



Guía docente				
Datos Identificativos				2015/16
Asignatura (*)	INFORMÁTICA	Código	730G04004	
Titulación	Grao en enxeñaría en Tecnoloxías Industriais			
Descriptorios				
Ciclo	Periodo	Curso	Tipo	Créditos
Grado	1º cuatrimestre	Primero	Formación Básica	6
Idioma	Castellano			
Modalidad docente	Presencial			
Prerrequisitos				
Departamento	Computación			
Coordinador/a	Duro Fernandez, Richard Jose	Correo electrónico	richard.duro@udc.es	
Profesorado	Becerra Permuy, Jose Antonio Bellas Bouza, Francisco Javier Duro Fernandez, Richard Jose Monroy Camafreita, Juan	Correo electrónico	jose.antonio.becerra.permuy@udc.es francisco.bellas@udc.es richard.duro@udc.es juan.monroy@udc.es	
Web	https://moodle.udc.es			
Descripción general	La materia objeto de esta guía constituye la única disciplina de formación básica de ámbito puramente informático en la titulación. Por sus contenidos, y dado el marcado carácter instrumental de la materia y que la mayor parte de las materias de Ingeniería necesitan métodos de cálculo susceptibles de ser realizados con ayuda de ordenador, observamos que podrían ser prácticamente todas las materias a las que la Informática apoyara desde sus contenidos como herramienta indispensable para el ingeniero.			

Competencias del título	
Código	Competencias del título

Resultados de aprendizaje			
Resultados de aprendizaje	Competencias del título		
Conocer la estructura funcional de un computador y sus componentes principales.	A3	B4	C1 C4 C5
Comprender la representación de la información en el computador.	A3	B9	C1 C5
Adquirir conocimientos sobre la estructura y funciones de un sistema operativo	A3		C1 C5
Conocer los fundamentos de las redes de computadores y de Internet			C1 C5
Capacidad de resolver problemas mediante el computador, en este caso el desarrollo de algoritmos y/o programas	A3 A12	B2 B3 B6 B7	C1 C5

Contenidos	
Tema	Subtema
BLOQUE DIDÁCTICO I	-----



Tema 1: Representación de la información	1.1.- Medida de la información 1.2.- Sistemas de numeración usuales en Informática 1.2.1 Sistema de numeración binario 1.2.2 Códigos intermedios 1.3.- Representación de números enteros 1.3.1.- Representación binaria sin signo 1.3.2.- Representación binaria signo-magnitud 1.3.3.- Representación binaria en complemento a 1 1.3.4.- Representación binaria en complemento a 2 1.3.5.- Aritmética con enteros 1.3.6.- Representación decimal BCD 1.4.- Representación de caracteres 1.4.1.- Código ASCII 1.5.- Representación de información analógica 1.5.1.- Representación de sonido 1.5.2.- Representación de imágenes
Tema 2: Arquitectura de ordenadores	2.1.- Arquitectura Von Neumann y extensiones 2.2.- Memoria Principal 2.2.1.- Tipos de Memoria Principal 2.2.2.- Rendimiento CPU-Memoria 2.2.3.- Jerarquía de memorias: memoria caché 2.3.- CPU (Unidad Central de Proceso) 2.3.1.- La Unidad Aritmético-Lógica 2.3.2.- La Unidad de Control 2.3.3.- Registros de la CPU 2.4.- Sistemas de almacenamiento masivo 2.4.1.- Discos magnéticos 2.4.2.- Discos ópticos 2.4.3.- Memorias de estado sólido y USB 2.5.- Conexiones y puertos
Tema 3: Sistemas operativos	3.1.- Estructura y funciones de un sistema operativo 3.2.- Tipos de sistemas operativos 3.3.- Gestión de recursos 3.3.1.- Gestión de archivos y directorios 3.3.2.- Gestión del procesador 3.3.3.- Gestión de la memoria principal 3.3.4.- Gestión de entrada/salida 3.3.5.- Gestión de la seguridad
Tema 4: Redes de datos e Internet	4.1.- Sistemas y medios de transmisión 4.2.- Redes de comunicación y topologías de red 4.3.- Tipos de redes 4.4.- Protocolos de red 4.5.- Internet y la web
BLOQUE DIDÁCTICO II	-----



