



| Guía Docente | | | | |
|-----------------------|--|--------------------|---|----------|
| Datos Identificativos | | | | 2022/23 |
| Asignatura (*) | Fundamentos de Programación I | Código | 614G02004 | |
| Titulación | | | | |
| Descritores | | | | |
| Ciclo | Período | Curso | Tipo | Créditos |
| Grao | 1º cuatrimestre | Primeiro | Formación básica | 6 |
| Idioma | CastelánGalego | | | |
| Modalidade docente | Híbrida | | | |
| Prerrequisitos | | | | |
| Departamento | Ciencias da Computación e Tecnoloxías da Información | | | |
| Coordinación | Rabuñal Dopico, Juan Ramon | Correo electrónico | juan.rabunal@udc.es | |
| Profesorado | Alvarellos González, Alberto José Cedrón Santaefemia, Francisco Abel Rabuñal Dopico, Juan Ramon Rodríguez Tajés, Álvaro | Correo electrónico | alberto.alvarellos@udc.es francisco.cedron@udc.es juan.rabunal@udc.es a.tajes@udc.es | |
| Web | campusvirtual.udc.gal | | | |
| Descrición xeral | Esta materia é unha introdución á programación, na que se ve cómo resolver problemas nunha linguaxe de programación. Nela axúdase ao alumno a comprender os tipos e estruturas de datos básicos, ao mesmo tempo que se sentan as bases para deseñar correctamente un algoritmo. Para asentar os coñecementos fundamentais da programación de forma máis rápida e óptima é necesario empregar unha linguaxe que permita a posta en práctica dos coñecementos adquiridos e sirva de base para o bó desenvolvemento dun programador informático; utilizarase a Linguaxe Python, tanto para as prácticas como para os exemplos teóricos. | | | |

| Competencias / Resultados do título | |
|-------------------------------------|-------------------------------------|
| Código | Competencias / Resultados do título |

| Resultados da aprendizaxe | | | |
|--|-------------------------------------|----------------|----|
| Resultados de aprendizaxe | Competencias / Resultados do título | | |
| Saber analizar problemas e deseñar, programar e depurar algoritmos que os resolvan utilizando unha linguaxe de programación imperativa | A4 | B5 B6 | C1 |
| Coñecer os aspectos básicos que conducen a un bo deseño de programas | A4 | B5 B6 | C1 |
| Saber elixir e utilizar as estratexias de resolución de problemas máis relevantes | A4 | B1 B5 B6 | C1 |
| Coñecer e saber usar linguaxes de programación de relevancia actual | A4 | B1 B5 B6 | C1 |

| Contidos | |
|----------|----------|
| Temas | Subtemas |
| | |



1 CONCEPTOS BÁSICOS

- 1.1 Algoritmos
 - 1.1.1 Representación de algoritmos
- 1.2 Programas
 - 1.2.1 Tipos de programas
- 1.3 Linguaxes de programación
 - 1.3.1 Unha visión histórica
 - 1.3.2 Clasificación das linguaxes
 - 1.3.3 Instrucións máis importantes
 - 1.3.4 Propiedades das linguaxes
- 1.4 Tradutores
- 1.5. Descrición das linguaxes
- 1.6 Estrutura dun programa
- 1.7 Elementos dun programa
 - 1.7.1 Símbolos predefinidos
 - 1.7.2 Símbolos especiais
 - 1.7.3 Identificadores
 - 1.7.4 Etiquetas
 - 1.7.5 Comentarios
 - 1.7.6 Directivas
 - 1.7.7 Constantes
 - 1.7.8 Números
 - 1.7.9 Cadeas de caracteres
 - 1.7.10 Variables: Declaración e iniciación
 - 1.7.11 Variables: Dirección de Memoria
- 1.8 Saída e Entrada
 - 1.8.1 Sentenzas de saída
 - 1.8.2 Sentenzas de entrada
- 1.9 Tipos de datos e operadores
 - 1.9.1 Tipos de datos
 - 1.9.2 Operadores
 - 1.9.3 Expresións
- 1.10 Depuración de programas



