácter orientativo, considerando a heteroxeneidade do alumnado

Metodoloxías	
Metodoloxías	Descrición
Sesión maxistral	Nestas sesións analizarase en profundidade os conceptos teóricos e técnicas principais para a implementación dun recoñedor da linguaxe
Proba obxectiva	Realizarase un exame ó final do cuatrimestre e a súa correspondente 2ª oportunidade. Este exame centrarase principalmente nos aspectos teóricos i exercicios asociados. Tamén poderase formular algunha cuestión relacionada coas prácticas de implementación realizadas.
Traballos tutelados	Levarase a cabo un proxecto a proposta de un grupo de 2 estudantes do que se fará un seguimento no horario de traballo en grupo.
Prácticas de laboratorio	Levaranse a cabo pequenas prácticas sobre os diferentes temas da asignatura que se irán explicando nas sesións maxistráis. Algunhas delas serán exercicios de desenvolvemento e outras de implementación.

Atención personalizada	
Metodoloxías	Descrición



Sesión maxistral Prácticas de laboratorio Traballos tutelados	Especialmente no caso do proxecto a desenvolver, organizado por grupos de alumnos, realizarase un seguimento. A iso se adicarán principalmente as horas de traballo en grupo reducido.
---	--

Avaliación		
Metodoloxías	Descrición	Cualificación
Prácticas de laboratorio	Levaranse a cabo pequenas prácticas sobre os diferentes temas da asignatura que se irán explicando nas sesións maxistráis. Algunhas delas serán exercicios de desenvolvemento e outras de implementación. A porcentaxe por asistencia i entrega destas pequenas prácticas é testimonial (10%). O 20% restante corresponde ó desenvolvemento do proxecto.	30
Proba obxectiva	Realizarase un exame ó final do cuatrimestre e a súa correspondente 2ª oportunidade. Este exame centrarase principalmente nos aspectos teóricos i exercicios asociados. Tamén poderase formular algunha cuestión relacionada coas prácticas de implementación realizadas.	50
Traballos tutelados	Levarase a cabo un proxecto a proposta de un grupo de 2 estudantes do que se fará un seguimento no horario de traballo en grupo.	20
Outros		

#### Observacións avaliación

En calquera caso, é preciso aprobar as dúas partes (proba obxectiva e prácticas-traballo). En caso contrario, a nota máxima que se poderá acadar é un 4.5.

**IMPORTANTE:** O 40% da nota apórtao o proxecto a propoñer polos grupos, o 20% corresponde ó seguimento e o 20% á entrega final.

#### Fontes de información

<b>Bibliografía básica</b>	
<b>Bibliografía complementaria</b>	

#### Recomendacións

##### Materias que se recomenda ter cursado previamente

Deseño das Linguaxes de Programación/614G01065

##### Materias que se recomenda cursar simultaneamente

##### Materias que continúan o temario

Teoría da computación/614G01039

#### Observacións

A asignatura troncal de Enxeñería Informática e Enxeñería Técnica en Informática de Sistemas &quot;Teoría de autómatas e linguaxes formais&quot; é de gran utilidade para a comprensión da asignatura de Compiladores.

(\*A Guía docente é o documento onde se visualiza a proposta académica da UDC. Este documento é público e non se pode modificar, salvo casos excepcionais baixo a revisión do órgano competente dacordo coa normativa vixente que establece o proceso de elaboración de guías