

DATOS DE LA ASIGNATURA*

* Asignatura en experiencia piloto de implantación del sistema de créditos ECTS

Nombre:			
FUNDAMENTOS DE INFORMÁTICA			
Denominación en inglés¹:			
FUNDAMENTALS OF COMPUTER SCIENCE			
Código:	Año del Plan de Estudios:	Tipo:	
300099005	Publicación BOE: 20-05-1999	<input checked="" type="checkbox"/> Troncal <input type="checkbox"/> Obligatoria <input type="checkbox"/> Optativa	
Créditos:			
	Totales:	Teóricos:	Prácticos:
Créditos L.R.U.	6,00	3,00	3,00
Créditos E.C.T.S.	4,8	2,4	2,4
Departamento:			
Tecnologías de la Información			
Área de Conocimiento:			
Lenguaje y Sistemas Informáticos			
Curso:	Cuatrimestre:	Ciclo:	
Primero	2º Cuatrimestre	Primero	
Web de la asignatura:			
Aula Virtual en la plataforma Webct: http://www.uhu.es/sevirtual/#inicio y también en http://www.uhu.es/04004/			

¹Para su inclusión en el Complemento Europeo al Título

DATOS DE LOS PROFESORES

Nombre:	e-mail:	Teléfono:	Despacho:
Antonio Angel Márquez Hernández	amarquez@dti.uhu.es	959217388	52
Fco. Javier Fernández Bejarano	javier.fernandez@dti.uhu.es	959217648	29
María del Pilar Polo Almohano	polo@dti.uhu.es	959217386	56
Antonio J. Suárez Fábrega	asuarez@dti.uhu.es	959217677	39
Francisco Roche Beltrán	roche@dti.uhu.es	959217654	22
José Manuel Martín Ramos	jmartin@dti.uhu.es	959217637	46
Ana María Roldán Ruiz	amroldan@dti.uhu.es	959217387	54
Pedro Abad Herrera	abadhe@dti.uhu.es	959217678	41
Lourdes Ortiz Sañudo	lourdes@dti.uhu.es	959217391	47

DATOS ESPECÍFICOS DE LA ASIGNATURA

1.1. Descriptores de la asignatura:

Estructura de Computadores. Sistemas Operativos. Programación.

1.2. Descriptores de la asignatura (en inglés)²:

Computers structure. Programming. Operative Systems

²Para su inclusión en el Complemento Europeo al Título

2. Situación de la asignatura.

2.1. Prerrequisitos:

Los alumnos que cursen esta asignatura deberían :

- Saber leer, comprender y expresarse correctamente tanto de forma oral como escrita en castellano.
- Tener conocimientos matemáticos a nivel de Bachillerato.
- Conocer el idioma inglés a nivel escrito (nivel medio)

2.2. Contexto dentro de la titulación:

Por sus contenidos, de acuerdo con los descriptores del BOE, y dado el marcado carácter instrumental de la materia y que la mayor parte de las asignaturas de Ingeniería necesitan métodos de cálculo susceptibles de ser realizados con ayuda de ordenador, observamos que podrían ser todas prácticamente las materias a las que la materia troncal Fundamentos de Informática (en lo sucesivo FI) apoyara desde sus contenidos, como herramienta indispensable para el ingeniero.

2.3. Recomendaciones:

El alumno debe estudiar la asignatura consultando la bibliografía sugerida por los profesores y asistir con regularidad a las tutorías que estos ofertan. Debido al marcado carácter práctico de la asignatura, se recomienda que el alumnado realice la mayor parte de los ejercicios propuestos de las relaciones de problemas.

3. Competencias a adquirir por los estudiantes.

3.1. Competencias transversales o genéricas.

3.1.1. Competencias instrumentales:

<input checked="" type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Capacidad de análisis y síntesis.
<input type="checkbox"/> Alto	<input checked="" type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Capacidad de organización y planificación.
<input type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Comunicación oral y escrita en lengua nativa.
<input type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Conocimiento de una lengua extranjera.
<input checked="" type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Capacidad de gestión de la información.
<input checked="" type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Resolución de problemas.
<input type="checkbox"/> Alto	<input checked="" type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Toma de decisiones.
<input type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Conocimientos generales básicos.
<input type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Conocimientos básicos de la profesión.
<input type="checkbox"/> Alto	<input checked="" type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Conocimientos de informática.
<input type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Otras: Especificar.

