



Universidad  
Politécnica  
de Cartagena



## Guía docente de la asignatura

# Informática Aplicada

**Titulación: Grado en Ingeniería Química Industrial**

CSV:	bNqoMLNsdpT5gXmyqFMTm0K4k		Fecha:	16/01/2019 13:20:37	
Normativa:	Este documento es copia auténtica imprimible de un documento administrativo firmado electrónicamente y archivado por la Universidad Politécnica de Cartagena.				
Firmado Por:	Universidad Politécnica de Cartagena - Q8050013E				
Url Validación:	https://validador.upct.es/csv/bNqoMLNsdpT5gXmyqFMTm0K4k		Página:	1/15	

## 1. Datos de la asignatura

Nombre	Informática Aplicada				
Materia	Informática				
Módulo	Materias básicas				
Código	509101004				
Titulación	Grado en Ingeniería Química Industrial				
Plan de estudios	2010				
Centro	Escuela Técnica Superior de Ingeniería Industrial				
Tipo	Obligatoria				
Periodo lectivo	Cuatrimstral	Cuatrimstre	C1	Curso	1
Idioma	Español				
ECTS	6	Horas / ECTS	30	Carga total de trabajo (horas)	180

\* Todos los términos marcados con un asterisco que aparecen en este documento están definidos en *Referencias para la actividad docente en la UPCT y Glosario de términos*:

<http://repositorio.bib.upct.es/dspace/bitstream/10317/3330/1/isbn8469531360.pdf>

## 2. Datos del profesorado

<b>Profesor responsable</b>	Diego Alonso Cáceres		
<b>Departamento</b>	Tecnologías de la Información y las Comunicaciones		
<b>Área de conocimiento</b>	Lenguajes y Sistemas Informáticos		
<b>Ubicación del despacho</b>	2ª Planta Antiguo Cuartel de Antiguones, despacho 25		
<b>Teléfono</b>	968 325341	<b>Fax</b>	968 325973
<b>Correo electrónico</b>	diego.alonso@upct.es		
<b>URL / WEB</b>	Aula Virtual UPCT		
<b>Horario de atención / Tutorías</b>	Según establezca el profesor		
<b>Ubicación durante las tutorías</b>	Despacho del profesor		

<b>Titulación</b>	Ingeniero Industrial (Universidad Politécnica Valencia) Doctorado con Mención Europea (UPCT)
<b>Vinculación con la UPCT</b>	Permanente, Profesor Titular de Universidad
<b>Año de ingreso en la UPCT</b>	2004
<b>Nº de quinquenios (si procede)</b>	2
<b>Líneas de investigación (si procede)</b>	Ingeniería del Software para Sistemas Reactivos, Robótica, Desarrollo de Software Dirigido por Modelos, Desarrollo Basado en Componentes, Sistemas de Tiempo-Real, Programación de Agentes
<b>Nº de sexenios (si procede)</b>	2
<b>Experiencia profesional (si procede)</b>	
<b>Otros temas de interés</b>	

<b>Profesor (tardes)</b>	Jose Miguel Morales Illán		
<b>Departamento</b>	Tecnologías de la Información y las Comunicaciones		
<b>Área de conocimiento</b>	Lenguajes y Sistemas Informáticos		
<b>Ubicación del despacho</b>	2ª Planta Antiguo Cuartel de Antigones, despacho 28		
<b>Teléfono</b>	<u>968 326460</u>	<b>Fax</b>	<u>968 325973</u>
<b>Correo electrónico</b>	<u>josemiguel.morales@upct.es</u>		
<b>URL / WEB</b>	Aula Virtual UPCT		
<b>Horario de atención / Tutorías</b>	Lunes de 18:00 a 20:00 y martes de 16:00 a 18:00		
<b>Ubicación durante las tutorías</b>	Despacho número 28, edificio Antigones, zona departamental, Campus Muralla del Mar.		

