



Universidad
Politécnica
de Cartagena

Campus
de Excelencia
Internacional



Guía docente

INFORMÁTICA

Curso 2019-20



GRADO EN INGENIERÍA EN ORGANIZACIÓN INDUSTRIAL (BOE 21-12-2012)

CENTRO UNIVERSITARIO DE LA DEFENSA

Universidad Politécnica de Cartagena



1. Descripción general

Nombre	INFORMÁTICA
Código	511101005
Carácter	Básica
ECTS	6
Unidad temporal	Cuatrimestral
Despliegue temporal	Curso 1º - Segundo cuatrimestre
Menciones / especialidades	
Idioma en la que se imparte	Castellano
Modalidad de impartición	Presencial



2. Datos del profesorado

Nombre y apellidos	Skorin-Kapov, Nina
Área de conocimiento	Ingeniería Telemática
Departamento	Ingeniería y Técnicas Aplicadas (CUD)
Teléfono	968189923
Correo electrónico	nina.skorinkapov@ cud.upct.es
Horario de atención y ubicación durante las tutorías	Despacho nº 24. Martes y Jueves 12:50-14:35 Como criterio general, el alumno que desee realizar una tutoría deberá previamente (al menos con un día de antelación) enviar un e-mail al profesor solicitando una cita previa con el fin de poder organizar debidamente la atención de todo el alumnado
Titulación	Licenciatura en telecomunicaciones por la Universidad de Zagreb, Croacia; Doctora por la Universidad de Zagreb, Croacia (homologado por la UPCT) Acreditación ANECA Profesor Titular de Universidad
Categoría profesional	Profesor/a Contratado/a Doctor/a de Facultades y Escuelas Superiores
Nº de quinquenios	3
Nº de sexenios	2
Currículum vitae	

Nombre y apellidos	Pereñiguez García, Fernando
Área de conocimiento	Ingeniería Telemática
Departamento	Ciencias e Informática (CUD)
Teléfono	968189946
Correo electrónico	fernando.pereniguez@ cud.upct.es
Horario de atención y ubicación durante las tutorías	Despacho nº 38. Martes y Jueves 12:35-14:35 Como criterio general, el alumno que desee realizar una tutoría deberá previamente (al menos con un día de antelación) enviar un e-mail al profesor solicitando una cita previa con el fin de poder organizar debidamente la atención de todo el alumnado.
Titulación	Ingeniero en Informática; Máster en Tecnologías de la Información y Telemáticas Avanzadas; Doctor Ingeniero Informático. Acreditación ANECA Profesor Titular de Universidad, Área Ingeniería Telemática
Categoría profesional	Profesor/a Contratado/a Doctor/a de Facultades y Escuelas Superiores



Nº de quinquenios
Nº de sexenios
Currículum vitae

Nombre y apellidos	Martínez Inglés, María Teresa
Área de conocimiento	Teoría de la Señal y Comunicaciones
Departamento	Ingeniería y Técnicas Aplicadas (CUD)
Teléfono	968189916
Correo electrónico	mteresa.martinez@ cud.upct.es
Horario de atención y ubicación durante las tutorías	Despacho nº 2. Martes y Jueves 12:50-14:35 Como criterio general, el alumno que desee realizar una tutoría deberá previamente (al menos con un día de antelación) enviar un e-mail al profesor solicitando una cita previa con el fin de poder organizar debidamente la atención de todo el alumnado
Titulación	Doctora Ingeniera de Telecomunicación. Área de Tecnologías de la Información y Comunicaciones. Acreditación Profesor Contratado Doctor.
Categoría profesional	Profesor/a Contratado/a Doctor/a de Facultades y Escuelas Superiores
Nº de quinquenios	
Nº de sexenios	
Currículum vitae	



3. Competencias y resultados del aprendizaje

3.1. Competencias básicas del plan de estudios asociadas a la asignatura

[CB1]. Que los estudiantes hayan demostrado poseer y comprender conocimientos en un área de estudio que parte de la base de la educación secundaria general, y se suele encontrar a un nivel que, si bien se apoya en libros de texto avanzados, incluye también algunos aspectos que implican conocimientos procedentes de la vanguardia de su campo de estudio

3.2. Competencias generales del plan de estudios asociadas a la asignatura

[CG2]. Aplicar las tecnologías generales y las materias fundamentales en el ámbito industrial para la resolución de problemas propios de la ingeniería.