3.1.2. Competencias personales:

<input type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Trabajo en equipo.
<input type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Trabajo en un equipo de carácter interdisciplinar.
<input type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Trabajo en un contexto internacional.
<input type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Habilidades en las relaciones interpersonales.
<input type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input checked="" type="checkbox"/> Bajo	Capacidad para comunicarse con expertos de otras áreas.
<input type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Reconocimiento a la diversidad y la multiculturalidad.
<input type="checkbox"/> Alto	<input checked="" type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Razonamiento crítico.
<input type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Compromiso ético.
<input type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Otras: Especificar.

3.1.3. Competencias sistémicas:

<input checked="" type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Aprendizaje autónomo.
<input checked="" type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Adaptación a nuevas situaciones.
<input checked="" type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Capacidad de aplicar los conocimientos en la práctica.
<input checked="" type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Habilidad para trabajar de forma autónoma.
<input checked="" type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Creatividad.
<input type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Liderazgo.
<input type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Conocimiento de otras culturas y costumbres.
<input type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Iniciativa y espíritu emprendedor.
<input type="checkbox"/> Alto	<input checked="" type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Motivación por la calidad.
<input type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Sensibilidad hacia temas medioambientales.
<input type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Diseño y gestión de proyectos.
<input type="checkbox"/> Alto	<input checked="" type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Motivación de logro.
<input type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Otras: Especificar.

3.2. Competencias específicas.

3.2.1. Competencias cognitivas (saber):

- Nuevas Tecnologías (E3).
- Conocimientos de Informática (E18).

3.2.2. Competencias procedimentales e instrumentales (saber hacer):

- Métodos de Diseño (E7): en la materia FI se hace especial hincapié en las metodologías de diseño de programas, que pueden en muchos casos, extrapolarse a otras materias.
- Planificación y Programación (E21): la realización de un algoritmo requiere una capacidad de organización y de planificación previa, así como una capacidad de abstracción, vital para su posterior desarrollo.

3.2.2. Competencias actitudinales (ser):

- Aprendizaje Autónomo (G15): los métodos de aprendizaje de la materia exigen la necesidad de incrementar la capacidad autónoma de razonamiento y análisis, proporcionando además la capacidad posterior al alumno para continuar el aprendizaje LLL (Long Life Learning), en ésta o en otras materias.
- Creatividad (G17): la resolución de nuevos problemas mediante la programación, requiere un esfuerzo creativo de diseño. La programación, como acto final de la resolución de un problema, es en sí, un acto creativo.

4. Objetivos:
Organizados por descriptor, los objetivos serían:
ESTRUCTURA DE COMPUTADORES:
<ul style="list-style-type: none"> Comprender los conceptos fundamentales sobre la estructura y organización interna de los computadores actuales, tanto a nivel físico como lógico, y familiarizarse con la terminología informática real, así como con los últimos desarrollos tecnológicos. Relacionado con las competencias: Conocimientos de Informática, Nuevas Tecnologías TIC.
SISTEMAS OPERATIVOS:
<ul style="list-style-type: none"> Conocer los conceptos fundamentales y los fundamentos básicos necesarios para la utilización de los sistemas operativos. Relacionado con las competencias: Conocimientos de Informática, Aprendizaje Autónomo, Nuevas Tecnologías TIC.
PROGRAMACIÓN:
<ul style="list-style-type: none"> Ser capaz de realizar programas de dificultad media/baja siguiendo una o varias metodologías de descripción de algoritmos, utilizando programación estructurada y siguiendo una metodología de diseño descendente. Ser capaz de traducir a un lenguaje de programación concreto (lenguaje C), los programas descritos mediante la metodología utilizada. Desarrollar en el alumnado la capacidad de resolución de problemas. Relacionados con las competencias: Conocimientos de Informática, Resolución de Problemas, Aprendizaje Autónomo, Creatividad, Nuevas Tecnologías TIC, Métodos de Diseño, Planificación y Organización.