<b>Titulación</b>	Dr. Informática
<b>Vinculación con la UPCT</b>	Profesor Asociado
<b>Año de ingreso en la UPCT</b>	2008
<b>Nº de quinquenios (si procede)</b>	--
<b>Líneas de investigación (si procede)</b>	Ingeniería del Software, Sistemas Reactivos, Robótica
<b>Nº de sexenios (si procede)</b>	--
<b>Experiencia profesional (si procede)</b>	18 años en empresas como programador y analista de sistemas
<b>Otros temas de interés</b>	

### 3. Descripción de la asignatura

#### 3.1. Descripción general de la asignatura

La formación del ingeniero industrial integra las habilidades propias de la ingeniería con los métodos de la matemática y la informática para formular y construir modelos para el diseño, análisis, evaluación y predicción de sistemas. Por ello, el alumno debe ser capaz de hacer uso de los recursos propios de un sistema informático en el ámbito de su especialidad ingenieril.

Con la asignatura *Informática Aplicada*, el alumno será capaz de hacer uso de lenguajes de programación con los que desarrollar aplicaciones de propósito general y dispondrá de conocimientos generales sobre bases de datos, modelos de datos y realización de consultas. Por otra parte, se proporciona al alumno conocimientos básicos sobre las partes que constituyen un sistema informático y sobre la estructura y funcionamiento de sistemas operativos. La formación aportada al alumno le capacita para poder extrapolar los conocimientos adquiridos a otras infraestructuras (lenguajes de programación, sistemas operativos, plataformas de desarrollo, etc.), de manera que las competencias adquiridas sean útiles en su desarrollo profesional a medio y largo plazo.

#### 3.2. Aportación de la asignatura al ejercicio profesional

La asignatura proporciona a los graduados y graduadas las competencias básicas necesarias para conocer las peculiaridades propias del desarrollo de programas de ordenador y su importancia en el ámbito de la Ingeniería Industrial, especialmente en aquellos aspectos relacionados con la Ingeniería Química.

#### 3.3. Relación con otras asignaturas del plan de estudios

Esta asignatura introduce conceptos de programación que el estudiante podrá aplicar en los programas utilizados en otras asignaturas del Grado, como Excel, Matlab, Maxima, etc.

#### 3.4. Incompatibilidades de la asignatura definidas en el plan de estudios

No existen

#### 3.5. Recomendaciones para cursar la asignatura

Es una asignatura eminentemente práctica, en la que se enseña cómo realizar programas de ordenador para resolver problemas sencillos. La programación implica el uso de Lógica, así que es conveniente que los alumnos repasen los principios de la misma. Se recomienda además resolver y probar los ejercicios con un ordenador y no solo en papel.

#### 3.6. Medidas especiales previstas

Se adoptarán las medidas especiales que especifique la E.T.S.I.I. en virtud de la normativa vigente y del estudio de cada caso particular. En cualquier caso, el alumno afectado debe contactar con el profesor al principio del curso.

## 4. Competencias y resultados del aprendizaje

### 4.1. 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.

### 4.2. Competencias generales del plan de estudios asociadas a la asignatura

CG3 - Conocimiento en materias básicas y tecnológicas, que les capacite para el aprendizaje de nuevos métodos y teorías, y les dote de versatilidad para adaptarse a nuevas situaciones.

### 4.3. Competencias específicas\* del plan de estudios asociadas a la asignatura

CE3 - Conocimientos fundamentales sobre el uso y programación de los ordenadores, sistemas operativos, bases de datos y programas informáticos con aplicación en Ingeniería.

### 4.4. Competencias transversales del plan de estudios asociadas a la asignatura

CT2 – Trabajar en equipo

### 4.5. Resultados\*\* del aprendizaje de la asignatura

Al finalizar con éxito esta asignatura, los estudiantes deben ser capaces de:

- (1) Describir los conceptos básicos de arquitecturas de ordenadores, sistemas operativos y bases de datos.
- (2) Diferenciar y emplear los distintos mecanismos de representación de datos en un ordenador.
- (3) Identificar los tipos de lenguajes de programación, así como los principios básicos y herramientas necesarias para el desarrollo de programas.
- (4) Aplicar la técnica de la programación estructurada en el diseño de algoritmos.
- (5) Desarrollar programas de ordenador a partir de los mecanismos de la *Programación Estructurada*.
- (6) Emplear los tipos de datos y estructuras de control ofrecidos por un lenguaje estructurado en el desarrollo de programas de ordenador.
- (7) Aplicar el desarrollo modular de programas en el diseño de aplicaciones de ordenador de tamaño medio.

