UNIVERSIDADE DA CORUÑA

Guía Docente				
Datos Identificativos				2015/16
Asignatura (*)	Deseño das Linguaxes de Programación		Código	614G01065
Titulación				
Descriptores				
Ciclo	Período	Curso	Tipo	Créditos
Grao	1º cuadrimestre	Cuarto	Obrigatoria	6
Idioma	Castelán			
Modalidade docente	Presencial			
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	En esta asignatura se tratan aspectos de la especificación y diseño de lenguajes de programación: * Criterios de diseño en las principales estructuras de control y de datos de los lenguajes de programación * Diseño de lenguajes de programación orientados a objetos. * Modelos para la definición formal de la semántica de los lenguajes de programación * Especificación formal de los sistemas de tipos. Relaciones de subtipado * Computabilidad. Análisis de complejidad y su relación con el diseño de lenguajes de programación.			

Competencias / Resultados do título	
Código	Competencias / Resultados do título

Resultados da aprendizaxe			
Resultados de aprendizaxe	Competencias / Resultados do título		
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	C6
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	C6
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	C6
Introducir el lambda-cálculo, tipado y no tipado, como núcleo fundamental de los lenguajes de programación.	A39 A40	B1	C2 C6
Comprender los fundamentos formales de los sistemas de tipado y subtipado	A39 A40	B1	C2 C6
Manejar de forma práctica los conceptos de complejidad algorítmica y sus implicaciones en el diseño de lenguajes de programación	A39 A40	B1	C6

Contidos	
Temas	Subtemas
Principios de diseño de lenguajes de programción	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 axiómática Introducción al lambda cálculo Lambda cálculo tipado
Computabilidad y complejidad	Computabilidad y lambda cálculo Clases de complejidad

Planificación				
Metodoloxías / probas	Competencias / Resultados	Horas lectivas (presenciais e virtuais)	Horas traballo autónomo	Horas totais
Prácticas de laboratorio	A39 C2 C6	14	42	56
Obradoiro	B1 C6	7	14	21
Proba obxectiva	A40 B1	2	6	8
Sesión maxistral	A40 C2	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 obxetivo de transmitir coñecemento así como de estimular o razoamento crítico do estudiante.

Atención personalizada	
Metodoloxías	Descrición
Obradoiro Prácticas de laboratorio	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. 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.

Avaliación			
Metodoloxías	Competencias / Resultados	Descrición	**Cualificación**
Obradoiro	B1 C6	Otras actividades evaluables, cuyo contenido se desarrollarán en las horas de TGR.	20
Proba obxectiva	A40 B1	Ejercicio escrito	40
Prácticas de laboratorio	A39 C2 C6	Trabajos de laboratorio	40

Observacións avaliación
La parte teórica 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. Se podrán obtener puntos adicionales por realizar las actividades en inglés (por ejemplo, entregar el informe de una práctica en inglés, presentar una práctica en inglés, etc). En ningún caso se penalizará por realizar las actividades en español y/o gallego.

Fontes de información	
Bibliografía básica	- Michael L. Scott (2009). Programming Language Pragmatics. Third edition. Morgan Kaufmann Publish- ers, Burlington, MA - Kim B. Bruce (2002). Foundations of Object-Oriented Languages: Types and Semantics. The MIT Press, Cambridge, MA - Benjamin C. Pierce (2002). Types and Programming Languages. The MIT Press, Cambridge, MA - Fortnow, Lance (2013). P, NP, and the search for the impossible. Princeton University Press
Bibliografía complementaria	- David A. Watt (2004). Programming Language Design Concepts. ohn Wiley and sons, Chichester, West Suusex, England - Franklyn A. Turbak and David K. Gifford (2008). Design Concepts in Programming Languages. MIT Press, Cambridge, MA - Robert W. Sebesta (2010). Concepts of Programming Languages. Pearson

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

(*)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