5. Metodología (en horas de trabajo del estudiante):			
		Primer Cuatrimestre	Segundo Cuatrimestre
		Presenciales	
	Clases de teoría	0,0	21,0
	Clases de problemas	0,0	9,0
	Clases prácticas	0,0	26,0
	Actividades académicas dirigidas	0,0	4,0
		Exámenes	6,0
		No presenciales	
	Estudio de clases teóricas (factor de trabajo: 0,75)	0,0	15,7
	Estudio de clases de problemas y prácticas (factor de trabajo: 1,00)	0,0	35,0
	Preparación de actividades académicamente dirigidas y otras actividades	0,0	11,4
	Total:	0,0	128,1
Trabajo total del estudiante: 128,1 horas.			
Horas presenciales:	60,0	Horas no presenciales:	62,1
		Exámenes:	6,0

6. Técnicas docentes.	
6.1. Técnicas docentes utilizadas:	
<input checked="" type="checkbox"/>	Sesiones académicas de teoría
<input checked="" type="checkbox"/>	Sesiones académicas de problemas
<input checked="" type="checkbox"/>	Sesiones prácticas en laboratorio
<input checked="" type="checkbox"/>	Seminarios, exposiciones y debates
<input type="checkbox"/>	Trabajo en grupos reducidos
<input type="checkbox"/>	Resolución y entrega de problemas/prácticas
<input checked="" type="checkbox"/>	Realización de pruebas parciales evaluables
<input type="checkbox"/>	Otras: Especificar

6.2. Desarrollo y justificación:
<i>Sesiones Académicas de Teoría:</i>
<p>Consisten en clases magistrales donde se impartirá la base teórica de la asignatura y se expondrán ejemplos aclaratorios de la misma al grupo, que se supone compuesto de no más de 80 alumnos. Las sesiones serán de una hora y se irán intercalando con las sesiones de problemas a lo largo del curso, de manera que una vez finalizado un tema teórico con sus correspondientes sesiones académicas de teoría, se impartirán sesiones de problemas.</p> <p>La metodología usada para impartir la teoría y los ejemplos aclaratorios será la exposición mediante transparencias y uso de pizarra. El profesor solicitará la participación activa del alumno mediante preguntas rápidas.</p> <p>En la página web de la asignatura se encontrarán las transparencias y otros materiales de referencia necesarios para el seguimiento de las sesiones.</p>
<i>Seminarios:</i>
<p>Consiste en la descripción del concepto de ordenador, viendo cuáles son sus componentes y la importancia que cada uno de ellos tiene. Además se introduce el concepto de sistema operativo, así como la clasificación del resto del software. Finalmente se describe el entorno de compilación usado para el desarrollo práctico de la asignatura.</p> <p>Estos seminarios se impartirán en grupos reducidos de no más de 24 alumnos en los laboratorios de prácticas a fin de que los alumnos puedan conocer físicamente los componentes hardware de un PC y su funcionamiento.</p>
<i>Sesiones de Problemas:</i>
<p>Consisten en la realización de problemas relacionados con los conceptos y métodos operativos de la asignatura. Las sesiones serán de una hora. El profesor explicará uno o varios problemas tipo. En estas sesiones se fomentará la participación del alumnado en la resolución de los problemas planteados.</p> <p>En la página web de la asignatura se encontrará un compendio de problemas para resolver, y se irán indicando aquellos que van siendo resueltos en las sesiones. El resto de problemas debe de ser resueltos por los alumnos de forma voluntaria y las soluciones propuestas por ellos podrán ser comprobadas haciendo uso de las horas de tutorías y de atención al alumno del profesor.</p>
<i>Trabajos o actividades académicas dirigidas:</i>
<p>Se realizarán dos pruebas de progreso, a lo largo del curso, con el fin de llevar una evaluación continua, a la par que cada alumno pueda realimentarse de sus propios resultados midiendo así la evolución de su aprendizaje.</p> <p>Las dos pruebas consistirán en un problema práctico a resolver mediante la construcción de un programa en C en el laboratorio. Para la realización de los programas, el alumno únicamente podrá utilizar la ayuda proporcionada por el compilador, sin que tenga permitido el uso de ningún tipo de ayuda adicional tanto en formato digital (transparencias en powerpoint, pdf, códigos fuentes de otros programas, etc.) como en papel (apuntes, libros, programas resueltos, etc.)</p> <p>Estas pruebas se realizarán en grupos reducidos de no más de 24 alumnos en los laboratorios a fin de que cada alumno tenga a su disposición un PC en el que poder trabajar.</p>
<i>Prácticas de Laboratorio:</i>
<p>Consisten en el diseño e implementación de programas escritos en C. Los alumnos dispondrán con antelación la relación de problemas a resolver y la metodología de trabajo. Los grupos de prácticas serán de no más de 24 alumnos y el trabajo se realizará de forma individual. Las sesiones serán de dos horas.</p> <p>La asistencia a las sesiones de laboratorio es obligatoria. La participación activa de los alumnos en la resolución de problemas será valorada.</p>

