

Plan Docente de una materia

“Elementos de Programación”

I. Descripción y contextualización

<i>Identificación y características de la materia</i>				
<i>Denominación</i>	Elementos de Programación			
<i>Curso y Titulación</i>	1º de Ingeniería Técnica en Informática de Sistemas			
<i>Área</i>	Lenguajes y Sistemas Informáticos			
<i>Departamento</i>	Informática			
<i>Tipo</i>	Troncal (6+3 ctos. LRU)		Nivel Medio (primer ciclo)	
<i>Coeficientes</i>	Practicidad: 3 (Medio-alto, profesional)		Agrupamiento: 1 (Bajo)	
<i>Duración ECTS (créditos)</i>	Anual		7.8 ECTS (195 h.)	
<i>Distribución ECTS (rangos)</i>	Grupo Grande:	Seminario-Lab.:	Tutoría ECTS:	No presenciales:
	21%	10%	3%	66%
	40 horas	20 horas	6 horas	129 horas
<i>Descriptor</i> <i>(según BOE)</i>	Diseño de algoritmos. Análisis de algoritmos. Lenguajes de programación. Diseño de programas. Técnicas de verificación y prueba de programas.			
<i>Coordinador-Profesor/es</i>	Valentín Masero Vargas			
<i>Tutorías complementarias (1)</i>	Despacho 21	Ext Teléf. 2574	vmasero@unex.es	

Contextualización profesional*

Conexión de la asignatura con los perfiles profesionales de la Titulación

Los tres grandes perfiles que se proponen responden a las tendencias profesionales y que pretenden abarcar las diferentes propuestas existentes dentro de la amplitud y diversidad de perfiles, son:

- Perfil profesional de Desarrollo Software
- Perfil profesional de Sistemas
- Perfil profesional de Gestión y Explotación de Tecnologías de la Información

Un Ingeniero Técnico en Informática con *perfil profesional de Desarrollo Software* debe estar preparado para participar y desarrollar cualquiera de las actividades implicadas en las fases del ciclo de vida de desarrollo de software, en productos software y aplicaciones de dimensión media. Es decir, es capaz de analizar, modelar las soluciones y gestionar los requisitos del producto. Es el perfil en el que tiene mayor importancia la asignatura Elementos de Programación, por sentar ésta las bases para el conocimiento necesario para el desarrollo de Software: la Programación.

El *perfil profesional de Sistemas* capacita a un profesional para analizar, diseñar, construir e implementar sistemas basados en computadoras, que soporten aplicaciones técnicas, comerciales, industriales, no convencionales y de negocios en general, utilizando técnicas y métodos que aseguren eficiencia. Administra centros de cómputo o de sistemas de información de datos, utiliza y orienta el empleo de software de aplicación e investiga en materias de tecnologías de información.

Un Ingeniero en Informática con *perfil profesional de Gestión y Explotación de Tecnologías de la Información* es responsable de asegurar que las necesidades de Gestión de la Información y del Conocimiento de las organizaciones se satisfacen con el desarrollo y la implantación de soluciones informáticas. Conoce la estrategia empresarial y las diferentes soluciones de las Tecnologías de la Información y de las Comunicaciones necesarias para apoyar dicha estrategia.

Resumiendo, el *grado de implicación* en cada uno de los perfiles de la asignatura:

En Elementos de programación en las titulaciones de I.T.I.S. e I.T.I.G. el estudiante entra en contacto con el mundo de la programación, la algoritmia y las estructuras de datos. Esta asignatura debe preparar a los alumnos para abordar con éxito problemas de pequeño y mediano tamaño, de tal forma que, aunque es una asignatura importante en los tres perfiles, tiene mayor implicación en los dos primeros, y sobre todo, en el primero, al sentar las bases del tema central del perfil, la programación de aplicaciones software.

- * Perfil de desarrollo software : grado alto
- * Perfil de sistemas: grado alto
- * Perfil de gestión y explotación de la TI: grado medio

Con respecto a los perfiles establecidos anteriormente se pueden establecer unas características que suelen requerir las entidades que dan empleo a nuestros titulados. Entre estas características, las siguientes tienen relación con esta asignatura (estas características están tomadas de distintas ofertas de trabajo):

- Perfil de Desarrollo de Software:

- o Dirigir el equipo de trabajo compuesto de Analistas Funcionales, Analistas de aplicaciones, Programadores.
- o Supervisar y coordinar el desarrollo completo de aplicaciones y administrar la introducción de los sistemas de gestión.
- o Asesorar a los programadores en los problemas que se les plantean con la programación de los sistemas.

