Guía Docente				
Datos Identificativos				2022/23
Asignatura (*)	Programación I		Código	614G03006
Titulación	Grao en Intelixencia Artificial			
Descriptores				
Ciclo	Período	Curso	Tipo	Créditos
Grao	1º cuadrimestre	Primeiro	Formación básica	6
Idioma	CastelánGalego			
Modalidade docente	Presencial			
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 Martinez Perez, Maria Rabuñal Dopico, Juan Ramon	Correo electrónico	alberto.alvarellos@udc.es maria.martinez@udc.es juan.rabunal@udc.es	
Web	campusvirtual.udc.gal			
Descrición xeral	Esta materia é unha introdución á programación, na que se ve como resolver problemas nunha linguaxe estruturada. Nela axúdase ao alumno para comprender os tipos e estruturas de datos básicos, ao mesmo tempo que sentan as bases para deseñar correctamente un algoritmo. E para asentar os coñecementos fundamentais da programación de forma máis rápida e óptima é necesario utilizar unha linguaxe que permita a posta en práctica dos coñecementos adquiridos e sirva de base para o bo desenvolvemento dun programador informático; utilizarase a Linguaxe de programación Python, tanto para as prácticas como para os exemplos teóricos.			

Competencias do título	
Código	Competencias do título
A2	Capacidade para resolver problemas de intelixencia artificial que precisen algoritmos, aplicando correctamente metodoloxías de desenvolvemento software e deseño centrado en usuario/a.
A3	Capacidade para comprender e dominar os conceptos básicos de lóxica, gramáticas e linguaxes formais para analizar e mellorar as solucións baseadas en intelixencia artificial.
B2	Que o alumnado saiba aplicar os seus coñecementos ao seu traballo ou vocación dunha forma profesional e posúa as competencias que adoitan demostrarse por medio da elaboración e defensa de argumentos e a resolución de problemas dentro da súa área de estudo.
B3	Que o alumnado teña a capacidade de reunir e interpretar datos relevantes (normalmente dentro da súa área de estudo) para emitir xuízos que inclúan unha reflexión sobre temas relevantes de índole social, científica ou ética.
B4	Que o alumnado poida transmitir información, ideas, problemas e solucións a un público tanto especializado como non especializado.
B5	Que o alumnado desenvolva aquelas habilidades de aprendizaxe necesarias para emprender estudos posteriores cun alto grao de autonomía.
B6	Capacidade para concibir, redactar, organizar, planificar, e desenvolver modelos, aplicacións e servizos no ámbito da intelixencia artificial, identificando obxectivos, prioridades, prazos recursos e riscos, e controlando os procesos establecidos.
B7	Capacidade para resolver problemas con iniciativa, toma de decisións, autonomía e creatividade.
B8	Capacidade para deseñar e crear modelos e solucións de calidade baseadas en Intelixencia Artificial que sexan eficientes, robustas, transparentes e responsables.
B9	Capacidade para seleccionar e xustificar os métodos e técnicas adecuadas para resolver un problema concreto, ou para desenvolver e propor novos métodos baseados en intelixencia artificial.
C2	Capacidade de traballo en equipo, en contornas interdisciplinares e xestionando conflitos.
C3	Capacidade para crear novos modelos e solucións de forma autónoma e creativa, adaptándose a novas situacións. Iniciativa e espírito emprendedor.
C6	Capacidade para integrar aspectos xurídicos, sociais, ambientais e económicos inherentes á intelixencia artificial, analizando os seus impactos, e comprometéndose coa procura de solucións compatibles cun desenvolvemento sustentable.