3.3. Competencias específicas del plan de estudios asociadas a la asignatura

[CE3]. Usar y programar los ordenadores, sistemas operativos, bases de datos y programas informáticos con aplicación en ingeniería.

Competencias específicas de la asignatura (para aquellas asignaturas optativas que las tengan)

3.4. Competencias transversales del plan de estudios asociadas a la asignatura

[CT7]. Diseñar y emprender proyectos innovadores.

3.5. Resultados del aprendizaje de la asignatura

Al finalizar con éxito esta asignatura, el alumno deberá ser capaz de:

Conocer los principios básicos de arquitectura de computadores, sistemas operativos, y los tipos de lenguajes de programación.

Aplicar los mecanismos básicos de construcción de algoritmos en el paradigma de la programación estructurada para el diseño de programas.

Conocer los mecanismos básicos del lenguaje de programación C para representar tipos de datos (tanto primitivos como compuestos).

Desarrollar programas utilizando estructuras de control de la programación estructurada, y transformar a dicho lenguaje los diseños realizados.

Desarrollar programas utilizando funciones y los mecanismos de paso de parámetros.

Conocer las características básicas de una base de datos y los programas de computador más comunes en el ámbito de la ingeniería de organización.

Las actividades de enseñanza/aprendizaje diseñadas permitirán al alumno desarrollar su capacidad de: trabajo en equipo, análisis y síntesis de información, expresión escrita y comunicación oral mediante el desarrollo de un programa de ordenador y su defensa oral al finalizar el cuatrimestre.



4. Contenidos

4.1 Contenidos del plan de estudios asociados a la asignatura

Programación estructurada de aplicaciones informáticas. Lenguajes de programación. Edición y compilación de programas. Estructura y funciones de un sistema operativo. Tipos de sistemas operativos. Administración básica de sistemas operativos. Bases de Datos relacionales. Modelos de Datos. Herramientas de gestión de bases de datos. Componentes de un sistema informático. Categorías de aplicaciones informáticas. Recursos utilizados en un sistema informático. Aplicaciones informáticas habituales en ámbito ingenieril.

4.2. Programa de teoría

Unidades didácticas

Temas

UD 1. Sistemas Informáticos y Arquitectura de Computadores

Tema 1. Introducción a los sistemas informáticos: estructura básica de un computador y sistemas operativos.
Tema 2. Introducción a la programación estructurada y la algoritmia.

UD 2. La Programación Estructurada en C

Tema 3. Representación de la información.
Tema 4. Tipos de datos primitivos.
Tema 5. Operaciones de entrada/salida.
Tema 6. Estructuras de control: selección y repetición
Tema 7. Abstracción funcional. Paso de parámetros por valor.
Tema 8. Tipos de datos estructurados I: Arrays numéricos
Tema 9. Tipos de datos estructurados II: Cadenas de caracteres.
Tema 10. Tipos de datos estructurados III: Registros.

UD 3. Bases de Datos y Aplicaciones Informáticas

Tema 11. Introducción a las bases de datos.
Tema 12. Introducción a las aplicaciones informáticas en la ingeniería de organización.



4.3. Programa de prácticas

Nombre	Descripción
Práctica 1. Introducción al entorno de programación. Conceptos básicos para entrada / salida por consola.	En esta práctica introductoria se familiariza al alumno con el entorno de programación y se explican las operaciones básicas de entrada/salida de datos.
Práctica 2. Estructuras de control condicionales.	Esta práctica presenta las principales estructuras de control condicionales: if, if-else, switch.
Práctica 3. Estructuras de repetición (I).	Esta práctica consiste de diversos ejercicios que pretenden afianzar el uso de las estructuras de repetición: for, while, do-while.
Práctica 4. Estructuras de repetición (II).	En esta práctica de ampliación, además de proceder con ejercicios de mayor dificultad que la anterior práctica, se introducen las sentencias break y continue.
Práctica 5. Funciones. Paso de parámetros por valor.	En esta práctica el alumno afianza los conceptos relacionados con el uso de funciones haciendo uso del método básico de paso de parámetros por valor.
Práctica 6. Arrays numéricos.	En esta práctica se introduce al alumno con el manejo de arrays, que son un tipo básico de datos agregado que permiten almacenar colecciones de datos del mismo tipo.
Práctica 7. Cadenas de caracteres.	Para complementar la formación en el uso de arrays, esta práctica versa sobre la manipulación de arrays que almacenan caracteres, también conocidos como cadenas de caracteres.
Práctica 8. Registros.	En esta práctica, el alumno completará su formación en el uso de los tipos de datos estructurados aprendiendo el uso de los registros, también conocidos como estructuras.
Práctica 9. Apoyo a la resolución del Trabajo Final.	En esta última práctica se realizará una presentación del Trabajo Final de la asignatura y se proporcionarán orientaciones que sirvan de ayuda al alumno para su resolución.
Observaciones	