- Perfil de Sistemas:

- o Estimación de volúmenes de las estructuras de datos, definiendo mecanismos de migración y carga inicial de datos.
- o En producción se ocupa de la gestión y operativa asociada a las bases de datos y el software en el que están implementadas.
- o Diseño de las soluciones informáticas relacionadas con los cambios en los sistemas existentes o con los Nuevos Sistemas.
- o Dirección y asesoramiento a los Programadores en la realización de los programas.
- o Creación de los test de pruebas para verificar que los Sistemas Informáticos cumplen los requisitos y especificaciones de Análisis y Diseño.
- o Asesoramiento a usuarios, Programadores y Jefe de Estudios en la redacción de la Documentación de Usuario, Instalación y Explotación.

- Perfil de Gestión y Explotación de las TI

- o Organizar y distribuir el trabajo de los equipos de análisis y de desarrollo (jefes de proyectos, responsables de aplicación).
- o Planificación del desarrollo de un Proyecto Informático.
- o Motivar y coordinar los equipos de desarrollo en el marco de aplicación de las normas y métodos en vigor.
- o Intermediario y consejero de cara a los desarrollos que se realicen.

Contextualización curricular**Conexión con las Competencias Genéricas y Específicas del Título*

En la actualidad se imparten en el Centro Universitario de Mérida dos titulaciones de Informática: Ingeniero Técnico en Informática de Sistemas e Ingeniero Técnico en Informática de Gestión. Los actuales planes de estudio entraron en vigor en el curso 1998-1999 (B.O.E. del 9 de febrero de 1999). En ambos casos, el plan se estructura en un primer ciclo de 3 años.

Los Reales Decretos 1459/1990, 1460/1990 y 1461/1990 (B.O.E. de 20 de noviembre de 1990) establecen la creación de los títulos universitarios oficiales y las directrices generales propias de los planes de estudio de Ingeniero en Informática, Ingeniero Técnico en Informática de Sistemas e Ingeniero Técnico en Informática de Gestión. En estos Reales Decretos se especifican las materias troncales de obligatoria inclusión en todos los planes de estudio de las diferentes universidades.

Existe un Libro Blanco sobre las Titulaciones Universitarias de Informática en el nuevo Espacio Europeo de Educación Superior, elaborado por representantes de 56 Universidades del Estado Español en el marco del Proyecto EICE (Estudios de Informática y Convergencia Europea). Dicho Libro Blanco marcará las líneas generales del contexto curricular de los próximos años.

En los Reales Decretos señalados anteriormente se concretan elementos tales como la denominación del correspondiente título oficial, los objetivos formativos de la enseñanza, la estructura en ciclos, la carga lectiva mínima y máxima en créditos de cada ciclo, las materias troncales y los descriptores, la distribución en módulos teóricos y prácticos de cada materia troncal, y la vinculación de éstos a una o más áreas de conocimiento. La tabla siguiente muestra la materia troncal que ha de ser incluida en los planes de estudio de la Universidad de Extremadura y que tiene que ver con las dos asignaturas que se estudian en esta sección.

Materia troncal	Créditos (LRU)	Contenido
Metodología y tecnología de la programación	ITIS 12	Diseño de algoritmos. Análisis de Algoritmos. Lenguajes de programación. Diseño de programas: descomposición modular y documentación. Técnicas de verificación y pruebas de programas.

Materia troncal de Ingeniería Técnica en Informática de Sistemas relacionada con la asignatura en estudio.

Estos descriptores son cubiertos por las asignaturas en las que se ha desglosado la materia en los planes de estudio conducentes a los títulos de Ingeniero Técnico en Informática de Sistemas e Ingeniero Técnico en Informática de Gestión del Centro Universitario de Mérida.

Las Competencias Específicas del Título con las que más se vincula la asignatura, para los perfiles (1) Desarrollo de Software, (2) Sistemas y (3) Gestión y Explotación de Tecnologías de la Información, son las siguientes:

- Bases de datos [Perfiles 1, 2 y 3]
- Capacidad para entender y evaluar especificaciones internas y externas [Perfiles 1, 2 y 3]
- Conocimiento de productos tecnológicos y tendencias de la tecnología, asociados al segmento del mercado [Perfiles 1, 2 y 3]
- Dirección, planificación y gestión de proyectos [Perfiles 1, 2 y 3]
- Diseño y arquitectura de sistemas de información [Perfiles 1, 2 y 3]
- Documentación técnica [Perfiles 1, 2 y 3]
- Gestión del cambio y del conocimiento [Perfiles 1, 2 y 3]
- Ingeniería de software [Perfiles 1, 2 y 3]
- Integración de sistemas [Perfiles 1, 2 y 3]
- Interfaz con el usuario final [Perfiles 1, 2 y 3]
- Matemáticas [Perfiles 1, 2 y 3]
- Metodologías de Configuración [Perfil 2]
- Métodos y Herramientas para el diseño y desarrollo de sistemas basados en computadores [Perfiles 1, 2 y 3]
- Programación [Perfiles 1, 2 y 3]

