Guia docente					
Datos Identificativos					2019/20
Asignatura (*)	Programación I			Código	614G01001
Titulación	Grao en Enxeñaría Informática				
Descriptores					
Ciclo	Periodo	Curso		Tipo	Créditos
Grado	1º cuatrimestre	Primero		Formación básica	6
Idioma	CastellanoInglés				
Modalidad docente	Presencial				
Prerrequisitos					
Departamento	Ciencias da Computación e Tecnoloxías da InformaciónComputación				
Coordinador/a	Rabuñal Dopico, Juan Ramon		Correo electrónico	juan.rabunal@udc.es	
Profesorado	Arcay Varela, Bernardino Boveda alvarez, Maria del Carmen Castro Martinez, Alfonso Cedrón Santaeufemia, Francisco Abel Martinez Perez, Maria Munteanu , Cristian Robert Rabuñal Dopico, Juan Ramon		Correo electrónico	bernardino.arcay@udc.es carmen.boveda@udc.es alfonso.castro@udc.es francisco.cedron@udc.es maria.martinez@udc.es c.munteanu@udc.es juan.rabunal@udc.es	
Web	moodle.udc.es/				
Descripción general	Esta materia es una introducción a la programación, en la que se ve cómo resolver problemas en un lenguaje estructurado. En ella se ayuda al alumno a comprender los tipos y estructuras de datos básicos, al mismo tiempo que se sientan las bases para diseñar correctamente un algoritmo. Y para asentar los conocimientos fundamentales de la programación de forma más rápida y óptima es necesario utilizar un lenguaje que permita la puesta en práctica de los conocimientos adquiridos y sirva de base para el buen desarrollo de un programador informático; se utilizará el Lenguaje de programación C, tanto para las prácticas como para los ejemplos teóricos.				

Competencias del título	
Código	Competencias del título
A4	Conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos, bases de datos y programas informáticos con aplicación en ingeniería.
A5	Conocimiento de la estructura, organización, funcionamiento e interconexión de los sistemas informáticos, los fundamentos de su programación, y su aplicación para la resolución de problemas propios de la ingeniería.
B1	Capacidad de resolución de problemas
B3	Capacidad de análisis y síntesis
B4	Capacidad para organizar y planificar
C3	Utilizar las herramientas básicas de las tecnologías de la información y las comunicaciones (TIC) necesarias para el ejercicio de su profesión y para el aprendizaje a lo largo de su vida.
C6	Valorar críticamente el conocimiento, la tecnología y la información disponible para resolver los problemas con los que deben enfrentarse.
C7	Asumir como profesional y ciudadano la importancia del aprendizaje a lo largo de la vida.

Resultados de aprendizaje	
Resultados de aprendizaje	Competencias del título

Conocer y comprender la importancia de los objetivos de la programación. Conocer los aspectos generales sobre los lenguajes y paradigmas de la programación. Conocer pseudocódigo y la sintaxis del lenguaje C utilizado para describir algoritmos y programas. Conocer los pasos para la realización de un programa y sus principales componentes. Conocer los tipos de datos básicos usando el Lenguaje C. Conocer las estructuras de control de la programación estructurada y las diferencias entre ellas. Conocer todos los aspectos relacionados con la realización de funciones y procedimientos.	A4 A5	B1 B3 B4	
Conocer y comprender la importancia de los objetivos de la programación. Conocer los aspectos generales sobre los lenguajes y paradigmas de la programación. Conocer pseudocódigo y la sintaxis del Lenguaje C utilizado para describir algoritmos y programas. Conocer los pasos para la realización de un programa y sus principales componentes. Conocer los tipos de datos básicos usando Lenguaje C . Conocer las estructuras de control de la programación estructurada y las diferencias entre ellas. Conocer todos los aspectos relacionados con la realización de funciones y procedimientos.	A4 A5	B1 B3 B4	
Ser capaz de realizar el seguimiento de un algoritmo (en pseudocódigo) o programa (en Lenguaje C), 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. 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.	A4 A5	B1 B3 B4	C3 C6 C7
Ser capaz de realizar el seguimiento de un algoritmo (en pseudocódigo) o programa (en Lenguaje C), 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. 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.		B1 B3 B4	C3 C6 C7
Aprendizaje autónomo. Planificación de las actividades a desarrollar. Capacidad de abstracción. Toma de decisiónes. Capacidad de iniciativa y participación.		B3 B4	C3 C6 C7
Aprendizaje autónomo. Planificación de las actividades a desarrollar. Capacidad de abstracción. Toma de decisiónes. Capacidad de iniciativa y participación.		B3 B4	C3 C6 C7