Se desarrollan 9 sesiones de aula de informática (de 2 horas de duración cada una) donde los alumnos además de familiarizarse con el uso de un computador y conocer las herramientas habituales para el desarrollo de programas, serán capaces de resolver y probar con la asistencia del profesor de prácticas pequeños ejemplos guiados. Estos ejemplos tienen la complejidad suficiente para poder ser seguidos sin dificultad al tiempo que refuercen los conocimientos adquiridos en las clases de teoría/problemas.

Prevención de riesgos

La Universidad Politécnica de Cartagena considera como uno de sus principios básicos y objetivos fundamentales la promoción de la mejora continua de las condiciones de trabajo y estudio de toda la Comunidad Universitaria.

Este compromiso con la prevención y las responsabilidades que se derivan atañe a todos los niveles que integran la Universidad: órganos de gobierno, equipo de dirección, personal docente e investigador, personal de administración y servicios y estudiantes.

El Servicio de Prevención de Riesgos Laborales de la UPCT ha elaborado un "Manual de acogida al estudiante en materia de prevención de riesgos" que puedes encontrar en el Aula Virtual, y en el que encontraras instrucciones y recomendaciones acerca de cómo actuar de forma correcta, desde el punto de vista de la prevención (seguridad, ergonomía, etc.), cuando desarrolles cualquier tipo de actividad en la Universidad. También encontrarás recomendaciones sobre cómo proceder en caso de emergencia o que se produzca algún incidente.

En especial, cuando realices prácticas docentes en laboratorios, talleres o trabajo de campo, debes seguir todas las instrucciones del profesorado, que es la persona responsable de tu seguridad y salud durante su realización. Consúltale todas las dudas que te surjan y no pongas en riesgo tu seguridad ni la de tus compañeros.

4.4. Programa de teoría en inglés

Unidades didácticas

Temas

PART I. Computer Systems & Architecture.

UNIT 1. Introduction to computer systems: Basic structure of a computer and operating systems.
UNIT 2. Introduction to structured programming and algorithms.

PART II. Structured Programming in C.

UNIT 3. Information representation.
UNIT 4. Primitive data types.
UNIT 5. Input/output operations
UNIT 6. Control structures: conditional and loops.
UNIT 7. Functional abstraction. Passing parameters by value.
UNIT 8. Structured data types I: Numeric arrays.
UNIT 9. Structured data types II: Strings.
UNIT 10. Structured data types II:



4.4. Programa de teoría en inglés

Unidades didácticas

Temas

Registers.

PART III. Databases & Software applications

UNIT 11. Introduction to databases.
UNIT 12. Introduction to software applications in industrial management engineering.

4.5. Observaciones

A continuación se detalla la relación de objetivos de aprendizaje para cada unidad didáctica.

UD 1. Sistemas Informáticos y Arquitectura de Computadores.

- Conocer la evolución tecnológica de los sistemas informáticos.
- Describir los niveles de abstracción de todo sistema informático.
- Exponer la arquitectura básica de von Neumann.
- Describir las características principales y la secuencia de funcionamiento de la unidad central de procesos.
- Conocer las diferentes unidades de entrada/salida de un sistema informático.
- Enumerar los buses de un sistema informático y principales tipos de periféricos.
- Conocer las funciones principales de sistema operativo y su estructura.
- Explicar los principios básicos y las fases del desarrollo de programas.
- Explicar los mecanismos básicos para la construcción de algoritmos en el paradigma de la programación estructurada.
- Construir algoritmos a partir de enunciados de problemas.

UD 2. La Programación Estructurada en C.