Interrelaciones con otras materias:

- Laboratorio de Programación I (primer curso), donde los alumnos ponen en práctica muchos de los conocimientos adquiridos en Elementos de Programación, además de impartirse algunos conceptos nuevos.
- Estructuras de Datos y Algoritmos (segundo curso), donde los alumnos tienen una continuación de la asignatura Elementos de Programación, centrándose principalmente en las estructuras de datos, algunas de las cuales ya se estudian en Elementos de Programación.
- Laboratorio de Programación II (segundo curso), donde se ponen en práctica los conocimientos adquiridos en Estructuras de Datos y Algoritmos, además de ampliar algunos conceptos vistos de forma superficial en Elementos de Programación, como la ampliación de Programación Orientada a Objetos con conceptos como Polimorfismo o Herencia.
- Análisis y Diseño de Sistemas (segundo curso), donde se entra en el mundo de la Ingeniería del Software, aunque el alumno, sin saberlo, ya se ha puesto en contacto con ella en Elementos de Programación, de una forma muy intuitiva.
- Estructuras de Almacenamiento de la Información (tercer curso), donde se estudian nuevas estructuras de datos, en este caso orientadas al almacenamiento en memoria secundaria.
- Programación Concurrente (tercer curso), donde se estudia un tipo de programación no estudiada hasta ese momento, pero que se basa en los mismos principios ya vistos en Elementos de Programación.

Este análisis muestra la importancia que tiene esta asignatura, pues sienta las bases de la programación, que es necesaria para varias asignaturas posteriores.

Contextualización personal*

Itinerarios de procedencia y requisitos formativos de los alumnos

Las diferentes opciones de acceso a la titulación de Ingeniería Técnica en Informática de Sistemas y de Gestión se pueden clasificar en las siguientes:

- C.O.U.
- Bachillerato Experimental.
- Bachillerato LOGSE.
- Con prueba de acceso de selectividad.
- Sin prueba de acceso de selectividad.
- Formación profesional. (ciclos formativos “Desarrollo de Aplicaciones Informáticas”, “Administración de Sistemas Informáticos”, “Sistemas de Telecomunicación e Informáticos”).
- Otras.

Los *conocimientos y experiencias previas* tienen bastante importancia, dado que es una asignatura que está pensada para iniciar al alumno en el mundo de la Programación, de tal forma que la procedencia y conocimientos previos van a distinguir claramente a aquellos alumnos que ya se hayan iniciado previamente en este mundo, principalmente, a que los alumnos que provienen de Formación Profesional (ciclos formativos “Administración de Sistemas Informáticos”, “Sistemas de Telecomunicación e Informáticos” y especialmente los alumnos del ciclo “Desarrollo de Aplicaciones Informáticas”) que ya habrán estado matriculados en varias asignaturas en la que han cursado muchas horas de Programación, lo que hace que los primeros meses de la asignatura sean mucho más accesibles para estos alumnos. Aproximadamente, el 6% acceden desde Formación Profesional. En cualquier caso, esta ventaja que los alumnos tienen en esta asignatura, se compensa con una menor cantidad de conocimientos previos en asignaturas como Matemáticas o Física, a las que se les dedica menor tiempo de clase en Formación Profesional que en Bachillerato y COU. Esta última “deficiencia” también se nota en la asignatura Elementos de Programación en cuanto es necesario utilizar matrices y cálculo de matrices (que son necesarios porque son bastante utilizados en algunos ámbitos más avanzados de programación) o cuando se utilizan cálculos numéricos como ejemplo.

Como acabamos de ver, los conocimientos básicos de Programación pueden condicionar (positivamente si se tienen) los primeros meses de trabajo en la asignatura. Así, podemos destacar algunos conocimientos previos que serían deseables que el alumno tuviera, (aunque actualmente no se exige a los alumnos que los tengan y son explicados suponiendo que no los tienen):

- Conocimientos básicos de Algoritmia: cómo funciona un algoritmo y cómo automatizar un proceso.
- Conocimientos básicos de algún lenguaje de Programación: estar familiarizado con algún lenguaje y con los procesos de compilación, depurado o ejecución.
- Conocimientos matemáticos necesarios para formalizar.

Aun así, algunos problemas de aprendizaje tienen su origen en algunas preconcepciones erróneas que suelen condicionar la comprensión de algunos conceptos. Un ejemplo de estas preconcepciones a evitar sería la teoría implícita acerca de la corrección de un programa informático (que un programa funcione no implica que sea correcto, existen muchos otros criterios posibles para estudiar su corrección).