**\*\* Véase también la *Guía de apoyo para la redacción, puesta en práctica y evaluación de los resultados del aprendizaje*, de ANECA:**

[http://www.aneca.es/content/download/12765/158329/file/learningoutcomes\\_v02.pdf](http://www.aneca.es/content/download/12765/158329/file/learningoutcomes_v02.pdf)

## 5. Contenidos

### 5.1. 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.

### 5.2. Programa de teoría (unidades didácticas y temas)

#### UD 1. Sistemas Informáticos, Arquitectura de Computadores y Sistemas Operativos

- Tema 1. Introducción a los sistemas informáticos y bases de datos, y sus aplicaciones en Ingeniería
- Tema 2. Estructura básica de un computador y representación de la información
- Tema 3. Conceptos básicos de sistemas operativos
- Tema 4. Conceptos básicos de bases de datos

#### UD 2. La Programación Imperativa y Estructurada en C

- Tema 5. Introducción a la programación estructurada y la algoritmia
- Tema 6. Tipos de datos primitivos y operaciones de entrada/salida
- Tema 7. Estructuras de control: selección y repetición
- Tema 8. Tipos estructurados de datos: vectores y estructuras
- Tema 9. Abstracción funcional. Paso de parámetros a funciones

### 5.3. Programa de prácticas (nombre y descripción de cada práctica)

#### Sesiones de Aula de Informática:


Se desarrollan diez sesiones de aula de informática donde los alumnos además de familiarizarse con el uso de un computador y conocer las herramientas habituales para el desarrollo de programas, sean 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 el desarrollo de las clases de pizarra. Las prácticas de laboratorio a desarrollar en la totalidad de sesiones serán las siguientes:

- Práctica 1. Introducción al entorno de programación (1 sesión)
- Práctica 2. Tipos de datos y entrada/salida básica en el lenguaje C (1 sesión)
- Práctica 3. Estructuras de selección del flujo de control en el lenguaje C (2 sesiones)
- Práctica 4. Estructuras de repetición del flujo de control en el lenguaje C (2 sesiones)
- Práctica 5. Estructuras estáticas de datos en el lenguaje C (2 sesiones)
- Práctica 6. Definición y uso de funciones en el lenguaje C (2 sesiones)

#### 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.

CSV:	bNqoMLNsdpT5gXmyqFMTm0K4k		Fecha:	16/01/2019 13:20:37	
Normativa:	Este documento es copia auténtica imprimible de un documento administrativo firmado electrónicamente y archivado por la Universidad Politécnica de Cartagena.				
Firmado Por:	Universidad Politécnica de Cartagena - Q8050013E				
Url Validación:	https://validador.upct.es/csv/bNqoMLNsdpT5gXmyqFMTm0K4k		Página:	7/15	

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.

## 5.4. Programa de teoría en inglés (unidades didácticas y temas)

### U 1. Computer Systems and Computer Architecture

- L1. Introduction to computer systems and relational databases, and their application in Engineering
- L2. Basic computer architecture and information representation
- L3. Operating systems concepts
- L4. Introduction to databases

### U 2. Structured Programming in C

- L5. Introduction to structured programming and algorithms
- L6. Primitive data types and input/output functions
- L7. Control structures: selection and iteration
- L8. Structured data types: arrays and structs
- L9. Functional abstraction. Parameter types

## 5.5. Objetivos del aprendizaje detallados por unidades didácticas

**Tema:** Introducción a los Sistemas Informáticos y bases de datos, y sus aplicaciones en Ingeniería.

- Objetivos de aprendizaje:**
- Definir los tipos y las características de los sistemas informáticos industriales.
  - Describir los niveles de abstracción que se pueden considerar en todo sistema informático.
  - Describir las características fundamentales de las bases de datos
  - Enumerar los principales usos de la Informática en la aplicación de la Ingeniería y las herramientas más habituales

