



Guía docente				
Datos Identificativos				2013/14
Asignatura (*)	Paradigmas de Programación	Código	614G01014	
Titulación	Grao en Enxeñaría Informática			
Descritores				
Ciclo	Periodo	Curso	Tipo	Créditos
Grado	1º cuatrimestre	Segundo	Obligatoria	6
Idioma	Castellano			
Prerrequisitos				
Departamento	Computación			
Coordinador/a	Molinelli Barba, Jose Maria	Correo electrónico	jose.molinelli@udc.es	
Profesorado	Graña Gil, Jorge	Correo electrónico	jorge.grana@udc.es	
	Molinelli Barba, Jose Maria		jose.molinelli@udc.es	
	Paris Fernandez, Javier		javier.paris@udc.es	
	Vilares Ferro, Jesus		jesus.vilares@udc.es	
Web	campusvirtual.udc.es/moodle/			
Descripción general	Resolución de problemas usando diferentes técnicas de programación: estructurada, orientada a objetos, declarativa, etc.			

Competencias de la titulación	
Código	Competencias de la titulación
A3	Capacidad para comprender y dominar los conceptos básicos de matemática discreta, lógica, algorítmica y complejidad computacional, y su aplicación para la resolución de problemas propios de la ingeniería.
A7	Capacidad para diseñar, desarrollar, seleccionar y evaluar aplicaciones y sistemas informáticos, asegurando su fiabilidad, seguridad y calidad, conforme a principios éticos y a la legislación y normativa vigente.
A13	Conocimiento, diseño y utilización de forma eficiente de los tipos y estructuras de datos más adecuados a la resolución de un problema.
A14	Capacidad para analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados.
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
Capacidade para deseñar, desenvolver, seleccionar e avaliar aplicacións e sistemas informáticos que aseguren a súa fiabilidade, seguraza 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 lista competencias titulación)			C2 C3 C4 C6 C7 C8
B9, B11 (ver lista competencias titulación)		B1 B3	

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



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

Planificación			
Metodologías / pruebas	Horas presenciales	Horas no presenciales / trabajo autónomo	Horas totales
Sesión magistral	30	20	50
Discusión dirigida	8	0	8
Prueba objetiva	4	16	20
Trabajos tutelados	2	20	22
Prácticas de laboratorio	20	20	40
Atención personalizada	10	0	10

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

Metodologías	
Metodologías	Descripción
Sesión magistral	Exposición en el aula de los contenidos básicos de la asignatura.
Discusión dirigida	En las horas de tutorías en grupos reducidos, se discutirá con los estudiantes el planteamiento y resolución de problemas avanzados.
Prueba objetiva	Examen escrito.
Trabajos 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	
Metodologías	Descripción
Prácticas de laboratorio Trabajos tutelados	Durante el horario de prácticas de laboratorio se supervisará el trabajo de los estudiantes y se les asesorará en la resolución de los ejercicios. Se asesorará personalmente a los estudiantes en la preparación de los trabajos tutelados para su presentación en las horas de tutorías en grupos reducidos.

Evaluación		
Metodologías	Descripción	Calificación
Prácticas de laboratorio	Asistencia, realización e entrega de prácticas de laboratorio.	20
Prueba objetiva	Exame escrito	60



Trabajos 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 el 80 %	20
--------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----

Observaciones evaluación

Fuentes de información

Básica	<ul style="list-style-type: none">- WIKSTRÖM, A. (). Functional Programming Using Standard ML. Prentice Hall- Joshua B. Smith (2006). Practical Ocaml. Apress
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.

Recomendaciones

Asignaturas que se recomienda haber cursado previamente

Concurrencia y Paralelismo/614G01018

Sistemas Inteligentes/614G01020

Asignaturas que se recomienda cursar simultáneamente

Algoritmos/614G01011

Diseño Software/614G01015

Asignaturas que continúan el temario

Programación I/614G01001

Matemática Discreta/614G01004

Programación II/614G01006

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