Además, las características de la metodología de la programación exige del alumno que muestre una actitud positiva e interés por el trabajo bien hecho y su calidad final, y no limitarse a desarrollar un programa que realice la tarea encomendada.

II. Objetivos

<i>Relacionados con competencias académicas y disciplinares</i>	<i>Vinculación</i>
Descripción	<i>CET¹</i>
1.- Conocer y comprender los principios y conceptos fundamentales de la Programación.	21
2.- Analizar de manera crítica un determinado problema diseñando el algoritmo, y aprender a saber elegir la estructura de datos más adecuada para resolverlo.	5, 8, 9, 10, 11, 14, 16, 19, 21
3- Ser capaz de aplicar las metodologías de diseño descendente, programación estructurada y modular, al realizar el diseño de un algoritmo.	5, 7, 14, 16, 19, 21
4- Que el alumno sea capaz de comprender la programación orientada a objetos introduciéndose en ella mediante el uso de TADs.	14, 19, 21
5- Ser capaz de analizar, diseñar, codificar y corregir programas de complejidad pequeña-media.	5, 14, 19, 21
6- Que el alumno sea capaz de cumplir las normas de estilo apropiadas para la legibilidad de código y su posterior mantenimiento.	7, 11, 14, 19, 21
7- Garantizar que el alumno sea capaz de realizar un programa lo suficientemente completo incluyendo el uso de diferentes TAD's.	14, 19, 21

<i>Relacionados con otras competencias personales y profesionales</i>	<i>Vinculación</i>
Descripción ¹	<i>CG</i>
8- Adquirir hábitos correctos en la construcción de programas, y unas destrezas mínimas sobre creación y diseño de algoritmos, así como saber aplicarlas en un lenguaje de programación.	11, 14, 19, 21

¹ Algunas capacidades son objetivo de la titulación y no de una materia concreta.

III. Contenidos

<i>Secuenciación de bloques temáticos y temas</i>
1. Introducción.
1.1.- Conceptos básicos: algoritmo, proceso y procesador. 1.2.- Breve historia. 1.3.- Estructura y funcionamiento básico de un ordenador. 1.4.- Codificación y almacenamiento de la información. 1.5.- Paradigmas y lenguajes de programación.
2. Objetos, expresiones y acciones elementales.
2.1.- Entorno: objetos básicos. 2.2.- Tipos simples de datos. 2.3.- Operaciones y expresiones. 2.4.- Acciones elementales. 2.5.- Representación de algoritmos. 2.6.- Aplicación en C.
3. Programación Estructurada.
3.1.- Introducción. 3.2. Estructuras básicas de control. 3.3. Otras estructuras de control. 3.4. Metodología de resolución de problemas 3.5. Aplicación en C.
4. Programación Modular.
4.1.- Introducción. 4.2.- Diseño descendente. 4.3.- Módulos: concepto, tipos y parámetros. 4.4.- Recursividad. 4.5.- Aplicación en C.
5. Análisis de Algoritmos.
5.1.- Introducción. 5.2.- Eficiencia de programas. 5.3.- Complejidad algorítmica. Notación O.
6. Estructuras y Tipos Abstractos de Datos (TAD).
6.1.- Introducción. Estructuras de datos. 6.2.- Tipos abstractos de datos. Especificación de un TAD. 6.3.- Representación en C.
7. Registros.
7.1.- Introducción. 7.2.- Definición. 7.3.- Operaciones básicas. 7.4.- Registros variantes.
8. Vectores.
8.1.- Introducción. 8.2.- Definición. 8.3.- Operaciones básicas. 8.4.- Vectores de varias dimensiones. 8.5.- Búsqueda. 8.6.- Algoritmos de ordenación. 8.7.- Vectores con ocupación variable. 8.8.- Cadena de caracteres.

9. TADs lineales: Utilización e implementación estática.
9.1.- Introducción. 9.2.- TAD Pila. 9.3.- TAD Cola. 9.4.- TAD Lista. 9.5.- TAD Conjunto.
10. Objetos.
10.1.- Introducción. Conceptos básicos. 10.2.- Principios fundamentales de la Programación Orientada a Objetos. 10.3.- Representación. 10.4.- Implementación de TADs lineales con objetos.
11. Gestión dinámica de la memoria.
11.1.- Introducción. 11.2.- Punteros y variables dinámicas. 11.3.- Estructuras dinámicas. 11.4.- Lista enlazada.
12. TADs lineales: Utilización e implementación dinámica.
12.1.- TAD Lista. 12.2.- TAD Pila. 12.3.- TAD Cola. 12.4.- TAD Conjunto. 12.5.- TAD Cadena de caracteres. 12.6.- Otras estructuras dinámicas.