**Tema:** Estructura Básica de un Computador y Representación de la Información.

- Objetivos de aprendizaje:**
- Exponer la arquitectura básica de los sistemas informáticos: la arquitectura de von Neumann.
  - Describir las características principales y la secuencia de funcionamiento de la unidad central de procesos.
  - Construir programas simples que utilicen la unidad central de procesos mediante un simulador de la misma.
  - Describir el funcionamiento de la memoria de un sistema informático y los tipos de memoria existentes (magnética, óptica y de estado sólido).
  - Exponer la clasificación de la memoria según la jerarquía de memoria de un PC.
  - Explicar el funcionamiento del subsistema de entrada/salida de un sistema informático.



- Enumerar los buses de un sistema informático y principales tipos de periféricos.
- Conocer los distintos mecanismos de representación de la información en un computador digital.
- Enumerar las principales bases numéricas que se utilizan en informática
- Computar la conversión entre distintas bases.
- Exponer las formas de codificar números enteros.
- Computar la conversión de un número entero entre las distintas formas de codificarlo.
- Describir las formas de codificar números reales.
- Enumerar las formas de codificar caracteres.

**Tema: Conceptos básicos de Sistemas Operativos**

**Objetivos de aprendizaje:**

- Describir los tipos de programas que se pueden ejecutar en un sistema informático.
- Definir los programas encargados de controlar el funcionamiento de un sistema informático: sistemas operativos.
- Exponer la historia de los sistemas operativos.
- Enumerar los tipos de sistemas operativos que existen en la actualidad y sus características más sobresalientes.
- Exponer la secuencia de arranque de un sistema informático y las funciones principales de un sistema operativo.
- Describir la gestión del tiempo de ejecución de los programas de un sistema informático que realiza un sistema operativo.
- Relacionar la gestión de la memoria del sistema por parte del sistema operativo con la jerarquía de memoria de un sistema informático.
- Exponer la gestión del almacenamiento que realiza un sistema operativo.

**Tema:  
Objetivos  
de  
aprendizaje:**

**Introducción a la Programación Estructurada y la Algoritmia**

- Explicar los principios básicos del desarrollo de programas.
- Explicar las fases de construcción de un programa.
- Enumerar los principales paradigmas de programación existentes.
- Distinguir entre lenguajes de programación de alto y bajo nivel de abstracción y entre lenguajes compilados e interpretados.
- Conocer las herramientas necesarias para el desarrollo de programas.
- Explicar los principios de la programación imperativa y estructurada.
- Explicar las ventajas de la programación estructurada.
- Explicar los mecanismos básicos para la construcción de algoritmos en el paradigma de la programación estructurada.
- Dar ejemplos de algoritmos sencillos resueltos.

**Tema: Tipos de Datos Primitivos y Operaciones de Entrada/Salida**

**Objetivos de aprendizaje:**

- Definir las características de los tipos de datos.
- Explicar los tipos de datos primitivos que ofrece el lenguaje C.
- Dar ejemplos del uso de un tipo de datos.
- Seleccionar el tipo de datos más adecuado a las necesidades de un problema.

- Explicar la declaración y uso de variables y constantes.
- Explicar los operadores aritméticos y lógicos y las reglas de precedencia de los operadores.
- Explicar las conversiones de tipos explícitas e implícitas.
- Dar ejemplos de la declaración y uso de variables y constantes en el lenguaje C.
- Resolver distintas expresiones aritméticas y lógicas en el lenguaje C.

**Tema: Estructuras de Control: Selección y Repetición**

- Objetivos de aprendizaje:**
- Enumerar las siete estructuras de control disponibles en el Lenguaje C.
  - Distinguir entre la selección única, la selección con alternativa y la selección múltiple.
  - Describir el funcionamiento de las estructuras de selección if, if/else 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 capaz 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. Modificar la programación de las estructuras de repetición para evitar el uso de ambas instrucciones.
  - Utilizar las estructuras de control combinándolas en apilamiento y anidamiento, para resolver algoritmos complejos.

