Guia docente				
Datos Identificativos				2020/21
Asignatura (*)	Paradigmas de Programación		Código	614G01014
Titulación	Grao en Enxeñaría Informática			
Descriptores				
Ciclo	Periodo	Curso	Tipo	Créditos
Grado	1º cuatrimestre	Segundo	Obligatoria	6
Idioma	Castellano			
Modalidad docente	Híbrida			
Prerrequisitos				
Departamento	Ciencias da Computación e Tecnoloxías da InformaciónComputación			
Coordinador/a	Graña Gil, Jorge		Correo electrónico	jorge.grana@udc.es
Profesorado	Graña Gil, Jorge Molinelli Barba, Jose Maria Paris Fernandez, Javier Vilares Ferro, Jesus		Correo electrónico	jorge.grana@udc.es jose.molinelli@udc.es javier.paris@udc.es jesus.vilares@udc.es
Web	moodle.udc.es			
Descripción general	Resolución de problemas usando diferentes técnicas de programación: estructurada, orientada a objetos, declarativa, etc.			
Plan de contingencia	1. Modificaciones en los contenidos: Ninguna. 2. Metodologías: *Metodologías docentes que se mantienen: Todas. *Metodologías docentes que se modifican: En caso de necesidad, todas las metodologías empleadas podrían aplicarse de modo no presencial. 3. Mecanismos de atención personalizada al alumnado: Atención continuada en Teams, Moodle y correo electrónico. 4. Modificaciones en la evauiación: No son necesarias. *Observaciones de evaluación: Ninguna. 5. Modificaciones de la bibliografía o webgrafía: Ninguna.			

Competencias del título	
Código	Competencias del título
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
C6	Valorar críticamente el conocimiento, la tecnología y la información disponible para resolver los problemas con los que deben enfrentarse.

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			
Resultados de aprendizaje	Competencias del título		
Conocer los fundamentos y principios básicos de la programación, incluyendo variables, tipos, expresiones, estructuras de control, estructuras de datos y recurrencia.	A7 A13	B1	C6 C8
Emplear y aplicar los diferentes paradigmas de programación para la resolución de problemas.	A7 A14	B1	C6 C8

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	Competéncias	Horas presenciales	Horas no presenciales / trabajo autónomo	Horas totales
Sesión magistral	A7 A13 A14 B1 C6 C8	30	20	50
Prueba objetiva	A13 A14 B1	4	16	20
Prácticas de laboratorio	A7 A13 A14 B1	20	20	40
Prueba práctica	A7 A13 A14 B1 C6 C8	10	20	30
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.
Prueba objetiva	Examen escrito.
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.
Prueba práctica	Pruebas prácticas presenciales periódicas, aproximadamente una de cada tres semanas, realizadas en el laboratorio.

Atención personalizada	
Metodologías	Descripción
Prácticas de laboratorio Prueba práctica	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 las pruebas prácticas para su realización en las horas reservadas para las mismas. El profesor intentará solucionar aquellas dudas que puedan surgir respecto al temario de la materia.

Evaluación			
Metodologías	Competéncias	Descripción	**Calificación**
Prácticas de laboratorio	A7 A13 A14 B1	Asistencia, realización y entrega de prácticas de laboratorio.	20
Prueba objetiva	A13 A14 B1	Examen escrito.	60

Prueba práctica	A7 A13 A14 B1 C6 C8	Con la realización de las pruebas prácticas periódicas se podrá consolidar 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	- WIKSTRÖM, A. (). Functional Programming Using Standard ML. Prentice Hall - John Whitington (2013). OCaml from the very beginning. Coherent Press - Andrei De Araújo Formiga (2015). OCaml: Programação funcional na prática. Casa de Código Manual de Objective CamlManual de Objective Caml
Complementária	- WEIS, P. & LEROY, X. (1993). Le Languaje Caml. InterEditions - COUSINEAU, G, & MAUNY, M. (1998). The functional Approach to Programming. Cambridge University Press. - John Whitington (2014). More OCaml. Algorithms, Methods & Diversions. Coherent Press - Yaron Minsky, Anil Madhavapeddy & jason Hickey (2013). Real World OCaml. O'Reilly - PAULSON, L. C. (1991). ML for the Working Programmer. Cambridge University Press. - Michel Quercia (2000). Nouveaux exercices d'algorithmique. Éditions Vuibert, Paris - Philippe Narbe (2005). Programmation fonctionnelle, générique et objet: une introduction avec le langage OCaml. Vuibert, Paris - Jacques Rouablé (1997). Programmation en Caml. Eyrolles, Paris - Luc Albert (1997). Cours et exercices d'informatique. Thomson Publishing International, Paris - Joshua B. Smith (2006). Practical OCaml. Apress DOWNEY, A.; MONJE, N.: Think OCaml. How to Think Like a (Functional) ProgrammerCHAILLOUX, E.; MANOURY, P. & PAGANO, B.: Developing Applications With Objective Caml.DOWNEY, A.; MONJE, N.: Think OCaml. How to Think Like a (Functional) ProgrammerCHAILLOUX, E.; MANOURY, P. & PAGANO, B.: Developing Applications With Objective Caml.

Recomendaciones
Asignaturas que se recomienda haber cursado previamente
Programación I/614G01001 Matemática Discreta/614G01004 Programación II/614G01006
Asignaturas que se recomienda cursar simultáneamente
Algoritmos/614G01011 Diseño Software/614G01015
Asignaturas que continúan el temario
Concurrencia y Paralelismo/614G01018 Sistemas Inteligentes/614G01020
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