



Guía Docente				
Datos Identificativos			2019/20	
Asignatura (*)	Deseño das Linguaxes de Programación		Código	614G01065
Titulación				
Descritores				
Ciclo	Período	Curso	Tipo	Créditos
Grao	1º cuatrimestre	Cuarto	Optativa	6
Idioma	Castelán			
Modalidade docente	Presencial			
Prerrequisitos				
Departamento	Ciencias da Computación e Tecnoloxías da InformaciónComputación			
Coordinación	Alonso Pardo, Miguel angel		Correo electrónico	miguel.alonso@udc.es
Profesorado	Alonso Pardo, Miguel angel		Correo electrónico	miguel.alonso@udc.es
	Graña Gil, Jorge			jorge.grana@udc.es
	Vilares Ferro, Jesus			jesus.vilares@udc.es
Web	moodle.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 do título	
Código	Competencias do título

Resultados da aprendizaxe			
Resultados de aprendizaxe	Competencias do título		
Introducir o lambda-cálculo, tipado e non tipado, como núcleo fundamental das linguaxes de programación.	A39 A40	B1	C2 C6
Comprender os fundamentos formais dos sistemas de tipado e subtipado	A39 A40	B1	C2 C6
Comprender e dominar os principios de deseño das linguaxes orientadas a obxectos a das implicacións que as eleccións de deseño teñen no desenvolvemento de programas	A39 A40	B1	C6
Manexar os principios de dxseño das principais estruturas de control das linguaxes de programación a as súas implicacións no desenvolvemento de programas	A39 A40	B1	C6
Manexar os principios de deseño das principaies estruturas de datos das linguaxes de programación e as súas implicacións no desenvolvemento de programas	A39 A40	B1	C6

Contidos	
Temas	Subtemas
Formalización dos sistemas de tipos	Semántica operacional, denotacional e axiomática Lambda cálculo non tipado Lambda cálculo tipado Subtipado
Linguaxes orientadas a obxectos	Fundamentos de deseño de linguaxes orientadas a obxectos Problemas de tipado das linguaxes orientadas a obxectos



Principios de deseño de linguaxes de programación	Nomes, ámbito e ligazón Fluxo de control Tipos de datos Subrutinas
Computabilidade e complexidade	Computabilidade e lambda-cálculo Clases de complexidade

Planificación				
Metodoloxías / probas	Competencias	Horas presenciais	Horas non presenciais / 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 obxectivo de transmitir coñecemento así como de estimular o razoamento crítico do estudante.

Atención personalizada	
Metodoloxías	Descrición
Obradoiro Prácticas de laboratorio	O desenvolvemento das clases farase de acordo co progreso dos alumnos nas capacidades de comprensión e asimilación dos contidos ensinados, combinando o progreso xeral da clase cunha atención específica aos alumnos que presentan grandes dificultades na tarefa de aprendizaxe e con un apoio adicional a aqueles que presentan maior facilidade e queren ampliar o coñecemento. Dada a natureza personalizada das titorías, estas non deberían dedicarse a ampliar o contido con novos conceptos, senón para aclarar os conceptos xa discutidos. O profesor tamén debe usalos como unha interacción que lle permita sacar conclusións sobre o grao de asimilación da materia polos estudantes.

Avaliación			
Metodoloxías	Competencias	Descrición	Cualificación
Obradoiro	B1 C6	Otras actividades evaluables, cuyo contenido se desarrollará en las horas de TGR	20
Proba obxectiva	A40 B1	Examen escrito	40
Prácticas de laboratorio	A39 C2 C6	Trabajos de laboratorio	40

Observacións avaliación



A parte teórica da materia computa o 40% da nota. As actividades dos TGR computan o 20% da nota. A avaliación dos TGR farase xunto co exame teórico.

O 40% restante repártese entre as prácticas e calquera outra actividade de avaliación realizada ao longo do curso.

Para aprobar a materia é necesario aprobar todas e cada unha das seccións da avaliación. Para a segunda oportunidade, manteranse os resultados obtidos en cada sección da primeira oportunidade.

No caso de estudantes a tempo parcial, a falta de asistencia ás clases de prácticas e TGR xustificadas non será penalizada.

Pódense obter puntos adicionais por levar a cabo as actividades en inglés (por exemplo, enviando o informe dunha práctica en inglés, presentando unha práctica en inglés, etc.). En ningún caso se penalizará por realizar actividades en castelán e/ou galego.

Fontes de información

Bibliografía básica	- Benjamin C. Pierce (2002). Types and Programming Languages. The MIT Press, Cambridge, MA - Kim B. Bruce (2002). Foundations of Object-Oriented Languages: Types and Semantics. The MIT Press, Cambridge, MA - Michael L. Scott (2009). Programming Language Pragmatics. Third edition. Morgan Kaufmann Publishers, Burlington, MA - Fortnow, Lance (2013). P, NP, and the search for the impossible. Princeton University Press
Bibliografía complementaria	- 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 - David A. Watt (2004). Programming Language Design Concepts. John Wiley and sons, Chichester, West Sussex, England

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