



Guía Docente				
Datos Identificativos				2014/15
Asignatura (*)	Deseño das Linguaxes de Prgramación	Código	614G01065	
Titulación				
Descritores				
Ciclo	Período	Curso	Tipo	Créditos
Grao	1º cuatrimestre	Cuarto	Obrigatoria	6
Idioma	Castelán			
Prerrequisitos				
Departamento	Computación			
Coordinación	Alonso Pardo, Miguel angel	Correo electrónico	miguel.alonso@udc.es	
Profesorado	Alonso Pardo, Miguel angel Gómez Rodríguez, Carlos Vilares Ferro, Jesus	Correo electrónico	miguel.alonso@udc.es carlos.gomez@udc.es jesus.vilares@udc.es	
Web	campusvirtual.udc.es			
Descrición xeral	<p>En esta asignatura se tratan aspectos de la especificación y diseño de lenguajes de programación:</p> <ul style="list-style-type: none"> <li>* Criterios de diseño en las principales estructuras de control y de datos de los lenguajes de programación</li> <li>* Diseño de lenguajes de programación orientados a objetos.</li> <li>* Modelos para la definición formal de la semántica de los lenguajes de programación</li> <li>* Especificación formal de los sistemas de tipos. Relaciones de subtipado</li> <li>* Computabilidad. Análisis de complejidad y su relación con el diseño de lenguajes de programación.</li> </ul>			

Competencias da titulación	
Código	Competencias da titulación

Resultados da aprendizaxe			
Competencias de materia (Resultados de aprendizaxe)	Competencias da titulación		
Manejar los principios de diseño de las principales estructuras de control de los lenguajes de programación y sus implicaciones en el desarrollo de programas	A39 A40	B1 B3	C3 C6 C7
Manejar los principios de diseño de las principales estructuras de datos de los lenguajes de programación y sus implicaciones en el desarrollo de programas	A39 A40	B1 B3	C3 C6 C7
Comprender y dominar los principios de diseño de los lenguajes orientados a objetos y de las implicaciones que las elecciones de diseño tienen en el desarrollo de programas	A39 A40	B1 B3	C3 C6 C7 C8
Introducir el lambda-cálculo, tipado y no tipado, como núcleo fundamental de los lenguajes de programación.	A39 A40	B1 B3	C2 C6 C7 C8
Comprender los fundamentos formales de los sistemas de tipado y subtipado	A39 A40	B1 B3	C2 C4 C6 C7 C8



Manejar de forma práctica los conceptos de complejidad algorítmica y sus implicaciones en el diseño de lenguajes de programación	A39	B1	C3
	A40	B3	C4
			C6
			C7
			C8

Contidos	
Temas	Subtemas
Principios de diseño de lenguajes de programación	Nombres, ámbito y ligazón Flujo de control Tipos de datos Subrutinas
Lenguajes orientados a objetos	Fundamentos de diseño de lenguajes orientados a objetos Problemas de tipado de los lenguajes orientados a objetos
Formalización de los sistemas de tipos	Semántica operacional, denotacional y axiomática Introducción al lambda cálculo Lambda cálculo tipado
Computabilidad y complejidad	Computabilidad y lambda cálculo Clases de complejidad

Planificación			
Metodoloxías / probas	Horas presenciais	Horas non presenciais / traballo autónomo	Horas totais
Prácticas de laboratorio	14	42	56
Obradoiro	7	14	21
Proba obxectiva	2	6	8
Sesión maxistral	21	42	63
Atención personalizada	2	0	2

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

Metodoloxías	
Metodoloxías	Descrición
Prácticas de laboratorio	Actividade que permite que os estudantes aprendan de forma efectiva a través da realización de actividades de carácter práctico, neste caso prácticas, demostracións e exercicios.
Obradoiro	Realizaranse como complemento de todas as demais actividades, nalgúns casos de forma autónoma por parte do alumno e noutras ocasións dirixidas polo profesor.
Proba obxectiva	Proba na que se evaluarán os coñecementos adquiridos tanto na parte teórica como na parte práctica da materia.
Sesión maxistral	Exposición oral complementada co uso de medios audiovisuais e a formulación de preguntas dirixidas ós estudantes, co obxectivo de transmitir coñecemento así como de estimular o razoamento crítico do estudante.

Atención personalizada	
Metodoloxías	Descrición



Obradoiro Prácticas de laboratorio	<p>El desarrollo de las clases se realizará atendiendo al progreso de los alumnos en las capacidades de comprensión y asimilación de los contenidos impartidos, compaginando el avance general de la clase con una atención específica a aquellos alumnos que presenten mayores dificultades en la tarea del aprendizaje y con un apoyo adicional a aquellos otros que presenten mayor desenvolvura y deseen ampliar conocimientos.</p> <p>Dado el carácter personalizado de las tutorías, éstas no deben dedicarse a extender los contenidos con nuevos conceptos, sino a aclarar los conceptos ya expuestos. El profesor debe además utilizarlas como una interacción que le permita extraer conclusiones respecto al grado de asimilación de la materia por parte de los alumnos.</p>
---------------------------------------	--

Avaliación		
Metodoloxías	Descrición	Cualificación
Obradoiro	Otras actividades evaluables, cuyo contenido se desarrollarán en las horas de TGR.	20
Proba obxectiva	Ejercicio escrito	40
Prácticas de laboratorio	Trabajos de laboratorio	40

Observacións avaliación
<p>La parte teórico de la asignatura computa un 40% de la nota. Las actividades de los TGR computan un 20% de la nota. La evaluación de los TGR se realizará junto con el examen de teoría.</p> <p>El 40% restante se reparte entre las prácticas y cualquier otra actividad de evaluación que se realice a lo largo del curso.</p> <p>Para aprobar la asignatura es preciso aprobar todos y cada uno de los apartados de la evaluación. Para la Segunda Oportunidad, se conservarán los resultados obtenidos en cada apartado en la Primera Oportunidad.</p> <p>En el caso del alumnado a tiempo parcial, no se penalizará la inasistencia a clases de prácticas y TGR que estén debidamente justificadas.</p>

Fontes de información	
<b>Bibliografía básica</b>	<ul style="list-style-type: none"> <li>- Kim B. Bruce (2002). Foundations of Object-Oriented Languages: Types and Semantics. The MIT Press, Cambridge, MA</li> <li>- Michael L. Scott (2009). Programming Language Pragmatics. Third edition. Morgan Kaufmann Publishers, Burlington, MA</li> <li>- Benjamin C. Pierce (2002). Types and Programming Languages. The MIT Press, Cambridge, MA</li> </ul>
<b>Bibliografía complementaria</b>	<ul style="list-style-type: none"> <li>- Robert W. Sebesta (2010). Concepts of Programming Languages. Pearson</li> <li>- Franklyn A. Turbak and David K. Gifford (2008). Design Concepts in Programming Languages. MIT Press, Cambridge, MA</li> <li>- David A. Watt (2004). Programming Language Design Concepts. John Wiley and sons, Chichester, West Sussex, England</li> </ul>

Recomendacións
<b>Materias que se recomenda ter cursado previamente</b>
<b>Materias que se recomenda cursar simultaneamente</b>
Procesamento de Linguaxes/614G01067
<b>Materias que continúan o temario</b>
Paradigmas de Programación/614G01014
Teoría da computación/614G01039
<b>Observacións</b>



(\*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