<i>Interrelación</i>			
Requisitos (Rq) y redundancias (Rd)		Tema	<i>Procedencia</i>
Conocimientos matemáticos básicos.	Rq	1-12	Cualquiera de los posibles itinerarios de procedencia
Conocimientos básicos de Informática a nivel de usuario.	Rq	1-12	Cualquiera de los posibles itinerarios de procedencia
Pilas, listas y colas	Rd	9, 11, 12	Laboratorio de Programación I (1º de I.T.I.S.)
Programación Orientada a Objetos	Rd	10	Laboratorio de Programación I (1º de I.T.I.S.)
Programación Orientada a Objetos	Rd	10	Laboratorio de Programación II (2º de I.T.I.S.)
Pilas, listas y colas	Rd	9, 11, 12	Estructuras de Datos y Algoritmos (2º de I.T.I.S.)
Complejidad Algorítmica	Rd	5	Estructuras de Datos y Algoritmos (2º de I.T.I.S.)
Tipos Abstractos de Datos	Rd	6	Estructuras de Datos y Algoritmos (2º de I.T.I.S.)

IV. Metodología docente y plan de trabajo del estudiante

<i>Actividades de enseñanza-aprendizaje</i>				<i>Vinculación</i>	
<i>Descripción y secuenciación de actividades</i>	<i>Tipoⁱⁱ</i>		<i>Dⁱⁱⁱ</i>	<i>Tema</i>	<i>Objet.</i>
1. Presentación de la asignatura	GG	C-E (I)	0,5	1-12	Todos
2. Encuesta de contextualización personal (procedencia, conocimientos previos, expectativas formativas y profesionales ...)	GG	C-E (I)	0,5	1-12	1, 9, 10, 13
3. Lectura previa del resumen del tema	NP	T (II)	0,5	1	1
4. Exposición general de conceptos básicos	GG	T (II)	0,5	1	1
5. Estudio de los contenidos explicados	NP	T (II)	0,5	1	1
6. Lectura previa del resumen del tema	NP	T (II)	1	2	1
7. Explicación del tema y ejemplos en clase	GG	T-P	1,5	2	1
8. Estudio de los contenidos explicados	NP	T (II)	1	2	1
9. Discusión y resolución de problemas	S	P (IV)	2	2	5, 6, 9, 10, 11, 13
10. Resolución de problemas propuestos	NP	P	2	2	5, 6, 9, 11, 13
11. Lectura previa del resumen del tema	NP	T (II)	1	3	3, 5, 6, 8, 9, 12
12. Explicación del tema, discusión y ejemplos en clase	GG	T-P	6,5	3	3, 5, 6, 8, 9, 10, 12
13. Estudio de los contenidos explicados	NP	T-P	3	3	3, 5, 6, 8, 9, 12
14. Discusión y resolución de problemas	S	P (IV)	2	3.2, 3.4, 3.5	3, 5, 6, 8, 9, 10, 11, 12, 13
15. Estudio y resolución de problemas propuestos	NP	P	2	3.2, 3.4, 3.5	3, 5, 6, 8, 9, 11, 12, 13
16. Discusión y resolución de problemas	S	P (IV)	2	3.3, 3.4, 3.5	3, 5, 6, 8, 9, 10, 11, 12, 13
17. Estudio y resolución de problemas propuestos	NP	P	2	3.3, 3.4, 3.5	3, 5, 6, 8, 9, 11, 12, 13
18. Lectura previa del resumen del tema	NP	T (II)	1	4	3, 5, 6, 8, 9, 12
19. Explicación del tema, discusión y ejemplos en clase	GG	T-P	6,5	4	3, 5, 6, 8, 9, 10, 12
20. Estudio de los contenidos explicados	NP	T-P	3	4	3, 5, 6, 8, 9, 12
21. Discusión y resolución de problemas	S	P (IV)	2	4.2, 4.3	3, 5, 6, 8, 9, 10, 11, 12, 13
22. Estudio y resolución de problemas propuestos	NP	P	2	4.2, 4.3	3, 5, 6, 8, 9, 11, 12, 13
23. Discusión y resolución de problemas	S	P (IV)	2	4.2, 4.3	3, 5, 6, 8, 9, 10, 11, 12, 13