- Conocer los distintos mecanismos de representación de la información en un computador digital.
- Enumerar las principales bases numéricas que se utilizan en informática
- Realizar la conversión entre distintas bases.
- Ser capaz de codificar números enteros empleando diferentes formatos.
- Definir las características de los tipos de datos.
- Conocer los tipos de datos primitivos que ofrece el lenguaje C.
- Ser capaz de declarar y usar variables y constantes.
- Aplicar los operadores aritméticos y lógicos y las reglas de precedencia de los operadores.
- Resolver distintas expresiones aritméticas y lógicas en C.
- Describir el funcionamiento de las estructuras de selección if, if else, y switch y solucionar con ellas problemas sencillos.
- Distinguir las diferentes formas de repetición que aparecen en los algoritmos estudiados con anterioridad.
- Explicar el funcionamiento de las estructuras de repetición while y do-while, describir las diferencias entre ellas y resolver algoritmos sencillos.
- Explicar el funcionamiento de la estructura de repetición for, ser capaces de configurar las expresiones que usa y resolver con ella algoritmos sencillos.
- Interpretar la necesidad de alterar en ocasiones la repetición en curso y explicar el



funcionamiento de las instrucciones break y continue.

- Utilizar las estructuras de control combinándolas en apilamiento y anidamiento, para resolver algoritmos complejos.
- Conocer los mecanismos que ofrece el lenguaje C para definir estructuras estáticas de datos (arrays).
- Aplicar la sintaxis del lenguaje C para crear arrays.
- Explicar el procedimiento básico para recorrer arrays y acceder o modificar su contenido.
- Ser capaces de crear de arrays multidimensionales en el lenguaje C.
- Explicar los mecanismos básicos de manipulación de datos en arrays.
- Explicar el concepto de módulo y su correspondencia con la definición de una función en la programación estructurada.
- Explicar las partes que constituyen una función y la declaración de funciones en C.
- Dar ejemplos demostrativos de declaración, invocación e implementación de funciones.
- Aplicar la sintaxis del lenguaje C para crear registros básicos y anidados.
- Explicar el procedimiento para acceder a los elementos de un registro.
- Saber combinar el manejo de registros y arrays.

UD 3. Bases de Datos y Aplicaciones Informáticas.

- Explicar las características básicas de una base de datos.
- Exponer las ventajas del uso de un enfoque de bases de datos.
- Conocer los modelos de datos más empleados.
- Familiarizarse con las principales herramientas informáticas en el ámbito de la ingeniería de organización industrial.



5. Actividades formativas

Denominación	Descripción	Horas	Presencialidad
Clases teóricas en el aula		25	100
Preparación Trabajos/Informes		12	0
Preparación Trabajos/Informes en grupo		25	25
Realización de actividades de evaluación formativas y sumativas		2	100
Realización de exámenes oficiales		2	100
Clases de problemas en el aula		10	100
Sesiones Prácticas en Aula de Informática		20	100
Tutorías		2	50
Asistencia a Seminarios		2	100
Trabajo/Estudio Individual		50	0



6. Sistema de evaluación

6.1. Sistema de evaluación		
Denominación	Descripción y criterios de evaluación	Ponderación
Pruebas escritas oficiales	Para más información, ver Observaciones.	90 %
Actividades de evaluación formativas y sumativas, para la evaluación del desempeño de competencias: - Evaluación por el profesor, Autoevaluación y Coevaluación (evaluación por compañeros) mediante criterios de calidad desarrollados (rúbricas) de informes de laboratorio, problemas propuestos, actividades de Aprendizaje Cooperativo, etc. - Tablas de observación (check-list, escalas, rúbricas) para evaluar ejecuciones. - Portafolio y/o diario del alumno para evaluar la capacidad de autorreflexión y la dedicación. - Realización de tareas auténticas: simulaciones, estudio de casos y/o problemas aplicados reales, etc.	Para más información, ver Observaciones.	10 %



6.2. Evaluación formativa

Descripción

Prueba escrita parcial PEI_B1 (20%).
Problemas y ejercicios de complejidad similar a los resueltos en la bibliografía y en clase. La prueba escrita versará sobre los Temas 1-4

Examen Final Parte PEI_B1 (20%).
Problemas y ejercicios de complejidad similar a los resueltos en la bibliografía y en clase. La prueba escrita versará sobre los Temas 1-4

Examen Final Parte PEI_B2 (70%).
Problemas y ejercicios de complejidad similar a los resueltos en la bibliografía y en clase. La prueba escrita versará sobre los Temas 5-12

Trabajo Final (TF) de asignatura (10%).
Se propondrá un trabajo final, donde cada alumno deberá aplicar los distintos conocimientos adquiridos de programación en C.

Información

Tal como prevé el artículo 5.4 del Reglamento de las pruebas de evaluación de los títulos oficiales de grado y de máster con atribuciones profesionales de la UPCT, el estudiante en el que se den las circunstancias especiales recogidas en el Reglamento, y previa solicitud justificada al Departamento y admitida por este, tendrá derecho a una prueba global de evaluación. Esto no le exime de realizar los trabajos obligatorios que estén recogidos en la guía docente de la asignatura.