Tema 5: Introducción a la programación	<ul style="list-style-type: none">5.1.- La programación<ul style="list-style-type: none">5.1.1.- Fase de análisis5.1.2.- Fase de programación5.1.3.- Fase de codificación5.2.- Estructura de un programa<ul style="list-style-type: none">5.2.1.- Partes principales de un programa5.2.2.- Clasificación de las instrucciones5.2.3.- Elementos auxiliares de un programa5.3.- Descripción de programas<ul style="list-style-type: none">5.3.1.- Pseudocódigo5.3.2.- Organigramas de programa y sistema5.3.3.- Representación de las estructuras de control5.4.- Lenguajes de programación<ul style="list-style-type: none">5.4.1.- Lenguaje máquina5.4.2.- Lenguaje ensamblador5.4.3.- Lenguaje de alto nivel5.4.4.- Traductores de lenguaje
Tema 6: Introducción a un lenguaje de programación: lenguaje C	<ul style="list-style-type: none">6.1.- Descripción general6.2.- Estructura de un programa en C6.3.- Tipos de datos, operadores y expresiones6.4.- Declaración de variables y constantes6.5.- Entrada y salida estándar
Tema 7: Estructuras de control	<ul style="list-style-type: none">7.1.- Expresiones lógicas7.2.- Instrucciones selectivas7.3.- Instrucciones iterativas7.4.- Instrucciones de salto
Tema 8: Funciones	<ul style="list-style-type: none">8.1.- Definición, declaración y llamada de funciones8.2.- El ámbito de las variables8.3.- Paso de argumentos<ul style="list-style-type: none">8.3.1.- Concepto de apuntador o puntero8.3.2.- Operadores de dirección e indirección8.3.3.- Paso de argumentos por valor y por referencia
Tema 9: Tipos de datos estructurados	<ul style="list-style-type: none">9.1.- Vectores o arrays<ul style="list-style-type: none">9.1.1.- Definición y uso de un vector9.1.2.- Inicialización<ul style="list-style-type: none">9.1.2.- Reserva dinámica de memoria9.2.- Matrices multidimensionales<ul style="list-style-type: none">9.1.1.- Definición y uso de una matriz9.1.2.- Inicialización<ul style="list-style-type: none">9.1.2.- Reserva dinámica de memoria9.3.- Cadenas de caracteres9.4.- Estructuras
Tema 10: Ficheros	<ul style="list-style-type: none">10.1.- Declaración de ficheros10.3.- Apertura y cierre de ficheros10.4.- Lectura y escritura de datos10.5.- Acceso directo a los datos



Metodologías / pruebas	Competencias	Horas presenciales	Horas no presenciales / trabajo autónomo	Horas totales
Prueba mixta	A3 B2 B3 C1	4	0	4
Sesión magistral	A12 A3 B7 C4 C5	30	30	60
Prácticas de laboratorio	A3 B2 C1	20	20	40
Solución de problemas	A3 B2 C1	4	20	24
Seminario	B4 B6 B9 C1	10	10	20
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
Prueba mixta	Prueba de evaluación que se realizará al final de curso en las correspondientes convocatorias oficiales. Consistirá en una prueba escrita en la que habrá que responder la diferentes tipos de preguntas sobre el temario de teoría y resolver problemas prácticos de programación
Sesión magistral	Actividad presencial en el aula que sirve para establecer los conceptos fundamentales de la materia. Consiste en la exposición oral complementada con el uso de medios audiovisuales/multimedia y la introducción de algunas preguntas dirigidas a los estudiantes, con el fin de transmitir conocimientos y facilitar el aprendizaje.
Prácticas de laboratorio	Desarrollo de prácticas en el laboratorio de informática. Esta actividad consistirá en el estudio de casos y ejemplos además de la realización, por parte del alumnos, de ejercicios de programación
Solución de problemas	Consistirá en la realización por parte del alumno de diversos ejercicios de programación en lenguaje C de manera presencial en el aula a lo largo del cuatrimestre. Deberán ser entregados al finalizar la clase y serán evaluados mediante la corrección por parte del profesor
Seminario	Seminarios donde se explicarán los conceptos teóricos básicos sobre programación en lenguaje C y se realizarán clases de refuerzo en aquellos temas que cada grupo más necesite

Atención personalizada	
Metodologías	Descripción
Seminario Prácticas de laboratorio Solución de problemas	<p>Prácticas de laboratorio: la atención personalizada se realiza de forma activa durante las prácticas, ya que el profesor no plantea ejercicios y simplemente da tiempo para que los alumnos los resuelvan, sino que controla en todo momento que no existan alumnos que pierdan el hilo de la explicación o se queden estancados. De cara a mantener un nivel homogéneo en todo el grupo, es necesario que el profesor dedique más tiempo a aquellos alumnos que más lo necesiten.</p> <p>Solución de problemas: la atención personalizada se centrará en la corrección individualizada de las prácticas que se resuelvan a lo largo del curso, centrándose el profesor en destacar las virtudes y señalar los fallos de cada alumno de cara a lograr su máximo rendimiento y comprensión de la asignatura.</p> <p>Seminario: estos grupos se basan en la interacción entre el profesor y el alumno a la hora de comprender los fundamentos del lenguaje C, de modo que el aprendizaje pueda ser llevado directamente a la realización de los trabajos prácticos</p>

Evaluación			
Metodologías	Competencias	Descripción	Calificación
Prueba mixta	A3 B2 B3 C1	Prueba final de la materia. Esta prueba tendrá una parte de teoría y otra de práctica.	73
Prácticas de laboratorio	A3 B2 C1	Se valorará el trabajo semanal del alumno en las clases prácticas mediante la evaluación de los ejercicios de programación que se realicen en el aula	6



Solución de problemas	A3 B2 C1	Se valorarán los conceptos prácticos de programación mediante dos ejercicios de programación que se resolverán en el aula a lo largo del cuatrimestre de forma autónoma por parte del alumno	21
Otros			

Observaciones evaluación

La nota final de la asignatura se calculará de la siguiente forma:

$$\text{Nota Final (NF)} = 0,4 \cdot \text{Nota_Teoría} + 0,6 \cdot \text{Nota_Práctica}$$

siendo imprescindible que tanto la Nota de Teoría como la Nota de Prácticas sean mayores de 5 para aprobar la asignatura.