24. Estudio y resolución de problemas propuestos	NP	P	2	4.2, 4.3	3, 5, 6, 8, 9, 11, 12, 13
25. * Trabajo por grupos	Tut	T-P	2	4 - 12	Todos
26. * Elaboración de Trabajo por grupos	NP	T-P	6	4 - 12	Todos
27. Lectura previa del resumen del tema	NP	T (II)	0,5	5	5, 8, 9
28. Explicación del tema, discusión y ejemplos en clase	GG	T-P	0,5	5	5, 8, 9, 10, 11
29. Estudio de los contenidos explicados	NP	T-P	1	5	5, 8, 9, 11
30. Lectura previa del resumen del tema	NP	T (II)	1	6	3, 5, 7, 8, 9, 12
31. Explicación del tema, discusión y ejemplos en clase	GG	T-P	1	6	3, 5, 7, 8, 9, 10, 11, 12
32. Estudio de los contenidos explicados	NP	T-P	2	6	3, 5, 7, 8, 9, 11, 12
33. Análisis, discusión y resolución de problemas	S	P (IV, V)	2	6	3, 5, 7, 8, 9, 10, 11, 12, 13
34. Estudio y resolución de problemas propuestos	NP	P	2	6	3, 5, 7, 8, 9, 10, 11, 12, 13
35. Lectura previa del resumen del tema	NP	T (II)	0,5	7	2, 3, 5, 6, 7, 8, 9, 12
36. Explicación del tema, discusión y ejemplos en clase	GG	T-P	0,5	7	2, 3, 5, 6, 7, 8, 9, 10, 11, 12
37. Estudio de los contenidos explicados	NP	T-P	1	7	2, 3, 5, 6, 7, 8, 9, 12
38. Lectura previa del resumen del tema	NP	T (II)	1	8	2, 3, 5, 6, 7, 8, 9, 12
39. Explicación del tema, discusión y ejemplos en clase	GG	T-P	5,5	8	2, 3, 5, 6, 7, 8, 9, 10, 11, 12
40. Estudio de los contenidos explicados	NP	T-P	4	8	2, 3, 5, 6, 7, 8, 9, 12
41. Análisis, discusión y resolución de problemas	S	P (IV, V)	2	8.2, 8.3, 8.4	2, 3, 5, 6, 7, 8, 9, 10, 11, 12, 13
42. Estudio y resolución de problemas propuestos	NP	P	2	8.2, 8.3, 8.4	2, 3, 5, 6, 7, 8, 9, 11, 12, 13
43. Trabajo práctico (fase 1)	GG	P (VI)	0,5	2-8	2, 3, 5, 6, 7, 8, 9, 10, 11, 12
44. * Trabajo por grupos de la actividad anterior	NP	P (VI)	10	2-8	2, 3, 5, 6, 7, 8, 9, 10, 11, 12, 13
45. * Consulta bibliográfica y recursos web	NP	T-P	1,5	8.6	9, 11, 12, 13
46. Análisis, discusión y resolución de problemas	S	P (IV, V)	2	8.5, 8.6, 8.7, 8.8	2, 3, 5, 6, 7, 8, 9, 10, 11, 12, 13
47. Estudio y resolución de problemas propuestos	NP	P	2	8	2, 3, 5, 6, 7, 8, 9, 11, 12, 13
48. * Tutorización y evaluación del Trabajo práctico (fase 1)	Tut	P (VI)	2	2-8	2, 3, 5, 6, 7, 8, 9, 10, 11, 12
49. Trabajo práctico (fase 2)	GG	P (VI)	0,5	2-12	Todos
50. * Trabajo por grupos de la actividad anterior	NP	P (VI)	48	2-12	Todos
51. Lectura previa del resumen del tema	NP	T (II)	1	9	2, 3, 5, 6, 7, 8, 9, 11, 12
52. Explicación del tema, discusión y ejemplos en clase	GG	T-P	2	9	2, 3, 5, 6, 7, 8, 9, 10, 11, 12
53. Estudio de los contenidos explicados	NP	T-P	2	6, 8, 9	2, 3, 5, 6, 7, 8, 9, 11, 12
54. Análisis, discusión y resolución de problemas	S	P (IV, V)	2	9	2, 3, 5, 6, 7, 8, 9, 10, 11, 12, 13
55. Estudio y resolución de problemas propuestos	NP	P	2	9	2, 3, 5, 6, 7, 8, 9, 11, 12, 13
56. * Estudio y preparación del examen de evaluación continua	NP	TP (VII)	10	1-9	2, 3, 5, 6, 7, 8, 9, 11, 12, 13