**Tema: Tipos estructurados de datos: vectores y estructuras**

- Objetivos de aprendizaje:**
- Explicar las características de una estructura estática de datos y las posibilidades que ofrece.
  - Explicar los mecanismos que ofrece el lenguaje C para definir Estructuras Estáticas de datos.
  - Conocer la sintaxis del lenguaje C para definir vectores.
  - Explicar el procedimiento básico para recorrer vectores y acceder o modificar su contenido.
  - Explicar la necesidad y la forma de definir vectores multidimensionales en el lenguaje C.
  - Explicar los mecanismos básicos de búsqueda de datos en vectores uni y multidimensionales.
  - Conocer la sintaxis de C para definir y utilizar estructuras.
  - Saber declarar y utilizar estructuras en programas de tamaño pequeño.

**Tema: Abstracción Funcional. Paso de Parámetros**

- Objetivos de aprendizaje:**
- Exponer la necesidad de modularizar los programas de tamaño medio.
  - 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.

- Explicar el mecanismo de llamada a procedimiento síncrono.
- Clasificar los distintos métodos de pasos de parámetros (por valor y por referencia).
- Dar ejemplos demostrativos de declaración, invocación e implementación de funciones.
- Explicar los mecanismos de definición de funciones recursivas.
- Distinguir las situaciones en las que una definición recursiva es adecuada.

## 6. Metodología docente

### 6.1. Metodología docente\*

Actividad*	Técnicas docentes	Trabajo del estudiante	Horas
Clase de teoría	Clase expositiva utilizando técnicas de aprendizaje cooperativo informal de corta duración. Resolución de dudas de estudiantes. Orientación a los alumnos de los recursos documentales y multimedia disponibles.	Presencial: Toma de apuntes. Planteamiento de dudas.	25
		No presencial: Estudio de la materia.	38
Clase de problemas	Se resolverán problemas tipo y se analizarán casos prácticos. Se enfatizará el trabajo en plantear métodos de resolución y no en los resultados. Se plantearán problemas similares para que los alumnos lo vayan resolviendo, siendo guiados paso a paso por el profesor.	Presencial: Toma de apuntes y revisión con el compañero. Planteamiento de dudas.	15
		No presencial: Resolución de problemas y ejercicios como entregables semanales.	30
Clase de Prácticas. Sesiones de aula de informática	Las sesiones prácticas en el aula de informática son fundamentales para enlazar los contenidos teóricos y prácticos de forma directa. Mediante las sesiones de aula de informática se pretende que los alumnos apliquen los conocimientos de programación.	Presencial: Manejo del entorno de programación y resolución de problemas.	20
		No presencial: Lectura de la práctica con antelación al desarrollo de la misma. Resolución en casa de los ejercicios no completados en el aula.	10
Tutorías	Las tutorías serán individuales o de grupo con objeto de realizar un seguimiento del aprendizaje. Revisión de exámenes individual y por grupos y motivación por el aprendizaje.	Presencial: resolución de dudas concretas de los alumnos en horario oficial de tutorías	6
Realización de la práctica final de programación	Se explicará el enunciado de la práctica final, opcional, y se darán algunos consejos relativos a su programación.	No presencial: desarrollo de la práctica final de programación y preparación de informe.	18
Actividades de evaluación sumativa	Se realizarán dos pruebas escritas de carácter individual: una parcial, y otra final, que permiten comprobar el grado de consecución de las competencias específicas.	Presencial: asistencia a la prueba escrita y realización de la misma.	3
		No presencial: realización de tests	2
Realización de exámenes oficiales	Se realizará una prueba final escrita de tipo individual. Esta prueba se realiza al final del cuatrimestre y permite comprobar el grado de consecución de las competencias específicas.	Presencial: Asistencia a la prueba escrita y realización de la misma.	3
Otras actividades	El profesor planteará en clase la visualización de vídeos o la lectura de noticias relevantes para la asignatura en cuanto a avances tecnológicos. También se considera la instalación y configuración del software que se va a utilizar.	No presencial: Realizar las tareas planteadas por el profesor.	10
			180