Contenidos	
Tema	Subtema

1 CONCEPTOS BÁSICOS	1.1 Algoritmos 1.1.1 Representación de algoritmos 1.2 Programas 1.2.1 Tipos de programas 1.3 Lenguajes de programación 1.3.1 Una visión histórica 1.3.2 Clasificación de los lenguajes 1.3.3 Instrucciones más importantes 1.3.4 Propiedades de los lenguajes 1.4 Traductores y Compiladores 1.5 Estructura de un programa 1.6 Elementos de un programa 1.6.1 Símbolos predefinidos 1.6.2 Símbolos especiales 1.6.3 Identificadores 1.6.4 Etiquetas 1.6.5 Comentarios 1.6.6 Directivas 1.6.7 Constantes 1.6.8 Números 1.6.9 Cadenas de carácteres 1.6.10 Variables: Declaración e iniciación 1.6.11 Variables: Dirección de Memoria 1.7 Salida y Entrada 1.7.1 Sentencias de salida 1.7.2 Sentencias de entrada 1.8 Tipos de datos y operadores 1.8.1 Tipos de datos 1.8.2 Operadores 1.8.3 Expresiones

2 SENTENCIAS DE CONTROL	2.1 Secuencial 2.2 Alternativa 2.2.1 La sentencia condicional simple 2.2.2 La sentencia condicional múltiple 2.3 Repetitiva 2.3.1 Introducción 2.3.2 Variables asociadas a los bucles 2.3.3 Funcionamento de los diferentes tipos de bucles 2.3.4 Bucle FOR 2.3.5 Equivalencia entre bucles 2.3.6 Errores en los bucles 2.3.7 Diseño de bucles
3 ARQUITECTURA DE UN PROGRAMA	3.1 Procedimientos 3.1 Funciones i Procedimientos 3.1.1 Tipos de funciones y procedimientos 3.1.2 Parámetros por valor y referencia 3.1.3 Parámetros protegidos 3.1.4 La pila de activación de procedimientos y funciones 3.1.5 Variables globales y locales: Alcance 3.1.6 Efectos laterales 3.2 Recursividad 3.2.1 Naturaleza de la recursividad 3.2.2 Recursión infinita
4 ESTRUCTURAS SIMPLES DE DATOS	4.1 Arrays y Matrices 4.1.1 Tipo de dato ARRAY 4.1.2 Declaración de un Array 4.1.3 Arrays de más de una dimensión 4.1.4 Operaciones con Arrays y Matrices 4.2 Registros 4.2.1 Tipo de dato registro 4.2.2 Operaciones con registros 4.3 Cadenas de caracteres 4.3.1 Cadenas de longitud fija 4.3.2 Cadenas de longitud variable 4.4 Operaciones básicas sobre Arrays 4.4.1 Búsqueda 4.4.2 Ordenación
5 ENTRADA / SALIDA	5.1 Ficheros 5.2 Tipos 5.3 Operaciones y modos de acceso 5.4 Funciones y procedimientos predefinidos específicos

Planificación

Metodologías / pruebas	Competéncias	Horas presenciales	Horas no presenciales / trabajo autónomo	Horas totales
Sesión magistral	A4 A5 B1 B3 C6 C7	30	30	60
Prácticas de laboratorio	A4 A5 B1 B3 B4 C3 C6 C7	20	50	70
Seminario	B4 C3 C6	8	10	18
Atención personalizada		2	0	2
(*)Los datos que aparecen en la tabla de planificación són de carácter orientativo, considerando la heterogeneidad de los alumnos				

Metodologías	
Metodologías	Descripción
Sesión magistral	En las sesiones de teoría, el profesor describe los objetivos y los contenidos de la materia, para dar una visión particular del tema a tratar y relacionarlo con otros dentro de la asignatura Después se desarrolla el tema correspondiente en la forma de sesión magistral, ayudándose de las herramientas técnicas disponibles, haciendo hincapié en ciertas cuestiones en las que el alumno debe profundizar en su autoaprendizaje. El objetivo es que el alumno aprenda a algoritmizar, utilizar las estructuras básicas de datos y resolver sencillos problemas de programación. Se utilizará como lenguaje de codificación C
Prácticas de laboratorio	En las sesiones de prácticas el alumno realizará programas en papel para después codificarlo en Lenguaje C, compilarlo, ejecutarlo y comprobar su nivel de corrección. Los enunciados de los programas se proporcionará con la suficiente antelación para que los alumnos puedan aprovechar mejor su tiempo. Es misión del profesor supervisar el código generado por el alumno para resolver dudas, corregir malos estilos de programación y corregir errores, contando con que el profesor no es un compilador que busca errores.
Seminario	En las sesiones de seminario se realizarán ejercicios y prácticas con la finalidad de detectar en los alumnos lagunas de conocimiento en la materia impartida hasta ese momento, y dar las explicaciones y/o referencias necesarias para subsanarlas.