57.	Examen de evaluación continua	GG	C-E (I)	3	1-9	2, 3, 5, 6, 7, 8, 9,11,12,13
58.	Lectura previa del resumen del tema	NP	T (II)	1	10	4, 7, 8, 9, 11, 12
59.	Explicación del tema, discusión y ejemplos en clase	GG	T-P	3	10	4, 7, 8, 9, 10, 11, 12
60.	Estudio de los contenidos explicados	NP	T-P	1,5	9, 10	4, 7, 8, 9, 11, 12
61.	Estudio y resolución de problemas propuestos	NP	T-P	1,5	10	4, 7, 8, 9, 11, 12, 13
62.	* Tutorización y evaluación del Trabajo práctico (fase 2)	Tut	P (VI)	2	2-12	Todos
63.	Lectura previa del resumen del tema	NP	T (II)	1	11	2, 3, 5, 6, 7, 8, 9, 11, 12
64.	Explicación del tema, discusión y ejemplos en clase	GG	T-P	4	11	2, 3, 5, 6, 7, 8, 9, 10, 11, 12
65.	Estudio de los contenidos explicados	NP	T-P	1,5	11	2, 3, 5, 6, 7, 8, 9, 11, 12
66.	Estudio y resolución de problemas propuestos	NP	T-P	1,5	11	2, 3, 5, 6, 7, 8, 9, 11, 12, 13
67.	Lectura previa del resumen del tema	NP	T (II)	1	12	2, 3, 4, 5, 6, 7, 8, 9, 11, 12
68.	Explicación del tema, discusión y ejemplos en clase	GG	T-P	3,5	12	Todos
69.	Estudio de los contenidos explicados	NP	T-P	1,5	11-12	2, 3, 4, 5, 6, 7, 8, 9, 11, 12
70.	Estudio y resolución de problemas propuestos	NP	T-P	1,5	12	Todos
71.	* Consulta bibliográfica y recursos web	NP	T-P	1,5	8.6	9, 11, 12, 13
72.	* Estudio y preparación del examen final	NP	T-P(VII)	15	1-12	Todos
73.	Examen final	GG	C-E (I)	3	1-12	Todos
74.	Examen final prácticas	S	C-E (I)	2	1-12	Todos

<i>Distribución del tiempo (ECTS)</i>		<i>Dedicación del alumno</i>		<i>Dedicación del profesor</i>		
<i>Distribución de actividades</i>		<i>Nº alumnos</i>	<i>H. presenciales</i>	<i>H. no presenc.</i>	<i>H. presenciales</i>	<i>H. no presenc.</i>
Grupo grande (Más de 20 alumnos)	Coordinac./evaluac. (I)	60	7	-	7	60
	Teóricas (II y III)	60	21,5	12	21,5	50
	Prácticas (IV, V y VI)	60	11,5	28	11,5	0
	Subtotal	60	40	40	40	110
Seminario- Laboratorio (7-20 alumnos)	Coordinac./evaluac. (I)	20	2	-	6	30
	Teóricas (II y III)	20	0	-	-	-
	Prácticas (IV, V y VI)	20	18	18	54	36
	Subtotal	20	20	18	60	66
Tutoría ECTS (1-6 alumnos)	Coordinac./evaluac. (I)	6	0,5	-	5	60
	Teóricas (II y III)	6	-	3	-	-
	Prácticas (IV, V y VI)	6	5,5	43	55	10
	Subtotal	6	6	46	60	70
Tutoría comp. y preparación de ex. (VII)		1	-	25	-	10
Totales			66 (2,64 ECTS)	129 (5,16 ECTS)	160	256

*Otras consideraciones metodológicas**

Recursos y metodología de trabajo en las actividades presenciales

Las actividades expositivas en grupos grandes cumplirán la función de vertebrar el proceso de aprendizaje del alumno a lo largo de la asignatura. Previamente los alumnos deberán realizar una lectura previa de la documentación teórica de cada tema, lo que facilitará una base previa de construcción de los conocimientos, así como una mayor participación de los alumnos durante la explicación. Se trata de que el profesor puede dedicar más tiempo a “mediar” en ese proceso de reelaboración (y no tanto a “transmitir” información).

Para facilitar la integración y el registro de la información nueva que se explica en clase los alumnos contarán también con un esquema del tema, que deberán completar durante la explicación. El tipo de esquema se adecuará a la organización de las ideas que nos interese resaltar.

Los seminarios consistirán en hacer varios ejercicios que deberán realizar en el laboratorio y después ampliar en casa.

Recursos y metodología de trabajo en las actividades semi-presenciales y no presenciales

En los seminarios en grupos pequeños (tutorías ECTS) se realizarán principalmente dos tipos de actividades:

- Elaboración de materiales de los distintos temas vistos en teoría.
- Elaboración de una práctica que consistirá en el Análisis, Diseño e Implementación de un caso práctico simulando un problema real.

Las actividades referentes a la elaboración de la práctica se dividirá en dos fases:

- Fase 1: Corrección y posterior asesoramiento de la práctica a entregar. Esta práctica consistirá en el Análisis de un caso práctico simulando un problema real.
- Corrección y asesoramiento de la fase 2 de la práctica. En esta fase se diseñará e implementará el programa informático y la interacción profesor-alumno puede ser mayor que en la fase 1.