Observaciones

En relación a la realización de Pruebas Escritas Oficiales (PEI_B1 y PEI_B2):

- Habrá una prueba escrita individual (PEI_B1) parcial que versará sobre los Temas 1-4. Debe superarse con nota total igual o superior a 4 sobre 10 para eliminar esta parte de la asignatura de cara al examen final.
- La evaluación final constará de dos Pruebas Escritas Individuales (PEIs); en primer lugar, todos los alumnos realizarán la PEI correspondiente los temas 5-12 (PEI_B2). Tras un descanso, se realizará la PEI correspondiente a los Temas 1-4 (PEI_B1) para aquellos alumnos que no liberaron dicha materia en la PEI_B1 parcial, o aquellos que deseen mejorar su calificación. Todos los alumnos se pueden presentar a esta segunda PEI, teniendo en cuenta que, aquellos que liberaron materia en la prueba parcial, al entregar esta parte al profesor para su evaluación, perderán la calificación obtenida en la PEI parcial PEI_B1.
- Para aprobar la asignatura es necesario, pero no suficiente, obtener un mínimo de 4.0 en cada PEI. Si esta condición no se verifica, el alumno suspenderá la asignatura completa, siendo la nota máxima en esta situación 4.0 ¿ Suspense. Para aclarar este punto, supongamos un alumno que obtiene en PEI_B1 un 10.0, en PEI_2 un 3.0, y un 10 en el Trabajo Final (TF), la calificación media es 5.1 pero como no ha llegado al 4.0 en la PEI_B1 aparecerá en el acta de la asignatura con 4.0 ¿ Suspense. El trabajo final (TF) no tiene nota mínima.
- La calificación de la asignatura (N) se calculará según la siguiente expresión: $N=0,2^*$



$$(PEI_B1)+0,7*(PE_B2)+0,1*TF$$

- Para aprobar la asignatura es necesario obtener un mínimo de 4.0 en parte PEI_B1, un mínimo de 4.0 en parte PEI_B2, y en la calificación final de la asignatura N un mínimo de 5.0.
- Las notas obtenidas en cada PEI que superen 4.0 se mantendrán para la convocatoria de Agosto del presente curso académico.

En relación al Trabajo Final (TF) de la asignatura:

- La evaluación del Trabajo Final se realizará mediante entrevista personal dentro del periodo de exámenes de la convocatoria de Junio y utilizando una rúbrica
- El Trabajo Final de la asignatura será entregado en las fechas que indiquen los profesores dentro de la convocatoria de exámenes de Junio del presente curso.
- La puntuación del Trabajo Final se guardará hasta la convocatoria de Febrero del siguiente curso académico. Es decir, se mantendrá esa nota para la convocatoria de Agosto del presente curso académico y para la convocatoria de Febrero del siguiente curso académico.



7. Bibliografía y recursos

7.1. Bibliografía básica

Pedro M. Alcover Garau Informática Aplicada. Programación en C.. Universidad Politécnica de Cartagena. 2012. 978-84-693-9245-4

Pedro J. García Laencina Informática Aplicada. Prácticas para aprender a programar en Lenguaje C. Centro Universitario de la Defensa de San Javier. 2012. 978-84-939010-7-3

7.2. Bibliografía complementaria

Stallings, William. Sistemas operativos aspectos internos y principios de diseño. Pearson Prentice Hall,. 2005. 9788420544625

Stallings, William Organización y arquitectura de computadores diseño para optimizar prestaciones. Prentice-Hall. 2012. 9788489660823

Tanenbaum, Andrew S. Organización de computadoras un enfoque estructurado. Pearson Education. 2000. 9701703995

Antonakos, James L. Programación estructurada en C. Prentice Hall. 2004. 8489660239

Attaway, Stormy MATLAB: a practical introduction to programming and problem solving. Butterworth-Heinemann. 2009. 9780750687621

Celma Giménez, Matilde Bases de datos relacionales. Prentice Hall. 2003. 8420538507

Alberto Prieto y otros Introducción a la Informática. Mc. Graw Hill. 2006. 9788448146245

Fco. J. Ceballos Enciclopedia de Microsoft. Visual C#. Editorial RAMA (3ª Edición). 2010. 978-8478979868

7.3. Recursos en red y otros recursos

Todo el material de la asignatura está disponible en Aula Virtual