



Guía Docente				
Datos Identificativos				2012/13
Asignatura (*)	Paradigmas de Programación	Código	614G01014	
Titulación	Grao en Enxeñaría Informática			
Descritores				
Ciclo	Período	Curso	Tipo	Créditos
Grao	1º cuatrimestre	Segundo	Obrigatoria	6
Idioma	Castelán			
Prerrequisitos				
Departamento	Computación			
Coordinación	Molinelli Barba, Jose Maria	Correo electrónico	jose.molinelli@udc.es	
Profesorado	Alonso Ríos, David	Correo electrónico	david.alonso@udc.es	
	Graña Gil, Jorge		jorge.grana@udc.es	
	Molinelli Barba, Jose Maria		jose.molinelli@udc.es	
	Ortega Hortas, Marcos		m.ortega@udc.es	
	Vilares Ferro, Jesus		jesus.vilares@udc.es	
Web	https://campusvirtual.udc.es/moodle/			
Descrición xeral	Resolución de problemas usando diferentes técnicas de programación: estruturada, orientada a obxectos, declarativa, etc.			

Competencias da titulación	
Código	Competencias da titulación
A3	Capacidade para comprender e dominar os conceptos básicos de matemática discreta, lóxica, algorítmica e complexidade computacional e a súa aplicación para a resolución de problemas propios da enxeñaría.
A7	Capacidade para deseñar, desenvolver, seleccionar e avaliar aplicacións e sistemas informáticos que aseguren a súa fiabilidade, seguranza e calidade, conforme a principios éticos e á lexislación e normativa vixente.
A13	Coñecemento, deseño e utilización de forma eficiente dos tipos e estruturas de datos máis adecuados á resolución dun problema.
A14	Capacidade para analizar, deseñar, construír e manter aplicacións de forma robusta, segura e eficiente, elixindo o paradigma e as linguaxes de programación máis adecuados.
B9	Capacidade de resolución de problemas
B11	Capacidade de análise e síntese
C2	Dominar a expresión e a comprensión de forma oral e escrita dun idioma estranxeiro.
C3	Utilizar as ferramentas básicas das tecnoloxías da información e as comunicacións (TIC) necesarias para o exercicio da súa profesión e para a aprendizaxe ao longo da súa vida.
C4	Desenvolverse para o exercicio dunha cidadanía aberta, culta, crítica, comprometida, democrática e solidaria, capaz de analizar a realidade, diagnosticar problemas, formular e implantar solucións baseadas no coñecemento e orientadas ao ben común.
C6	Valorar criticamente o coñecemento, a tecnoloxía e a información dispoñible para resolver os problemas cos que deben enfrontarse.
C7	Asumir como profesional e cidadán a importancia da aprendizaxe ao longo da vida.
C8	Valorar a importancia que ten a investigación, a innovación e o desenvolvemento tecnolóxico no avance socioeconómico e cultural da sociedade.

Resultados da aprendizaxe			
Competencias de materia (Resultados de aprendizaxe)			Competencias da titulación
Capacidade para deseñar, desenvolver, seleccionar e avaliar aplicacións e sistemas informáticos que aseguren a súa fiabilidade, seguranza e calidade, conforme a principios éticos e á lexislación e normativa vixente.	A7		
Coñecemento, deseño e utilización de forma eficiente dos tipos e estruturas de datos máis adecuados á resolución dun problema.	A13		
Capacidade para analizar, deseñar, construír e manter aplicacións de forma robusta, segura e eficiente, elixindo o paradigma e as linguaxes de programación máis adecuados.	A14		



Capacidade para comprender e dominar os conceptos básicos de matemática discreta, lóxica, algorítmica e complexidade computacional e a súa aplicación para a resolución de problemas propios da enxeñaría.	A3		
C2,C3,C4,C6,C7,C8 (ver competencias titulación)			C2 C3 C4 C6 C7 C8
B9, B11 (ver competencias titulación)		B9 B11	

Contidos	
Temas	Subtemas
Programacion Declarativa: Programación Funcional	<p>Tipos y valores. Expresiones y definiciones.</p> <p>Pattern-matching.</p> <p>Funciones. Funciones recursivas. Terminación. Recursividad terminal. Currying. Funciones de orden superior.</p> <p>Tipos parametrizados. Tipos recursivos. Polimorfismo</p> <p>Transparencia referencial.</p> <p>Excepciones.</p>
Programación imperativa	<p>Estado de la máquina. Variables. Asignación.</p> <p>Programación estructurada. Estructuras de control: Composición secuencial, alternativa e iterativa.</p> <p>Procedimientos y funciones. Paso de parámetros por referencia y por valor. Efectos colaterales.</p> <p>Programación imperativa vs. declarativa.</p>
Programación Orientada a Objetos.	<p>Objetos, atributos y métodos.</p> <p>Clases y herencia.</p> <p>Polimorfismo.</p> <p>Programación Orientada a Objetos vs. imperativa</p> <p>Programación Orientada a Objetos vs. declarativa</p>