Resultados da aprendizaxe

Resultados de aprendizaxe	Competencias do título		
Coñecer e comprender a importancia dos obxectivos da programación. Coñecer os aspectos xerais sobre as linguaxes e paradigmas da programación. Coñecer pseudocódigo e a sintaxe da linguaxe Phyton utilizado para describir algoritmos e programas. Coñecer os pasos para a realización dun programa e os seus principais compoñentes. Coñecer os tipos de datos básicos usando a Linguaxe Phyton. Coñecer as estruturas de control da programación estruturada e as diferenzas entre elas. Coñecer todos os aspectos relacionados coa realización de funcións. Levar a cabo o proceso que permite, desde a abstracción, implementar código de alta calidade. Aplicar programación modular para resolver problemas específicos no ámbito de IA. Comprender a sintaxe e semántica da linguaxe de programación.	A2 A3	B2 B3 B5 B7 B8 B9	C3 C6
Ser capaz de realizar el seguimiento de un algoritmo (en pseudocódigo) o programa (en Lenguaje Phyton), explicar qué realiza, y encontrar posibles errores. Ser capaz de resolver pequeños algoritmos y programas. A partir del planteamiento de un problema de pequeña-mediana envergadura saber realizar el programa para resolverlo: teniendo en cuenta los objetivos de la programación estructurada. Realizar la descomposición adecuada implementando las funciones y procedimientos necesarios correctamente. Emplear un estilo de programación apropiado: saber hacer buen uso de identificadores, comentarios justos, saber establecer precondiciones y postcondiciones, saber realizar un buen diseño de las interfaces de procedimientos y funciones, saber elegir y utilizar los tipos y estructuras de datos adecuados, saber elegir y utilizar las estructuras de control convenientes. Saber hacer buen conocimiento de la parte del lenguaje que se explique. Adquirir competencias para resolver problemas de forma metodológica y práctica. Identificar y tener la capacidad para seleccionar en un entorno práctico las principales librerías, coger experiencia en la utilización de librerías, que será fundamental en el desarrollo de la profesión donde se necesite programación para la IA y Ciencia de Datos. Analizar las alternativas para afrontar un problema e identificar qué aspectos pueden abordarse, habilidad muy importante en la IA. Manejar técnicas y herramientas de prueba para asegurar la calidad de los resultados	A2 A3	B2 B3 B4 B5 B6 B7 B8 B9	C2 C3

Contidos	
Temas	Subtemas

1 CONCEPTOS BÁSICOS. PARADIGMA IMPERATIVO	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 dás linguaxes 1.3.3 Instrucións máis importantes 1.3.4 Propiedades das linguaxes 1.4 Tradutores 1.5. Descripción 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 inicialización 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 Importar e usar librerías 1.11 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 Introdución 2.3.2 Variables asociadas aos bucles 2.3.3 Funcionamento dos diferentes tipos de bucles 2.3.4 Bucle While 2.3.5 Bucle FOR 2.3.6 Equivalencia entre bucles 2.3.7 Erros nos bucles 2.3.8 Deseño de bucles
3 ARQUITECTURA DUN PROGRAMA	3.1 Funcións 3.1.1 Tipos de funcions 3.1.2 Función como argumentos 3.3 Recursividade 3.3.1 Natureza da recursividade 3.3.2 Recursión infinita
4 ESTRUCTURAS SIMPLES DE DATOS	4.1 Lista e Tuplas 4.1.1 Tipo de datos 4.1.2 Operacións con Listas e Tuplas 4.2 Diccionarios 4.3 Cadeas de caracteres 4.3.1 Cadeas de lonxitude variable
5 ENTRADA / SAÍDA	5.1 Ficheiros 5.2 Tipos de ficheiros
6 TESTEO E PROBAS DE PROGRAMAS	6.1 Depuración de programas 6.2 Detección de erros 6.2.1 En tempo de compilación 6.2.2 En tempo de execución