7. Bloques temáticos:

I. INTRODUCCIÓN A LA INFORMÁTICA

1. CONCEPTO DE ORDENADOR Y SISTEMAS OPERATIVOS

II. FUNDAMENTOS DE PROGRAMACIÓN

1. CONCEPTOS BÁSICOS DE ALGORÍTMICA
2. EL PROCESO DE COMPILACIÓN, DEL CÓDIGO FUENTE AL CÓDIGO MÁQUINA
3. ELEMENTOS BÁSICOS DE PROGRAMACIÓN
4. CONTROL DE FLUJO DEL PROGRAMA
5. PUNTEROS
6. DISEÑO DESCENDENTE: SUBPROGRAMAS
7. ESTRUCTURAS DE DATOS COMPLEJAS

8. Temario desarrollado:

Bloque Temático 1: INTRODUCCIÓN A LA INFORMÁTICA

TEMA 1: CONCEPTO DE ORDENADOR Y SISTEMAS OPERATIVOS

- 1.1. Introducción
- 1.2. La Máquina de Von Neumann
- 1.3. Concepto de Ordenador
 - 1.3.1. Componentes de un Ordenador
 - 1.3.2. El Ordenador físicamente
- 1.4. Representación de la Información
- 1.5. Introducción al concepto de Sistema Operativo

Bloque Temático 2: FUNDAMENTOS DE PROGRAMACIÓN

TEMA 2: CONCEPTOS BÁSICOS DE ALGORÍTMICA

- 2.1. Definición de Algoritmo
 - 2.1.1. Propiedades de los Algoritmos
- 2.2. ¿Qué es un Programa?
 - 2.2.1. Cómo se construye un Programa
- 2.3. Definición y uso de herramientas para describir soluciones: Diagramas de flujo, Pseudocódigo
 - 2.3.1. El Diagrama de Flujo
 - 2.3.2. El pseudocódigo
- 2.4. Traducción de ideas a un lenguaje de programación concreto: El problema de la implementación
 - 2.4.1. Acciones y Estructuras de control
 - 2.4.2. Procedimientos
- 2.5. Lenguajes de programación. Clasificación. Colocación en la clasificación del lenguaje C
 - 2.5.1. Clasificación de los lenguajes de programación
 - 2.5.2. El lenguaje C

TEMA 3: EL PROCESO DE COMPILACIÓN, DEL CÓDIGO FUENTE AL CÓDIGO MÁQUINA

- 3.1. Fase de compilación o interpretación, y linkado (link, montado o enlace).
- 3.2. Fase de ejecución de un programa.
- 3.3. El compilador de C. Características generales.

TEMA 4: ELEMENTOS BÁSICOS DE PROGRAMACIÓN

- 4.1. Definiciones. Palabras reservadas básicas del lenguaje C.
- 4.2. Tipos, Variables y Constantes.
 - 4.2.1. Tipos de Datos Fundamentales.
 - 4.2.2. Variables y Constantes.
 - 4.2.3. Declaración de variables.
 - 4.2.4. Definición de constantes.
- 4.3. Operaciones básicas aritmético-lógicas. Operadores y expresiones
 - 4.3.1. Operación de asignación
 - 4.3.2. Operadores aritméticos
 - 4.3.3. Operadores relacionales

- 4.3.4. Operadores lógicos
- 4.3.5. Expresiones
- 4.4. Operaciones básicas de entrada-salida
- 4.4.1. E/S por consola
- 4.4.2. E/S por consola con formato

TEMA 5: CONTROL DE FLUJO DEL PROGRAMA

- 5.1. Sentencias
- 5.2. Sentencias compuestas (bloques)
- 5.3. Estructuras de selección
- 5.3.1. if else
- 5.3.2. switch
- 5.4. Estructuras de repetición
- 5.4.1. while
- 5.4.2. do while
- 5.4.3. for
- 5.4.4. El bucle infinito
- 5.4.5. Bucles vacíos

