



| Guía Docente          |  |                    |                       |          |
|-----------------------|--|--------------------|-----------------------|----------|
| Datos Identificativos |  |                    |                       | 2022/23  |
| 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   |                    |                       |          |
| 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 do título |   |
|------------------------|---|
| Código                 | Competencias do título  |
| 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.   |
| B1                     | Capacidade de resolución de problemas   |
| C6                     | Valorar criticamente o coñecemento, a tecnoloxía e a información dispoñible para resolver os problemas cos que deben afrontarse.  |
| C8                     | Valorar a importancia que ten a investigación, a innovación e o desenvolvemento tecnolóxico no avance socioeconómico e cultural da sociedade.   |

| Resultados da aprendizaxe  |     |    |                        |
|--|-----|----|------------------------|
| Resultados de aprendizaxe  |     |    | Competencias 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 | <p>Tipos e valores. Expresións e definicións.</p> <p>&amp;quot;Pattern-matching&amp;quot;.</p> <p>Funcións. Funcións recursivas. Terminación. Recursividade terminal.<br/>&amp;quot;Currying&amp;quot;. Funcións de orde superior.</p> <p>Tipos parametrizados. Tipos recursivos. Polimorfismo.</p> <p>Transparencia referencial.</p> <p>Excepcións.</p> |
| Programación imperativa                          | <p>Estado da máquina. Variables. Asignación.</p> <p>Programación estruturada. Estructuras de control: Composición secuencial, alternativa e iterativa.</p> <p>Procedimentos e funcións. Paso de parámetros por referencia e por valor. Efectos colaterais.</p> <p>Programación imperativa vs. declarativa.</p>   |
| Programación Orientada a Obxectos                | <p>Obxectos, atributos e métodos.</p> <p>Clases e herdanza.</p> <p>Polimorfismo.</p> <p>Programación Orientada a Obxectos vs. imperativa.</p> <p>Programación Orientada a Obxectos vs. declarativa.</p>  |
| A linguaxe de programación Objective Caml        | <p>Programación funcional, imperativa e orientada a obxectos en Ocaml.</p> <p>Os compiladores de Ocaml.</p> <p>Entrada / Saída.</p> <p>Módulos e librerías.</p> <p>Abstracción, encapsulación e compilación separada. Módulos, interfaces e signaturas.</p>  |

## Planificación

| Metodoloxías / probas    | Competencias           | Horas presenciais | Horas non presenciais /<br>traballo autónomo | Horas totais |
|--------------------------|------------------------|-------------------|--|--------------|
| Sesión maxistral         | A7 A13 A14 B1 C6<br>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<br>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<br>Proba práctica | Durante o horario de prácticas de laboratorio supervisarase o traballo dos estudantes e asesoraráselles na resolución dos exercicios.<br><br>Asesorarase persoalmente aos estudantes na preparación das probas prácticas para a súa realización nas horas reservadas para as mesmas.<br><br>O profesor tentará solucionar aquelas dúbidas que poidan xurdir respecto ao temario da materia. |

| Avaliación               |                        |  |               |
|--------------------------|------------------------|--|---------------|
| Metodoloxías             | Competencias           | 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<br>C8 | Coa realización das probas prácticas periódicas poderás 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 |
|-------------------------|
|                         |

| Fontes de información      |   |
|----------------------------|---|
| <b>Bibliografía básica</b> | <ul style="list-style-type: none"><li>- WIKSTRÖM, A. (1988). Functional Programming Using Standard ML. Prentice Hall</li><li>- John Whittington (2013). OCaml from the very beginning. Coherent Press</li><li>- Andrei De Araújo Formiga (2015). OCaml: Programação funcional na prática. Casa de Código</li></ul> Functional Programming in OCaml (libro de texto do curso CS3110 da Universidade de Cornell)<br><a href="http://www.cs.cornell.edu/courses/cs3110/2021sp/textbook/Manual de Objective Caml">http://www.cs.cornell.edu/courses/cs3110/2021sp/textbook/Manual de Objective Caml</a><br>Functional Programming in OCaml (libro de texto do curso CS3110 da Universidade de Cornell)<br><a href="http://www.cs.cornell.edu/courses/cs3110/2021sp/textbook/Manual de Objective Caml">http://www.cs.cornell.edu/courses/cs3110/2021sp/textbook/Manual de Objective Caml</a> |



|                                    |   |
|------------------------------------|---|
| <b>Bibliografía complementaria</b> | <ul style="list-style-type: none"><li>- WEIS, P. &amp; LEROY, X. (1993). Le Languaje Caml. InterEditions</li><li>- COUSINEAU, G. &amp; MAUNY, M. (1998). The functional Approach to Programming. Cambridge University Press.</li><li>- John Whittington (2014). More OCaml. Algorithms, Methods &amp; Diversions. Coherent Press</li><li>- Yaron Minsky, Anil Madhavapeddy &amp; Jason Hickey (2013). Real World OCaml. O'Reilly</li><li>- PAULSON, L. C. (1991). ML for the Working Programmer. Cambridge University Press.</li><li>- Michel Quercia (2000). Nouveaux exercices d'algorithmique. Éditions Vuibert, Paris</li><li>- Philippe Narbe (2005). Programmation fonctionnelle, générique et objet: une introduction avec le langage OCaml. Vuibert, Paris</li><li>- Jacques Rouablé (1997). Programmation en Caml. Eyrolles, Paris</li><li>- Luc Albert (1997). Cours et exercices d'informatique. Thomson Publishing International, Paris</li><li>- Joshua B. Smith (2006). Practical OCaml. Apress</li><li>- Richard Bird (2014). Thinking Functionally With Haskell. Cambridge University Press</li><li>- Richard Bird &amp; Jeremy Gibbons (2020). Algorithm Design With Haskell. Cambridge University Press</li><li>DOWNEY, A.; MONJE, N.: Think OCaml. How to Think Like a (Functional) Programmer</li><li>CHAILLOUX, E.; MANOURY, P. &amp; PAGANO, B.: Developing Applications With Objective Caml.</li><li>DOWNEY, A.; MONJE, N.: Think OCaml. How to Think Like a (Functional) Programmer</li><li>CHAILLOUX, E.; MANOURY, P. &amp; PAGANO, B.: Developing Applications With Objective Caml.</li></ul> |
|------------------------------------|---|

## 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

Concorrenza e Paralelismo/614G01018  
Sistemas Intelixentes/614G01020

### 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