El lenguaje de programación Objective Caml	<p>Programación funcional, imperativa y orientada a objetos en Ocaml</p> <p>Los compiladores de Ocaml</p> <p>Entrada / Salida</p> <p>Módulos y librerías</p> <p>Abstracción, encapsulación y compilación separada. Módulos, interfaces y signaturas.</p>
--------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Planificación			
Metodoloxías / probas	Horas presenciais	Horas non presenciais / traballo autónomo	Horas totais
Sesión maxistral	30	20	50
Discusión dirixida	8	0	8
Proba obxectiva	4	16	20
Traballos tutelados	2	20	22
Prácticas de laboratorio	20	20	40
Atención personalizada	10	0	10

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

Metodoloxías	
Metodoloxías	Descrición
Sesión maxistral	Exposición en el aula de los contenidos básicos de la asignatura.
Discusión dirixida	En las horas de tutorías en grupos reducidos, se discutirá con los estudiantes el planteamiento y resolución de problemas avanzados.
Proba obxectiva	Examen escrito.
Traballos tutelados	El estudiante podrá acordar con el profesor la preparación de ciertos trabajos para su presentación y discusión con otros estudiantes durante las horas de tutorías en grupos reducidos.
Prácticas de laboratorio	Ejercicios de programación para la puesta en práctica de lo visto en las clases magistrales con atención personalizada por parte del profesor de prácticas en horario de laboratorio.

Atención personalizada	
Metodoloxías	Descrición
Prácticas de laboratorio Traballos tutelados	<p>Durante el horario de prácticas de laboratorio se supervisará el trabajo de los estudiantes y se les asesorará en la resolución de los ejercicios.</p> <p>Se asesorará personalmente a los estudiantes en la preparación de los trabajos tutelados para su presentación en las horas de tutorías en grupos reducidos.</p> <p>El profesor intentará solucionar aquellas dudas que puedan surgir respecto al temario de la asignatura.</p>

Avaliación		
Metodoloxías	Descrición	Cualificación
Prácticas de laboratorio	Asistencia, realización y entrega de prácticas de laboratorio.	20
Proba obxectiva	Examen escrito	60



Traballos tutelados	Con la realización de trabajos tutelados y su defensa y discusión durante las horas de Tutorías en Grupos Reducidos podrá consolidarse hasta un 20% de la nota final. El porcentaje no consolidado pasará a computarse en la prueba objetiva. La valoración del examen escrito se realizará por el porcentaje que reste hasta le 80 %	20
---------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----

Observacións avaliación

Fontes de información

Bibliografía básica	<ul style="list-style-type: none">- WIKSTRÖM, A. (). Functional Programming Using Standard ML. Prentice Hall- Joshua B. Smith (2006). Practical Ocaml. Apress
Bibliografía complementaria	<ul style="list-style-type: none">- Thérèse Accart Hardin and Véronique Donzeau-Gouge Viguié (). Concepts et outils de programmation. InterEditions- Luc Albert (1997). Cours et exercices d'informatique. Thomson Publishing International, Paris- WEIS, P. & LEROY, X. (1993). Le Langage Caml. InterEditions- PAULSON, L. C. (1991). ML for the Working Programmer. Cambridge University Press.- Michel Quercia (2000). Nouveaux exercices d'algorithmique. Éditions Vuibert, Paris- Jacques Rouablé (1997). Programmation en Caml. Eyrolles, Paris- Philippe Narbe (2005). Programmation fonctionnelle, générique et objet: une introduction avec le langage OCaml. Vuibert, Paris- COUSINEAU, G. & MAUNY, M. (1998). The functional Approach to Programming. Cambridge University Press.

Recomendacións

Materias que se recomenda ter cursado previamente

Concorrencia e Paralelismo/614G01018

Sistemas Intelixentes/614G01020

Materias que se recomenda cursar simultaneamente

Algoritmos/614G01011

Deseño Software/614G01015

Materias que continúan o temario

Programación I/614G01001

Matemática Discreta/614G01004

Programación II/614G01006

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