



Guía docente				
Datos Identificativos				2014/15
Asignatura (*)	Diseño de los Lenguajes de Programación	Código	614G01065	
Titulación	Grao en Enxeñaría Informática			
Descritores				
Ciclo	Periodo	Curso	Tipo	Créditos
Grado	1º cuatrimestre	Cuarto	Obligatoria	6
Idioma	Castellano			
Prerrequisitos				
Departamento	Computación			
Coordinador/a	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			
Descripción general	<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 de la titulación	
Código	Competencias de la titulación
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.
C3	Utilizar las herramientas básicas de las tecnologías de la información y las comunicaciones (TIC) necesarias para el ejercicio de su profesión y para el aprendizaje a lo largo de su vida.
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.
C7	Asumir como profesional y ciudadano la importancia del aprendizaje a lo largo de la vida.
C8	Valorar la importancia que tiene la investigación, la innovación y el desarrollo tecnológico en el avance socioeconómico y cultural de la sociedad.

Resultados de aprendizaje			
Competencias de materia (Resultados de aprendizaje)	Competencias de la 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



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 A40	B1 B3	C3 C4 C6 C7 C8

Contenidos	
Tema	Subtema
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			
Metodologías / pruebas	Horas presenciales	Horas no presenciales / trabajo autónomo	Horas totales
Prácticas de laboratorio	14	42	56
Taller	7	14	21
Prueba objetiva	2	6	8
Sesión magistral	21	42	63
Atención personalizada	2	0	2

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

Metodologías	
Metodologías	Descripción



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.
Taller	Realizaranse como complemento de todas as demais actividades, nalgúns casos de forma autónoma por parte do alumno e noutras ocasións dirixidas polo profesor.
Prueba objetiva	Proba na que se evaluarán os coñecementos adquiridos tanto na parte teórica como na parte práctica da materia.
Sesión magistral	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
Taller 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 desenvoltura 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>

### Evaluación

Metodoloxías	Descrición	Calificación
Taller	Otras actividades evaluables, cuyo contenido se desarrollarán en las horas de TGR.	20
Prueba objetiva	Ejercicio escrito	40
Prácticas de laboratorio	Trabajos de laboratorio	40

### Observaciones evaluación

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.

El 40% restante se reparte entre las prácticas y cualquier otra actividad de evaluación que se realice a lo largo del curso.

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.

En el caso del alumnado a tiempo parcial, no se penalizará la inasistencia a clases de prácticas y TGR que estén debidamente justificadas.

### Fuentes de información

<b>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>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>

### Recomendaciones

Asignaturas que se recomienda haber cursado previamente



Asignaturas que se recomienda cursar simultáneamente
Procesamiento de Lenguajes/614G01067
Asignaturas que continúan el temario
Paradigmas de Programación/614G01014 Teoría de la computación/614G01039
Otros comentarios

(\*) 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