



Universidad
de Huelva

ESCUELA POLITECNICA SUPERIOR

DATOS BÁSICOS DE LA ASIGNATURA					
NOMBRE:	FUNDAMENTOS DE INFORMÁTICA				
CÓDIGO:	05004	TIPO	TRONCAL		
AÑO DE PLAN DE ESTUDIOS:	1999				
CRÉDITOS:	Totales		Teóricos	Prácticos	
L.R.U.	6		3	3	
E.C.T.S.	5		2.5	2.5	
CURSO:	Primer curso	CUATRIMESTRE:	Segundo	CICLO:	1

DATOS BÁSICOS DE LOS PROFESORES					
NOMBRE:	Fco. Javier Fernández Bejarano (Teoría y Práctica)				
CENTRO/DEPARTAMENTO:	EPS La Rábida / DIESIA				
ÁREA:	Lenguajes y Sistemas Informáticos				
Nº DE DESPACHO:	29	TELÉFONO:	959 21 76 48		
E-MAIL:	javier.fernandez@diesia.uhu.es				
URL WEB:	http://www.uhu.es/javier.fernandez				
NOMBRE:	Antonio Angel Márquez Hernández (Práctica)				
CENTRO/DEPARTAMENTO:	EPS La Rábida / DIESIA				
ÁREA:	Lenguajes y Sistemas Informáticos				
Nº DE DESPACHO:	52	TELÉFONO:	959 21 73 88		
E-MAIL:	amarquez@uhu.es				
URL WEB:	http://www.uhu.es/antonio_marquez				
NOMBRE:	María del Pilar Polo Almohano (Práctica)				
CENTRO/DEPARTAMENTO:	EPS La Rábida / DIESIA				
ÁREA:	Lenguajes y Sistemas Informáticos				
Nº DE DESPACHO:	56	TELÉFONO:	959 21 73 86		
E-MAIL:	polo@diesia.uhu.es				
URL WEB:	http://www.uhu.es/mpilar.polo/				
NOMBRE:	Juan Luis Domínguez Olmedo (Práctica)				
CENTRO/DEPARTAMENTO:	EPS La Rábida / DIESIA				
ÁREA:	Lenguajes y Sistemas Informáticos				
Nº DE DESPACHO:	65	TELÉFONO:	959 21 73 71		
E-MAIL:	juan.dominguez@diesia.uhu.es				
URL WEB:					

DATOS ESPECÍFICOS DE LA ASIGNATURA

1. Descriptor según BOE

Estructura de Computadores.
Sistemas Operativos.
Programación.

2. Situación

2.1. Prerrequisitos

Los alumnos que cursen esta asignatura deberían :

- Saber leer, comprender y expresarse correctamente tanto de forma oral como escrita.
- Tener conocimientos matemáticos a nivel de Bachillerato.
- Conocer el idioma inglés tanto a nivel escrito como oral (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 que se desarrollan

3.1. Genéricas o transversales

- Conocimientos de Informática (G5).
- Resolución de Problemas (G7): la titulación de Ingeniería, sus atribuciones y competencias profesionales están enfocadas hacia la resolución de problemas del mundo real. La materia troncal FI proporciona la herramienta más poderosa para conseguir estos objetivos de formación.
- 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.

3.2. Específicas

Cognitivas(saber):

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

Procedimentales/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.

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:

- 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:

- 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:

- 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

5.1. Trabajo con presencia del profesor		Nº de horas
Clases teóricas		21
Clases prácticas		30
Exposiciones y seminarios		3
Tutorías especializadas	Colectivas	
	Individuales	
Realización de actividades académicas dirigidas:		
Trabajo sobre Estructuras de Control		2
Trabajo sobre Arrays y Funciones		2
Trabajo sobre Estructuras de Datos Complejas		2
Nº total de horas		60
5.2. Trabajo autónomo del alumno		Nº de horas
Estudio de las clases teóricas		31.5
Estudio de la clases prácticas		30
Preparación de las actividades académicas dirigidas		27,52
5.3. Realización de exámenes		Nº de horas
Realización de exámenes escritos		4.5
Realización de exámenes Prácticos		3
Nº total de horas		7.5
Trabajo total del estudiante		156,52

6. Técnicas docentes

- Sesiones académicas teóricas
- Exposición y debate
- Tutorías especializadas
- Sesiones académicas prácticas
- Controles de lecturas obligatorias

6.1. Desarrollo y justificación

Sesiones Académicas de Teoría:

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.

La metodología usada para impartir la teoría y los ejemplos aclaratorios será la exposición mediante transparencias y uso de pizarra. El profesor podrá solicitar la participación activa del alumno mediante preguntas rápidas, teniendo en cuenta los alumnos que más participen a la hora de evaluar.

En la página web de la asignatura se encontrarán las transparencias y otros materiales de referencia necesarios para el seguimiento de las sesiones.

Seminarios:

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

Sesiones de Problemas:

Consisten en la realización de problemas relacionados con los conceptos y métodos operativos de la asignatura. Los grupos serán de no más de 20 alumnos, siendo las sesiones de una hora. El profesor explicará uno o varios problemas tipo. En estas sesiones se valorará la participación del alumnado en la resolución de los problemas planteados.

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. La asistencia a las sesiones de problemas es obligatoria.

Trabajos o actividades académicas dirigidas:

Se realizarán tres 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.

Prácticas de Laboratorio:

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. La asistencia a las sesiones de laboratorio es obligatoria.

