



Guía Docente				
Datos Identificativos				2023/24
Asignatura (*)	Paradigmas de Programación		Código	614G01014
Titulación				
Descriptorios				
Ciclo	Período	Curso	Tipo	Créditos
Grao	1º cuadrimestre	Segundo	Obrigatoria	6
Idioma	Castelán			
Modalidade docente	Presencial			
Prerrequisitos				
Departamento	Ciencias da Computación e Tecnoloxías da InformaciónComputación			
Coordinación	Graña Gil, Jorge		Correo electrónico	jorge.grana@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			
Descrición xeral	Resolución de problemas usando diferentes técnicas de programación: estruturada, orientada a obxectos, declarativa, etc.			

Competencias / Resultados do título	
Código	Competencias / Resultados do título

Resultados da aprendizaxe			
Resultados de aprendizaxe	Competencias / Resultados do título		
Coñecer os fundamentos e principios básicos da programación, incluíndo variables, tipos, expresións, estruturas de control, estruturas de datos e recurrencia.	A7	B1	C6
	A13		C8
Empregar e aplicar os diferentes paradigmas de programación para a resolución de problemas.	A7	B1	C6
	A14		C8

Contidos	
Temas	Subtemas
Programación Declarativa: Programación Funcional	Tipos e valores. Expresións e definicións. "Pattern-matching". Funcións. Funcións recursivas. Terminación. Recursividade terminal. "Currying". Funcións de orde superior. Tipos parametrizados. Tipos recursivos. Polimorfismo. Transparencia referencial. Excepcións.



Programación imperativa	Estado da máquina. Variables. Asignación. Programación estruturada. Estructuras de control: Composición secuencial, alternativa e iterativa. Procedimentos e funcións. Paso de parámetros por referencia e por valor. Efectos colaterais. Programación imperativa vs. declarativa.
Programación Orientada a Obxectos	Obxectos, atributos e métodos. Clases e herdanza. Polimorfismo. Programación Orientada a Obxectos vs. imperativa. Programación Orientada a Obxectos vs. declarativa.
A linguaxe de programación Objective Caml	Programación funcional, imperativa e orientada a obxectos en Ocaml. Os compiladores de Ocaml. Entrada / Saída. Módulos e librerías. Abstracción, encapsulación e compilación separada. Módulos, interfaces e signatures.

Planificación				
Metodoloxías / probas	Competencias / Resultados	Horas lectivas (presenciais e virtuais)	Horas traballo autónomo	Horas totais
Sesión maxistral	A7 A13 A14 B1 C6 C8	30	20	50
Proba obxectiva	A13 A14 B1	4	16	20
Prácticas de laboratorio	A7 A13 A14 B1	20	20	40
Proba práctica	A7 A13 A14 B1 C6 C8	10	20	30
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 na aula dos contidos básicos da materia.
Proba obxectiva	Exame escrito.
Prácticas de laboratorio	Exercicios de programación para a posta en práctica do visto nas clases maxistras, con atención personalizada por parte do profesor de prácticas en horario de laboratorio.
Proba práctica	Aproximadamente unha de cada tres semanas, realizaranse probas prácticas no laboratorio ou ben proporase a resolución de problemas avanzados.



Atención personalizada

Metodoloxías	Descrición
Prácticas de laboratorio Proba práctica	Durante o horario de prácticas de laboratorio supervisarse o traballo dos estudantes e asesoraráselles na resolución dos exercicios. Asesorarase persoalmente aos estudantes na preparación das probas prácticas para a súa realización nas horas reservadas para as mesmas. O profesor tentará solucionar aquelas dúbidas que poidan xurdir respecto ao temario da materia.

Avaliación

Metodoloxías	Competencias / Resultados	Descrición	Cualificación
Prácticas de laboratorio	A7 A13 A14 B1	Asistencia, realización e entrega de prácticas de laboratorio.	20
Proba obxectiva	A13 A14 B1	Exame escrito.	60
Proba práctica	A7 A13 A14 B1 C6 C8	Coa realización das probas prácticas periódicas poderá consolidarse até un 20% da nota final. A porcentaxe non consolidada pasará a computarse na proba obxectiva. A valoración do exame escrito realizarase pola porcentaxe que reste até o 80%.	20

Observacións avaliación

A detección de copias e plaxios poderá levar a perda de todos os puntos obtidos nas seccións de prácticas de laboratorio e probas prácticas.
--

Fontes de información

Bibliografía básica	- WIKSTRÖM, A. (1988). Functional Programming Using Standard ML. Prentice Hall - John Whittington (2013). OCaml from the very beginning. Coherent Press - Andrei De Araújo Formiga (2015). OCaml: Programação funcional na prática. Casa de Código - Functional Programming in OCaml (libro de texto do curso CS3110 da Universidade de Cornell) http://www.cs.cornell.edu/courses/cs3110/2021sp/textbook/Manual de Objective Caml - Functional Programming in OCaml (libro de texto do curso CS3110 da Universidade de Cornell) http://www.cs.cornell.edu/courses/cs3110/2021sp/textbook/Manual de Objective Caml
Bibliografía complementaria	- WEIS, P. & LEROY, X. (1993). Le Langage CamL. InterEditions - COUSINEAU, G. & MAUNY, M. (1998). The functional Approach to Programming. Cambridge University Press. - John Whittington (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 - Richard Bird (2014). Thinking Functionally With Haskell. Cambridge University Press - Richard Bird & Jeremy Gibbons (2020). Algorithm Design With Haskell. Cambridge University Press - DOWNEY, A.; MONJE, N.: Think OCaml. How to Think Like a (Functional) Programmer - CHAILLOUX, E.; MANOURY, P. & PAGANO, B.: Developing Applications With Objective CamL. - DOWNEY, A.; MONJE, N.: Think OCaml. How to Think Like a (Functional) Programmer - CHAILLOUX, E.; MANOURY, P. & PAGANO, B.: Developing Applications With Objective CamL.



Recomendacións

Materias que se recomenda ter cursado previamente

Programación I/614G01001
Matemática Discreta/614G01004
Programación II/614G01006

Materias que se recomenda cursar simultaneamente

Algoritmos/614G01011
Deseño Software/614G01015

Materias que continúan o temario

Concorrencia e Paralelismo/614G01018
Sistemas Intelixentes/614G01020

Observacións

-Segundo se recolle nas distintas normativas de aplicación para a docencia universitaria incorporárase a perspectiva de xénero nesta materia (usarase linguaxe non sexista, propiciárase a intervención en clase de alumnos e alumnas., etc.).-Traballarase para identificar e modificar prexuízos e actitudes sexistas e influirase na contorna para modificalos e fomentar valores de respecto e igualdade.- No caso de detectar situacións de discriminación por razón de xénero proporáanse accións e medidas para corrixilas.

(*)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