TEMA 6: PUNTEROS

- 6.1. Los operadores de punteros
- 6.2. Asignaciones de punteros
- 6.3. Inicialización de punteros
- 6.4. Problemas con punteros
- 6.4.1. Ejemplo de puntero que no apunta al tipo correcto
- 6.4.2. Ejemplo de puntero usado de forma incorrecta

TEMA 7: DISEÑO DESCENDENTE: SUBPROGRAMAS

- 7.1. Niveles de abstracción.
- 7.2. Solución de problemas utilizando técnicas de Diseño Descendente.
- 7.3. Estilo en la creación de un programa.
- 7.3.1. Programación Modular.
- 7.3.2. Programación Estructurada.
- 7.4. Subprogramas.
- 7.5. Parámetros y Variables Locales. Variables Globales.
- 7.5.1. Ámbito de las variables.
- 7.5.2. Variables locales.
- 7.5.3. Parámetros por valor y parámetros por referencia
- 7.5.4. Variables Globales
- 7.6. Prototipos de Funciones
- 7.7. Sección de Includes: Ficheros cabecera
- 7.8. Compilación y enlazado (link)

TEMA 8: ESTRUCTURAS DE DATOS COMPLEJAS

- 8.1. Vectores Estáticos
- 8.1.1. Vectores o Arrays unidimensionales
- 8.1.2. Vectores multidimensionales
- 8.1.3. Inicialización de arrays
- 8.2. Cadenas o vectores de cadenas
- 8.2.1. Inicialización de cadenas
- 8.3. Paso de cadenas y arrays a funciones
- 8.4. Estructuras
- 8.4.1. Arrays y estructuras
- 8.4.2. Paso de estructuras a funciones
- 8.4.3. Paso de elementos de estructuras a funciones
- 8.4.4. Punteros a estructuras
- 8.5. Typedef

9. Bibliografía.

9.1. Bibliografía general:

- Fco. Javier Fernández, Antonio Márquez, Pilar Polo. "Fundamentos de Informática: Introducción a la Programación en C". Servicio Publicaciones Universidad de Huelva. 2005.
- Delannoy, Claude. "El libro del C. Primer lenguaje." Ediciones Gestion 2000, SA
- Joyanes, L; Zahonero, I: "Programación en C: Metodología, estructura de datos y objetos". Ed. McGraw-Hill, 2001.
- Gottfried, Byron: "Programación en C". McGraw-Hill, 2005
- Schildt, Herbert. "Turbo C/C++. Manual de referencia". McGraw-Hill, 1992.

9.2. Bibliografía específica:

- Prieto, A; Lloris, A; Torres, J.C. Introducción a la Informática. Editorial McGraw-Hill, 3era Edición, 2002.
- L. Joyanes. "Fundamentos de Programación. Algoritmos y Estructuras de Datos". Segunda Edición. Mc Graw-Hill, 1996
- Miguel Anasagasti, P. *Fundamentos de los Computadores*, 6ª Ed, Paraninfo, 1998.

10. Técnicas de evaluación.

10.1. Técnicas de evaluación utilizadas:

- Examen teórico-práctico
- Trabajos desarrollados durante el curso
- Participación activa en las sesiones académicas
- Controles periódicos de adquisición de conocimientos
- Examen práctico en aula de informática
- Otras: Se realizarán dos pruebas de progreso, a lo largo del curso, con el fin de llevar una evaluación continua, a la par que cada alumno pueda realimentarse de sus propios resultados midiendo así la evolución de su aprendizaje. Estas pruebas se realizarán tras el desarrollo de los siguientes contenidos:
 - Prueba 1: "Arrays".
 - Prueba 2: "Diseño Descendente y Subprogramas".
 Con estas pruebas o actividades académicas dirigidas, se evalúa principalmente la parte práctica de la asignatura, dado que es el mecanismo más directo para conocer la asimilación de conceptos por parte del alumnado, conjuntamente con la valoración de la participación activa en la resolución de los problemas prácticos propuestos en las prácticas de laboratorio.