## 6.2. Resultados (4.5) / actividades formativas (6.1)

Resultados del aprendizaje (4.5)							
Actividades formativas (6.1)	1	2	3	4	5	6	7
Clase de teoría	X	X	X	X		X	X
Clase de prácticas			X	X	X	X	X
Prácticas en aula de informática		X		X	X	X	X
Realización de la práctica final de programación				X	X	X	X

## 7. Metodología de evaluación

### 7.1. Metodología de evaluación\*

Actividad	Tipo		Sistema y criterios de evaluación*	Peso (%)	Resultados (4.5) evaluados
	Sumativa*	Formativa*			
(PE1) Prueba escrita individual parcial	X	X	3-4 preguntas de la UD1. Opcional. Para la convocatoria de <u>evaluación continua</u> .	20%	1-3
(PE2) Práctica final de programación	X	X	Ejercicio de programación. Opcional. No presencial. No recuperable. La calificación se guarda todo el curso académico.	+10%	4-7
(PE3) Realización de tests	X	X	Tests de evaluación. Opcional. Para la convocatoria de <u>evaluación continua</u> .	15%	2, 4, 5, 6, 7
(PEI) Prueba escrita individual	X		- Evaluación continua: 2-3 ejercicios de programación de UD2: 6,5 puntos. - Recuperación: 2-3 preguntas de UD1 (hasta 2,5 puntos) y 2-3 ejercicios de UD2 (mínimo 6,5 puntos)	65 o 100%	1-7
(PE4) Asistencia y realización de prácticas de informática		X	Mecanismos para evaluación del aprovechamiento en prácticas. No recuperable. Obligatoria su realización para aprobar la asignatura.	-	2, 4, 5, 6, 7

- Para aprobar la asignatura es necesario tener el APTO en prácticas (PE4) y sacar una nota mínima de un 4 sobre 10 en la prueba (PEI) en cualquiera de las tres convocatorias.
- Para estar APTO en prácticas (PE4) hay que estarlo en ASISTENCIA y en APROVECHAMIENTO. Dicha condición se guarda permanentemente.
  - El APTO-ASISTENCIA se consigue asistiendo a las 10 sesiones, salvo las faltas justificadas por tema de salud o familiar importante, que si el profesor lo estima necesario podrá recuperar yendo a alguno de los otros grupos de prácticas. Las faltas no justificadas se pueden recuperar asistiendo a los otros grupos de prácticas, avisando siempre al profesor responsable del grupo de prácticas asignado y al del grupo de prácticas al que se va a asistir.
  - El APTO-APROVECHAMIENTO se consigue en la última sesión de prácticas, siempre que el alumno tenga el APTO-ASISTENCIA. Se plantearán para ello extensiones sobre los problemas de prácticas que el estudiante resolverá en el laboratorio y subirá la solución a Moodle.
  - Ambos APTOS se pueden adquirir el día del examen oficial en cualquiera de las convocatorias del curso tras el examen (PEI), superando o bien la propuesta de extensión de un problema de prácticas (si el alumno ya tiene el APTO-ASISTENCIA), o bien un ejercicio de programación global (si no lo tenía).

- La primera convocatoria es de evaluación continua, por lo que tiene en cuenta los resultados de las pruebas de evaluación (PE1), (PE3) y (PEI).
- La segunda y tercera convocatoria son de recuperación, y no se tendrán en cuenta las notas de las pruebas (PE1) y (PE3).
- La práctica de programación (PE2) aporta hasta 1 punto extra en la nota de la asignatura, siempre que el alumno la haya aprobado (es decir, tenga una nota final igual o superior a 5 puntos y esté APTO en prácticas). Será además condición necesaria para obtener la calificación “Matrícula de Honor”.

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.

## 7.2. Mecanismos de control y seguimiento (opcional)

La frecuencia y naturaleza de las preguntas realizadas por el alumnado, así como la intensidad de los correos electrónicos, la asistencia en los horarios establecidos para las tutorías y la participación en el aula virtual, son instrumentos medibles para determinar el avance formativo del estudiante, siempre que éste las utilice.