| | |
|-------------------------------|---|
| 2 SENTENZAS DE CONTROL | 2.1 Secuencial 2.2 Condicional 2.2.1 A sentenza condicional simple 2.2.2 A sentenza condicional múltiple 2.3 Repetitiva 2.3.1 Introducción 2.3.2 Variables asociadas aos bucles 2.3.3 Funcionamento dos diferentes tipos de bucles 2.3.4 Bucle FOR 2.3.5 Equivalencia entre bucles 2.3.6 Erros nos bucles 2.3.7 Deseño de bucles |
| 3 ARQUITECTURA DUN PROGRAMA | 3.1 Funcións 3.1.1 Tipos de funcións 3.2 Función como argumentos 3.3 Corrutinas 3.4 Recursividade 3.4.1 Natureza da recursividade 3.4.2 Recursión infinita |
| 4 ESTRUTURAS SIMPLES DE DATOS | 4.1 Vectores 4.1.1 Tipo de datos 4.1.2 Declaración dun vector 4.2 Tuplas 4.2.1 Tipo de datos 4.2.2 Operacións con tuplas 4.3 Cadeas de caracteres 4.3.1 Cadeas de lonxitude variable 4.4 Listas e Dicionarios |
| 5 ALMACENAMIENTO PERSISTENTE | 5.1 Ficheiros 5.2 Tipos de ficheiros |

| Planificación | | | | |
|--------------------------|---------------------------|---|-------------------------|--------------|
| Metodoloxías / probas | Competencias / Resultados | Horas lectivas (presenciais e virtuais) | Horas traballo autónomo | Horas totais |
| Sesión maxistral | A4 B1 B5 B6 C1 | 30 | 30 | 60 |
| Prácticas de laboratorio | A4 B1 B5 B6 C1 | 20 | 50 | 70 |
| Seminario | B6 C1 | 8 | 10 | 18 |
| Atención personalizada | | 2 | 0 | 2 |

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

| Metodoloxías | |
|--------------|------------|
| Metodoloxías | Descrición |
| | |



| | |
|--------------------------|--|
| Sesión maxistral | <p>Nas sesións de teoría, o profesor describe os obxectivos e os contidos da materia, para dar unha visión particular do tema a tratar e relacionalo con outros dentro da asignatura</p> <p>Despois desenvólvese o tema correspondente na forma de sesión magistral, axudándose das ferramentas técnicas dispoñibles, facendo fincapé en certas cuestións nas que o alumno debe profundar no seu autoaprendizaje.</p> <p>O obxectivo é que o alumno aprenda a algoritmizar, utilizar as estruturas básicas de datos e resolver sinxelos problemas de programación. Utilizarase como linguaxe de codificación o Python</p> <p>As sesións maxistrais poden ser presenciais a través de plataformas informáticas como TEAMS. Tamén pódense incluír vídeos explicativos de diferentes partes dos contidos teóricos</p> |
| Prácticas de laboratorio | <p>Nas sesións de prácticas o alumno realizará programas en papel para despois codificarlo en Linguaxe Python, executalo e comprobar o seu nivel de corrección.</p> <p>Os enunciados dos programas proporcionarase coa suficiente antelación para que os alumnos poidan aproveitar mellor o seu tempo.</p> <p>É misión do profesor supervisar o código xerado polo alumno para resolver dúbidas, corrixir malos estilos de programación e corrixir erros.</p> |
| Seminario | <p>Nas sesións de seminario realizaranse exercicios e prácticas coa finalidade de detectar nos alumnos lagoas de coñecemento na materia impartida ata ese momento, e dar as explicacións e/ou referencias necesarias para emendalas.</p> <p>As sesións de seminario e para resolver dúbidas poden ser a través de plataformas informáticas como TEAMS</p> |

Atención personalizada

| Metodoloxías | Descrición |
|--|--|
| <p>Sesión maxistral</p> <p>Prácticas de laboratorio</p> <p>Seminario</p> | <p>Tanto nas sesións maxistrais como nos laboratorios de prácticas e nas sesións de seminario levarase unha atención personalizada do alumno, en distintos niveis segundo sexa o tipo de clase, detectando o nivel de asimilación e comprensión dos temas explicados e as prácticas requiridas a implantar.</p> <p>Nas sesións de seminario é onde se pode chegar máis ao alumno para coñecer as lagoas que presente e indicarlle o camiño para cubrilas.</p> <p>Os alumnos que teñan matrícula a tempo parcial deben falar, o comezo do curso, con o/os profesores encargados do seu grupo.</p> |