10.2. Criterios de evaluación y calificación:

Se resumen en esta tabla:

Aspecto	Criterio	Instrumento	Calificación
Conocimientos teóricos.	Exposición escrita de conocimientos y su aplicación a distintas situaciones.	Examen Teórico Escrito	7
Conocimientos prácticos.	Capacidad de implementar programas en C	Participación	0,50
		Actividad 1	1,25
		Actividad 2	1,25

La nota del alumno será:

Si (**Examen Teórico Escrito**) \geq 4 (sobre 10)
 (Examen Teórico Escrito*0,7) + Actividades (Conocimientos prácticos)

En caso contrario, si (**Examen Teórico Escrito**) $<$ 4 (sobre 10)
 no se aplica la fórmula anterior, no superando el alumno la asignatura. La nota en acta será la obtenida en el Examen Teórico Escrito.

Criterio para guardar las notas para convocatorias posteriores:

Examen Teórico:

- La nota del examen teórico se guarda hasta la convocatoria de diciembre siempre que se obtenga una nota igual o superior a 5 (sobre 10), no guardándose para convocatorias posteriores a la de diciembre.

Examen Práctico de Laboratorio:

- La nota de prácticas de laboratorio (hasta 3,0 puntos) se guarda hasta la convocatoria de diciembre inclusive, siempre y cuando el alumno obtenga 1 punto en prácticas, no guardándose para convocatorias posteriores a la de diciembre.

11. Organización docente semanal (en horas presenciales del alumno)

11.1. Primer cuatrimestre:

Semana	Horas de clases de teoría	Horas de clases de problemas	Horas de clases prácticas	Actividades Académicas Dirigidas		Horas de exámenes	Temas del temario a tratar
				Actividad	Horas		
1ª	0,0	0,0	0,0		0,0	0,0	
2ª	0,0	0,0	0,0		0,0	0,0	
3ª	0,0	0,0	0,0		0,0	0,0	
4ª	0,0	0,0	0,0		0,0	0,0	
5ª	0,0	0,0	0,0		0,0	0,0	
6ª	0,0	0,0	0,0		0,0	0,0	
7ª	0,0	0,0	0,0		0,0	0,0	
8ª	0,0	0,0	0,0		0,0	0,0	
9ª	0,0	0,0	0,0		0,0	0,0	
10ª	0,0	0,0	0,0		0,0	0,0	
11ª	0,0	0,0	0,0		0,0	0,0	
12ª	0,0	0,0	0,0		0,0	0,0	
13ª	0,0	0,0	0,0		0,0	0,0	
14ª	0,0	0,0	0,0		0,0	0,0	
15ª	0,0	0,0	0,0		0,0	0,0	
Periodo de exámenes						0,0	
Totales	0,0	0,0	0,0		0,0	0,0	

11.2. Segundo cuatrimestre:

Semana	Horas de clases de teoría	Horas de clases de problemas	Horas de clases prácticas	Actividades Académicas Dirigidas		Horas de exámenes	Temas del temario a tratar
				Actividad	Horas		
1ª	2,0	0,0	2,0		0,0	0,0	Presentación y Tema 2
2ª	1,0	1,0	2,0		0,0	0,0	Tema 2 y Tema 4
3ª	2,0	0,0	2,0		0,0	0,0	Tema4
4ª	1,0	1,0	2,0		0,0	0,0	Tema 4
5ª	2,0	0,0	2,0		0,0	0,0	Tema 5 (If – switch)
6ª	1,0	1,0	2,0		0,0	0,0	Tema 5 (Repetitivas)
7ª	1,0	1,0	2,0		0,0	0,0	Tema 5 (Repetitivas)
8ª	1,0	1,0	2,0		0,0	0,0	Tema 8 (Arrays)
9ª	1,0	1,0	2,0		0,0	0,0	Tema 8 y Tema 6
10ª	2,0	0,0	0,0	Prueba de progreso	2,0	0,0	Tema 7
11ª	1,0	1,0	2,0		0,0	0,0	Tema 7
12ª	2,0	0,0	2,0		0,0	0,0	Tema 7 y Tema 8
13ª	1,0	1,0	2,0		0,0	0,0	Tema 8
14ª	2,0	0,0	2,0		0,0	0,0	Tema 8
15ª	1,0	1,0	0,0	Prueba de progreso	2,0	0,0	Tema 8
Periodo de exámenes						6,0	
Totales	21,0	9,0	26,0		4,0	6,0	

12. Mecanismos de control y seguimiento:

- Se propone la realización de encuestas de satisfacción entre los alumnos.
- Índices de rendimiento (tasa de éxito/fracaso, % de presentados a examen, etc.)
- Control de Asistencia