La prueba parcial que se realiza en clase permite detectar posibles lagunas formativas al principio del proceso de aprendizaje. De manera más específica, el seguimiento se realizará mediante las siguientes actividades:

- Los resultados de las pruebas de evaluación sumativas y formativas.
- La actitud de los alumnos en las prácticas.
- Estadísticas de uso del material colocado en el Aula Virtual.
- Cuestiones planteadas en clase puntualmente.
- El tiempo de resolución de los problemas planteados en el aula de informática.

## 8 Bibliografía y recursos

### 8.1. Bibliografía básica\*

- *Introducción a la Informática*. Alberto Prieto y otros. Mc. Graw Hill, 4ª edición, 2006. (Unidad Didáctica 1).
- *Fundamentos de informática. Programación en C*. Pedro M. Alcover Garau. Edita la Universidad Politécnica de Cartagena, 2015. (Unidad Didáctica 2).
- *Informática aplicada: prácticas para aprender a programar en lenguaje C*. Pedro J. García Laencina, Pedro M. Alcover Garau. Edita la Universidad Politécnica de Cartagena, 2014. (Unidad Didáctica 2).

### 8.2. Bibliografía complementaria

Muy recomendados:

- *El lenguaje de programación C*, B. Kernighan y D. Ritchie, Editorial Prentice Hall, 1991. (Unidad Didáctica 2).
- *Programación estructurada en C*, J. Antonakos y K. Mansfield, Editorial Prentice Hall, 2004. (Unidad Didáctica 2).

Para ampliar conocimientos:

- *Organización de computadoras: un enfoque estructurado*, A. Tanenbaum, Editorial Pearson Education, 2000. (Unidad Didáctica 1).
- *Organización y arquitectura de computadores*, W. Stallings, Editorial Prentice-Hall, 2006. (Unidad Didáctica 1).
- *Sistemas operativos: aspectos internos y principios de diseño*, W. Stallings, Editorial Pearson Prentice Hall, 2005. (Unidad Didáctica 1).
- *Bases de datos relacionales*, Matilde Celma Giménez, Editorial Prentice Hall, 2003. (Unidad Didáctica 1).

### 8.3. Recursos en red y otros recursos

Aula Virtual de la UPCT

Entorno de compilación Codeblocks: <http://www.codeblocks.org/downloads/26>

Compilador online: <http://rextester.com/runcode>

Herramienta online “compiler explorer”: <https://godbolt.org/>

Tutoriales y compiladores online de múltiples lenguajes de programación y desarrollo web:

<http://www.tutorialspoint.com/codingground.htm>

## Informática Aplicada: Planificación Temporal

		Semana															Otros	Periodo exámenes	TOTAL HORAS POR ACTIVIDAD	
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15				
ACTIVIDADES PRESENCIALES	No convencional	TEMA/ACTIVIDAD																		
	Convencionales	PARCIAL																		
		Clases teoría	3	3	3	3	3	2	2	1	1	1	1	1	1			25		
		Clases problemas						1	1	2	2	1	1	1	2	2	2	15		
		Aula de informatica						2	2	2	2	2	2	2	2	2	2	20		
		TOTAL CONVENCIONALES	3	3	3	3	3	5	5	5	5	4	4	4	5	4	4	60		
		Parcial								1								1		
		Realización tests			0,4		0,4		0,4		0,4		0,4		0,5		0,5	3		
		PEI																4		
		TOTAL NO CONVENCIONALES			0,4		0,4		0,4	1,0	0,4		0,4		0,5		0,5	4	8	
ACTIVIDADES NO PRESENCIALES		Estudio		1	1	1	1	1	1	1	2	2	2	2	3	3	3	16	40	
		Problemas casa/semanales		1	1	2	3	3	3	3	2	2	2	2	2	2	2		30	
		Preparación prácticas						2	2	2	2	2	2	2	2	2	2		20	
		Preparación ejercicio programación													2	3	3	14	22	
		TOTAL NO PRESENCIALES	0	2	2	3	4	6	6	6	6	6	6	6	9	10	10	14	16	112
		TOTAL HORAS POR SEMANA	3	5	5	6	7	11	11	12	11	10	10	10	15	14	15	14	20	CARGA TOTAL
																			180	