Planificación				
Metodoloxías / probas	**Competencias**	**Horas presenciais**	**Horas non presenciais / traballo autónomo**	**Horas totais**
Prácticas de laboratorio	A2 A3 B2 B3 B4 B5 B6 B7 B8 B9 C2 C3 C6	28	60	88
Proba obxectiva	A2 A3 B4 B5 B6 B7 B8 B9	2	2	4
Sesión maxistral	A2 A3 B2 B3 B5 B7 B8 B9 C3 C6	28	28	56
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**
Prácticas de laboratorio	Nas sesións de prácticas o alumno diseñará o pseudocódigo do problema a resolver (papel ou ordenador) para despois codificalo en Linguaxe Phyton, compilalo, executalo e comprobar o seu nivel de corrección. Os enunciados dos programas proporcionarase coa suficiente antelación para que os alumnos poidan aproveitar mellor o seu tempo. É misión do profesor supervisar o código xerado polo alumno para resolver dúbidas, corrixir malos estilos de programación e corrixir erros, contando con que o profesor non é un compilador que busca erros.
Proba obxectiva	Para evaluar el aprendizaje, se realizará una prueba escrita que constará de varios ejercicios a realizar en lenguaje de programación Python. Se realizará en las fechas establecidas por la Xunta de Facultade.
Sesión maxistral	Nas sesións de teoría, o profesor describe os obxectivos e os contidos da materia, para dar unha visión particular do tema para tratar e relacionalo con outros dentro da materia Despois desenvólvese o tema correspondente na forma de sesión maxistral, axudándose das ferramentas técnicas dispoñibles, facendo fincapé en certas cuestións nas que o alumno debe profundar no seu autoaprendizaje. 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 Phyton As sesión maxistrais poden ser presenciais ou a través de plataformas informáticas como TEAMS (en casos excepcionais). Tamén se poden incluír vídeos explicativos de diferentes partes dos contidos teóricos

Atención personalizada	
Metodoloxías	**Descrición**
Sesión maxistral Prácticas de laboratorio Proba obxectiva	Tanto nas sesións maxistrais como nos laboratorios de prácticas levará 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. Nas sesións de práctica é onde se pode chegar máis ao alumno para coñecer as lagoas que ten, e indicarlle o camiño para cubrilas. Os alumnos que teñan matrícula a tempo parcial deben, ao comezo do curso, falar co/os profesores encargados do seu grupo.

Avaliación			
Metodoloxías	**Competencias**	**Descrición**	**Cualificación**
Prácticas de laboratorio	A2 A3 B2 B3 B4 B5 B6 B7 B8 B9 C2 C3 C6	Durante as últimas semanas con prácticas do curso realizarase unha proba no laboratorio usando ordenadores que terá un valor máximo de 4 puntos sobre a nota total do curso. Será necesario que o programa para realizar polo alumno no laboratorio compile e execute de forma correcta e completa.	40
Proba obxectiva	A2 A3 B4 B5 B6 B7 B8 B9	O EXAME FINAL, tanto na primeira convocatoria (xaneiro) como na segunda oportunidade (xuño/xullo) constará de varias preguntas ou exercicios que o alumno terá que desenvolver en código Phyton, e terá un valor de 6 puntos.	60

Observacións avaliación

A nota final virá dada pola nota obtida por AVALIACIÓN CONTINUA e a obtida no EXAME FINAL. O Exame Final constará de varias preguntas e problemas a codificar na linguaxe de programación empregada nas sesións prácticas. A realización fraudulenta das probas ou actividades de avaliación, unha vez comprobada, implicará directamente a cualificación de suspenso "0" na materia na convocatoria correspondente, invalidando así calquera cualificación obtida en todas as actividades de avaliación de cara a convocatoria extraordinaria. Para os alumnos que teñan matrícula a tempo parcial a asistencia as clases non é obligatoria, pero sí a asistencia a proba no laboratorio na data establecida para a evaluación da parte práctica de laboratorio.

Fontes de información	
Bibliografía básica	- Charles Russell Severance (Autor), Fernando Tardio 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 Alvarez (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	- 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
Introdución aos Computadores /614G03012
Materias que continúan o temario
Programación II/614G03007
Observacións
O alumno debe ter en conta que debe realizar un labor autodidacta moi importante, seguindo o seguinte esquema: Ler, atender, comprender, preguntar, estudar e practicar:- Ler: Lea o tema para 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 se non pregunte.- Preguntar: Pregunte todo o que non comprenda, non quede con dúbidas.- Estudar: Estude despois das sesións, para reter o comprendido. - Practicar: Faga moitos programas, os que se lle pidan, suxiran, e outros pola súa conta, tanto en papel como no computador. Programación é unha materia 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.É unha materia que, por medio do sistema de avaliación continua, pódese aprobar sen máis que seguir, de forma activa, o ritmo das distintas sesións teóricas e prácticas. Debe facer caso ás indicacións particulares de reforzo de estudo que lle sinale o profesor.