V. Evaluación

<i>Criterios de evaluación*</i>	<i>Vinculación*</i>	
	<i>Objetivo</i>	<i>CC^{iv}</i>
1. Demostrar la adquisición y comprensión de los conceptos fundamentales de la asignatura		30%
2. Resolver problemas aplicando los conocimientos teóricos y los conocimientos adquiridos en los supuestos prácticos realizados		35%
3. Desarrollar un programa informático que sea correcto desde el punto de vista de la metodología de la Programación		25%
4. Participar activamente en las tutorías ECTS y en los seminarios.		5% N.R.
5. Asistir a seminarios		5%

<i>Actividades e instrumentos de evaluación</i>		
Seminarios	<ul style="list-style-type: none"> • Valoración de las actividades prácticas junto con la evaluación continua del trabajo y la dedicación en el desarrollo de la misma. 	10% (NR) ^v
Tutorías ECTS	<ul style="list-style-type: none"> • Observación de la participación en las actividades prácticas • Registro y valoración de las actividades prácticas entregadas por el alumno. 	20% (NR) ^{vi}
Examen final	<ul style="list-style-type: none"> • Prueba escrita de preguntas cortas. • Prueba escrita orientada a valorar la capacidad de resolver casos prácticos 	70%

VI. Bibliografía

<i>Bibliografía de apoyo seleccionada</i>
<ul style="list-style-type: none"> ▪ S. Leestma, L. Nyhoff, Programación en Pascal 4ª edición. Prentice Hall, 1999. ▪ Deitel H. y Deitel P. “Cómo programar en C/C++”. Prentice Hall 1995. ▪ Joyanes L. “Programación en C++. Algoritmos, estructuras de datos y objetos”. McGraw-Hill, 2000. ▪ Savitch W. “Resolución de problemas con C++”. 2ª edición. Prentice Hall 2000. ▪ Eckel B. “Thinking in C++”. 2nd edition. Prentice Hall 2000. ▪ Schildt H. “Borland C++. Manual de referencia”. McGraw-Hill, 1999. ▪ Stroustrup B. “El lenguaje de programación C++”. Addison Wesley 2002.
<i>Bibliografía o documentación de lectura obligatoria*</i>
Resumen de cada uno de los temas (elaborados por los profesores de la asignatura)
<i>Bibliografía o documentación de ampliación, sitios web...*</i>
<ul style="list-style-type: none"> ▪ Castro J y otros, “Curso de programación”. McGraw-Hill, 1993 ▪ Joyanes L. “Problemas de metodología de la programación”. McGraw-Hill, 1990. ▪ Peña R. “Diseño de programas. Formalismo y abstracción”. Prentice may, 1998 ▪ Pressman R. “Software Engineering: A Practitioner’s Approach”. McGraw-Hill, 1997. ▪ Wirth N. “Algoritmos + Estructuras de datos = Programas”. Ediciones del Castillo, 1980. ▪ Aprenad ANSI C como si estuviera en primero. http://dalila.sip.ucm.es/miembros/albertoe/downloads/AnsiC/leng_c.pdf ▪ Aprenda C++ como si estuviera en primero. http://mec21.etsii.upm.es/ayudainf/aprendainf/Cpp/manualcpp.pdf

Códigos.-

ⁱ *CET*: Competencias Específicas del Título (véase el apartado de Contextualización curricular)

ⁱⁱ *Tipos de actividades*: GG (Grupo Grande); S (Seminario o Laboratorio); Tut (Tutoría ECTS); No presenciales (NP); C-E, I (Coordinación o evaluación); T, II (Teórica de carácter expositivo o de aprendizaje a partir de documentos); T, III (Teórica de discusión); P, IV (Prácticas basadas en la solución de problemas); P, V (Prácticas basadas en la observación, experimentación, aplicación de destrezas, estudio de casos...); P, VI (Prácticas con proyectos o trabajos dirigidos); T-P, VII (Otras teórico-prácticas).

ⁱⁱⁱ *D*: Duración en sesiones de 1 hora de trabajo presencial o no presencial (considerando en cada hora 50-55 minutos de trabajo neto y 5-10 de descanso).

^{iv} *CC*: Criterios de Calificación (ponderación del criterio de evaluación en la calificación cuantitativa final).

^v *NR*: actividad “no recuperable” o que no permite evaluación extraordinari

(*) Apartados no obligatorios

^{vi} *NR*: actividad “no recuperable” o que no permite evaluación extraordinari

(*) Apartados no obligatorios