La Nota de Teoría se obtendrá en un examen que se realizará a final de curso.

La Nota de Práctica se calcula mediante la expresión:

$$\text{Nota_Práctica} = 0,55 \cdot \text{Nota_examen_p} + 0,1 \cdot \text{Prácticas_laboratorio} + 0,1 \cdot \text{Solución_problema_1} + 0,25 \cdot \text{Solución_problema_2}$$

Nota_examen_p es la nota obtenida en un examen que se realizará a final de curso y para el cual el alumno tiene 2 convocatorias, una en Enero y otra en Julio.

Prácticas_laboratorio es la nota obtenida tras la corrección por parte del profesor de prácticas de los ejercicios de programación realizados en las clases prácticas durante el curso (todos ellos o aquellos que el profesor considere más relevantes). Estos ejercicios deberán realizarse durante el tiempo asignado a las clases prácticas y entregarse al final de las mismas. Durante la realización de estos ejercicios, el alumno puede exponer dudas al profesor o consultar los materiales que estime oportuno. Por tanto, esta actividad evaluará el trabajo diario del alumno en las clases prácticas.

Solución_problemas es la nota de los 2 problemas de programación propuestos a lo largo del curso y que resuelven de manera presencial en el aula.

Estos ejercicios deberán realizarse durante el tiempo asignado a las clases prácticas y entregarse al final de las mismas. La principal diferencia con las prácticas de laboratorio comentadas anteriormente es que en estas 2 pruebas no está permitida la consulta de ningún material adicional, ni se cuenta con la ayuda del profesor.

Las Notas de Práctica y Teoría se calculan del mismo modo en las convocatorias de Enero y Julio. Por tanto, la Nota del Examen Práctico tiene un valor del 55% de la parte práctica en ambas convocatorias. Las notas obtenidas durante el curso en las Prácticas de Laboratorio y en la Solución del Problemas de programación se guardan para la convocatoria de Julio, no siendo posible repetirlas.

Fuentes de información

Básica	<ul style="list-style-type: none"> - Gottfried, B. (2005). Programación en C. McGraw-Hill - García, F., Carretero, J., Fernández, J., Calderón, A. (2002). El lenguaje de programación C. Diseño e implementación de programas. Prentice Hall - Joyanes, L., Zahonero, I. (2005). Programación en C. Metodología, algoritmos y estructuras de datos. McGraw-Hill - Prieto, A., Lloris, A., Torres, J. C. (2006). Introducción a la Informática. McGraw-Hill - de Miguel Anasagasti, P. (2004). Fundamentos de los Computadores. International Thomson Learning Paraninfo
Complementaria	<ul style="list-style-type: none"> - Stallings, W. (2000). Organización y Arquitectura de Computadores. Prentice Hall - Tanenbaum, A. S. (2000). Organización de computadoras: Un enfoque estructurado. Pearson Educación - Joyanes Aguilar, L., Castillo Sanz, A., Sánchez García, L., Zahonero Martínez, I. (2002). Programación en C. Libro de problemas. McGraw-Hill - Joyanes Aguilar, L., Castillo Sanz, A., Sánchez García, L., Zahonero Martínez, I. (2005). C. Algoritmos, programación y estructuras de datos. McGraw-Hill

Recomendaciones

Asignaturas que se recomienda haber cursado previamente

DISEÑO Y ANÁLISIS ASISTIDO POR ORDENADOR/730G03033

FIABILIDAD ESTADÍSTICA Y MÉTODOS NUMÉRICOS/730G03046

Asignaturas que se recomienda cursar simultáneamente

Asignaturas que continúan el temario



Otros comentarios

Es muy recomendable el aprovechamiento de las clases prácticas de programación que se realizarán a lo largo del curso. Asimismo, a través de la página web de la materia irán exponiéndose diversos ejercicios para fomentar la capacidad de resolución problemas que será exigida al alumno. Es muy recomendable la realización de estos ejercicios para un buen aprovechamiento de la materia.

(*) La Guía Docente es el documento donde se visualiza la propuesta académica de la UDC. Este documento es público y no se puede modificar, salvo cosas excepcionales bajo la revisión del órgano competente de acuerdo a la normativa vigente que establece el proceso de elaboración de guías