Tal como se expresa en los apartados anteriores el grupo para la clase de teoría sería de no más de 80 alumnos, los grupos de sesiones de problemas y seminarios serían de no más de 20 alumnos y los grupos de laboratorio serían de no más de 24 alumnos. Actualmente con el profesorado existente esta carga docente no es posible de cubrir, luego en caso de no paliarse tal situación, esta guía no se puede llevar a efecto.

7. Bloques temáticos

INTRODUCCIÓN A LA INFORMÁTICA

- La Máquina de Von Neumann
- Concepto de Ordenador
- Componentes de un Ordenador
- El Ordenador físicamente
- Representación de la Información
- Introducción al concepto de Sistema Operativo

FUNDAMENTOS DE PROGRAMACIÓN

- Conceptos Básicos de Algorítmica
- El Proceso de Compilación, del Código Fuente al Código Máquina
- Elementos Básicos de Programación
- Control de Flujo del Programa
- Punteros
- Diseño Descendente: Subprogramas
- Estructuras de Datos Complejas

8. Bibliografía

8.1. General

- 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.

8.2. Específica

- 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
- Schildt, Herbert. "Turbo C/C++. Manual de referencia". McGraw-Hill, 1992.

9. Técnicas de evaluación

- Examen Teórico Escrito. Consistente en una prueba donde el alumno deberá resolver distintos problemas y/o cuestiones teóricas referentes a los temas teóricos desarrollados durante el cuatrimestre.
- Trabajos o actividades académicas dirigidas. Se realizarán tres 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: “Control de Flujo del Programa”.
 - Prueba 2: “Diseño Descendente y Subprogramas”.
 - Prueba 3: “Estructura de Datos Complejas”.
- Con estas pruebas o actividades académicas dirigidas, se evalúa principalmente la parte práctica de la asignatura, dado que el mecanismo más directo para conocer la asimilación de conceptos por parte del alumnado.
- Sesiones de problemas. En todas las actividades formativas el profesor valorará la participación activa del alumno. Se valorará de forma directa tanto en las sesiones de problemas como en las del laboratorio.

9.1. Criterios de evaluación y clasificación

Se resumen en esta tabla:

Aspecto	Criterio	Instrumento	Peso
Conocimientos teóricos.	Exposición escrita de conocimientos y su aplicación a distintas situaciones.	Examen Teórico Escrito	60
Conocimientos prácticos.	Capacidad de implementar programas en C	Actividad 1	5
		Actividad 2	12,5
		Actividad 3	12,5
Asistencia y Participación.	Participación activa en las sesiones, aportando soluciones válidas a los problemas.	Sesiones de problemas	10

La nota del alumno será:

Si (**Examen Teórico Escrito + Sesiones de problemas**) \geq 40

(Examen Teórico Escrito + Sesiones de problemas) + Actividades (Conocimientos prácticos)

En caso contrario, si (**Examen Teórico Escrito + Sesiones de problemas**) $<$ 40

no se aplica la fórmula anterior, no superando el alumno la asignatura.

10. Organización docente semanal

(Sólo indicar el número de horas que a ese tipo de sesión va a dedicar el estudiante cada semana)

N° de horas 2° Cuatr.	Sesiones Teóricas	Sesiones Prácticas (problemas)	Sesiones Prácticas	Exposiciones y Seminarios	Tutorías Especializadas Colectivas	Otras actividades		Exámenes	Temas de teoría a tratar
						Horas	Actividad		
1ª Semana	1	0.0	0.0	2	0.0	0.0		0.0	Presentación y Tema 1
2ª Semana	2	0.0	0.0	1	0.0	0.0		0.0	Tema 2 y Tema 3
3ª Semana	2	0.0	3	0.0	0.0	0.0		0.0	Tema 3 y tema4
4ª Semana	2	0.0	2	0.0	0.0	0.0		0.0	Tema 4
5ª Semana	2	0.0	2	0.0	0.0	0.0		0.0	Tema 5 (If – switch)
6ª Semana	1	0.0	3	0.0	0.0	0.0		0.0	Tema 5 (Repetitivas)
7ª Semana	1	0.0	3	0.0	0.0	0.0		0.0	Tema 5 (Repetitivas)
8ª Semana	1	0.0	3	0.0	0.0	0.0		0.0	Tema 5 (Arrays)
9ª Semana	1	2	2	0.0	0.0	0.0		0.0	Tema 6
10ª Semana	1	0.0	2	0.0	0.0	0.0		0.0	Tema 7
11ª Semana	3	0.0	2	0.0	0.0	0.0		0.0	Tema 7
12ª Semana	1	0.0	2	0.0	0.0	0.0		0.0	Tema 7
13ª Semana	2	0.0	2	0.0	0.0	0.0		0.0	Tema 8
14ª Semana	1	2	2	0.0	0.0	0.0		0.0	Tema 8
15ª Semana	0	2	2	0.0	0.0	0.0		7.5	
	21	6	30	3	0	0.0		7.5	

Distribuya el número de horas que ha respondido en el punto 5 en 15 semanas para una asignatura cuatrimestral y 30 anuales
Las semanas en rojo son no lectivas y las grises son de exámenes.

11. Temario desarrollado

Bloque Temático 1: INTRODUCCIÓN A LA INFORMÁTICA

TEMA 1: CONCEPTO DE ORDENADOR

- 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: Procedimientos y Funciones.
- 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

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