Atención personalizada	
Metodologías	Descripción
Prácticas de laboratorio Seminario Sesión magistral	Tanto en las sesiones magistrales como en los laboratorios de prácticas y en las sesiones de seminario se llevará una atención personalizada del alumno, en distintos niveles según sea el tipo de clase, detectando el nivel de asimilación y comprensión de los temas explicados y las prácticas requeridas a implantar. En las sesiones de seminario es donde se puede llegar más al alumno para conocer las lagunas que tiene, e indicarle el camino para cubrirlas. Los alumnos que tengan matrícula a tiempo parcial deben, al inicio del curso, hablar con el/los profesores encargados de su grupo.

Evaluación			
Metodologías	Competéncias	Descripción	**Calificación**

Prácticas de laboratorio	A4 A5 B1 B3 B4 C3 C6 C7	Durante las últimas semanas con prácticas del curso se realizará una prueba en el laboratorio usando ordenadores que tendrá un valor máximo de 3 puntos sobre la nota total del curso. Será necesario que el programa a realizar por el alumno en el laboratorio compile y ejecute de forma correcta y completa.	30
Sesión magistral	A4 A5 B1 B3 C6 C7	La nota de la asignatura será la suma de lo obtenido en la Evaluación Continua (durante las 15 semanas del periodo lectivo correspondiente a la asignatura) y lo obtenido en el Examen Final. La nota de EVALUACIÓN CONTINUA, valorada en 4 puntos, se divide en dos partes: 1.- A la mitad del curso se realizará una prueba escrita que valdrá 1 puntos. 2.- En las últimas semanas con prácticas del curso se realiza una prueba en el laboratorio utilizando ordenadores que valdrá un máximo de 3 puntos. El EXAMEN FINAL constará de tres ejercicios que el alumno tendrá que desarrollar en código C, y tendrá un valor de 6 puntos. El examen oficial, tanto en la primera(enero) como en la segunda oportunidad (julio) constará de varios ejercicios a desarrollar en código C. Dicho Examen Final tiene un valor máximo de 6 puntos, que se sumarán a lo obtenido en la Evaluación Continua.	70

Observaciones evaluación
La nota final vendrá dada por la nota obtenida por EVALUACIÓN CONTINUA e la obtenida en el EXAMEN FINAL. El Examen Final constará de varios problemas a codificar en el lenguaje de programación empleada en las sesiones prácticas

Fuentes de información	
Básica	- Kernighan, Brian W. Englewood Cliffs (1988). The C Programming Language. New Jersey. Prentice Hall - K.N. King (2008). C programming. A modern Approach. Second Edition.. - James L. Antonakos , Kenneth C. Mansfield (2004). Programación estructurada en C. Madrid. Prentice-Hall - Luis Joyanes Aguilar, Ignacio Zahonero Martínez (2005). Programación en C metodología, algoritmos y estructura de datos. Madrid. McGraw-Hill - José R. García-Bermejo Giner (2008). Programación estructurada en C. Pearson - Luis Joyanes Aguilar (2011). Fundamentos de programación : algoritmos, estructuras de datos y objetos. Madrid. McGraw-Hill
Complementária	- Gabriela Márquez, Sonia Osorio, Noemí Olvera (2011). Introducción a la Programación Estructurada en C. Pearson - Andrés Marzal, Isabel García (2017). Introducción a la Programación con C. Publicacions de la Universitat Jaume I. Servei de Comunicació i Publicacions - Luis Joyanes Aguilar (2002). Programación en C. libro de problemas. Madrid. McGraw-Hill

Recomendaciones
Asignaturas que se recomienda haber cursado previamente
Asignaturas que se recomienda cursar simultáneamente
Informática Básica/614G01002
Asignaturas que continúan el temario
Programación II/614G01006
Otros comentarios

<p> El alumno debe tener en cuenta que debe realizar una labor autodidacta muy importante, siguiendo el siguiente esquema: Leer, atender, comprender, preguntar, estudiar y practicar.

</p> Leer: Lea el tema a tratar antes de asistir a las sesiones teóricas. ¡ES MUY IMPORTANTE! Atender: Atienda en clase, no sólo esté de cuerpo presente. Comprender: Comprenda lo que se le dice en las sesiones de teoría, y si no
pregunte. Preguntar: Pregunte todo lo que no comprenda, no quede con dudas. Estudiar: Estudie después de las sesiones, para retener lo comprendido.
Practicar: Haga muchos programas, los que se le pidan, sugieran, y otros por su cuenta, tanto en papel como en el ordenador.

<p>Programación es una asignatura que no se puede aprender estudiando en dos días. El alumno debe ir madurando los conceptos, hacer sobre el papel y en la máquina muchos programas, aprendiendo también de los errores al realizarlos.

</p><p>Es una asignatura que, por medio del sistema de evaluación continua, se puede aprobar sin más que seguir, de forma activa, el ritmo de las distintas sesiones teóricas y prácticas. Debe hacer caso a las indicaciones particulares de refuerzo de estudio que le señale el profesor.
</p>