Avaliación

| Metodoloxías | Competencias / Resultados | Descrición | Cualificación |
|------------------|---------------------------|--|---------------|
| Sesión maxistral | A4 B1 B5 B6 C1 | <p>A nota da asignatura será a suma do obtido na Avaliación Continua (durante as 15 semanas do período lectivo correspondente á asignatura) e o obtido no Exame Final.</p> <p>O EXAME FINAL constará de preguntas, cuestións ou exercicios que o alumno terá que responder. Terá un valor de 6 puntos.</p> <p>O exame oficial, tanto na primeira (xaneiro) como na segunda (xullo) oportunidade constará de preguntas, cuestións ou exercicios que o alumno terá que responder. Devandito Exame Final terá un valor máximo de 6 puntos, que se sumarán ao obtido na Avaliación Continua.</p> | 60 |



| | | | |
|--------------------------|----------------|---|----|
| Prácticas de laboratorio | A4 B1 B5 B6 C1 | Durante as últimas semanas de prácticas do curso realizarase unha proba no laboratorio usando ordenadores que terá un valor máximo de 4 puntos sobre a nota total do curso (puntuada sobre 10). Será necesario que o programa a realizar polo alumno no laboratorio se execute de forma correcta e completa. A nota de AVALIACIÓN CONTINUA (prácticas de laboratorio), está valorada en 4 puntos. Como posibilidade de recuperación da Avaliación Continua (na convocatoria de xullo), o alumno poderá decidir ser examinado dun contido adicional ao examen final, que constará de varios problemas a desenvolver en código Python. De facer esta recuperación eliminarase a puntuación anterior automaticamente | 40 |
|--------------------------|----------------|---|----|

Observacións avaliación

A nota final virá dada pola nota obtida por AVALIACIÓN CONTINUA e a obtida no EXAME FINAL.

Fontes de información

| | |
|------------------------------------|---|
| Bibliografía básica | <ul style="list-style-type: none">- Charles Russell Severance (Autor), Fernando Tardío Muniz (Traductor) (2015). Python para informaticos: Explorando la informacion. .- Charles Russell Severance (2016). Python for Everybody: Exploring Data in Python 3.. Open Textbook Library- Alberto Cuevas Álvarez (2016). Python 3. Curso Práctico. Editorial RA-MA.- Mark Lutz (2013). Learning Python. Quinta edición . O'Reilly Media Inc.- Arturo Montejo Ráez, Salud María Jiménez Zafra (2019). Curso de Programación Python. Editorial Anaya |
| Bibliografía complementaria | <ul style="list-style-type: none">- Mark Summerfield (2010). Python 3.. Editorial Anaya- Sébastien Chazallet (2016). Python 3. Los fundamentos del lenguaje - 2ª edición. Ediciones-ENI- Raúl González Duque (2008). Python para todos.- John V. Guttag (2013). Introduction to Computation and Programming Using Python. The MIT Press |

Recomendacións

Materias que se recomenda ter cursado previamente

Materias que se recomenda cursar simultaneamente

Materias que continúan o temario

Fundamentos de Programación II/614G02009

Observacións



O alumno debe ter en conta que debe realizar unha labor autodidacta moi importante, seguindo o seguinte esquema: Ler, atender, comprender, preguntar, estudar e practicar. Ler: Lea o tema a tratar antes de asistir ás sesións teóricas. É MOI IMPORTANTE! Atender: Atenda en clase, non só estea de corpo presente. Comprender: Comprenda o que se lle di nas sesións de teoría, e si non pregunte. Preguntar: Pregunte todo o que non comprenda, non quede con dúbidas. Estudar: Estude logo das sesións, para reter o comprendido.

 Practicar: Faga moitos programas, os que se lle pidan, suxíran, e outros pola súa conta, tanto en papel como no ordenador. Fundamentos da programación I é unha asignatura que non se pode aprender estudando en dous días. O alumno debe ir madurando os conceptos, facer sobre o papel e na máquina moitos programas, aprendendo tamén dos erros ao realizalos. Esta é unha asignatura que, por medio do sistema de avaliación continua, se pode seguir de forma activa, o ritmo das distintas sesións teóricas e prácticas. Debe facer caso ás indicacións particulares de reforzo de estudo que lle sinala o profesor.

(*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