

## Plan Docente de una materia

# “Laboratorio de Programación I”

### I. Descripción y contextualización

<i>Identificación y características de la materia</i>				
<i>Denominación</i>	Laboratorio de Programación I			
<i>Curso y Titulación</i>	1º Curso. I.T. en Informática de Sistemas.			
<i>Área</i>	Lenguajes y Sistemas Informáticos.			
<i>Departamento</i>	<i>Departamento de informática.</i>			
<i>Tipo</i>	Obligatoria	6 (1.5 T + 4.5 P)		
<i>Coeficientes</i>	Practicidad 4		Agrupamiento 2	
<i>Duración ECTS (créditos)</i>	2º Cuatrimestre		5.2 ECTS (130 Horas).	
<i>Distribución ECTS (rangos)</i>	Grupo Grande: 10%	Seminario-Lab.: 30%	Tutoría ECTS: 5%	No presenciales: 55%
	13 horas	39-40 horas	6-7 horas	71-72 horas
<i>Descriptor</i> <i>(según BOE)</i>	Diseño de programas: descomposición modular y documentación			
<i>Coordinador-Profesor/es</i>	Francisco Chávez de la O			
<i>Tutorías complementarias (1)</i>	Despacho 9	Ext Telef. 2529	fchavez@unex.es	

## *Contextualización profesional*

En la planificación de un proyecto docente no se debe perder de vista en qué van a derivar las enseñanzas impartidas y observar hasta qué punto existe una cohesión o relación entre la materia de estudio y la necesidad de mercado. Por eso tratamos este contexto.

La actividad profesional del ingeniero técnico en informática responde a dos grandes líneas:

*Perfil profesional:* Determinado por la función a desarrollar (Desarrollo, diseño, comercial, etc.), el área de aplicación (microelectrónica, hardware, software, lenguajes informáticos, telecomunicaciones, etc.) y el nivel de decisión (técnico, económico, político, etc.)

*Entorno profesional:* Que hace referencia a las modalidades del ejercicio profesional (empleado, empresario, funcionario, etc.), el marco en que se desarrolla la actividad (empresa multinacional, nacional, pública, etc.).

A continuación se muestra una posible clasificación de funciones aplicables a posibles salidas profesionales del Ingeniero Técnico en Informática que tienen relación directa con el Plan Docente que aquí se describe.

Perfil profesional con relación directa:

### *Desarrollo.*

La función de desarrollo parte de ideas o dispositivos existentes para tratar de construir nuevos prototipos, nuevos programas para la conexión correcta de dichos dispositivos, programas de mejora en la interconexión de elementos internos del computador, etc.. Teniendo en cuenta esta rama del desarrollo es imprescindible que el profesional conozca tanto los componentes internos de un computador, lenguajes informáticos aplicable, modo de comunicación y funcionamiento, para que estos desarrollos tengan un fin con éxito.

### *Producción, Operación y Mantenimiento.*

Englobamos con el nombre fabricación a las funciones de producción, operación, mantenimiento y control de calidad de productos en serie, mientras que construcción se relaciona con grandes proyectos.

Ingeniería de producción es el término de todas aquellas tareas que se encargan del desarrollo del proceso de fabricación de cada parte del producto y de su ejecución. El ingeniero técnico de producción coordinará hombres y máquinas, y además de tener conocimientos técnicos generales, debe manejar técnicas de programación de trabajos, métodos de cálculo de tiempos y costes, fundamentos de derecho industrial y laboral, etc.

### *Administración de Sistemas*

El ingeniero técnico de administración y mantenimiento debe conocer el funcionamiento de los sistemas que debe administrar, por lo que es esencial conocer los lenguajes de programación capaces de interactuar con estos sistemas.

Perfil profesional con relación indirecta:

### *Aplicación y Ventas.*

La ingeniería comercial está ligada al mercado y al cliente. La Ingeniería de software se encarga de configurar un producto adaptado a las necesidades específicas de un cliente. El comercial, además de vender, debe también de servir como instructor del cliente en las características, usos, propiedades y ventajas de sus productos.

Las características que deben poseer estos profesionales son conocer los principios básicos de su rama y especializarse en los sistemas que venden, tener además don de gentes y habilidad en las relaciones humanas con los clientes. Para todo ello es fundamental conocer la tecnología tanto hardware como software disponibles y asesorar al cliente a la hora de tomar una decisión.

El directivo suele ser una persona con gran experiencia, que conoce diversos departamentos o funciones de ingeniería, capaz de tomar decisiones y hacerlas cumplir, y con un gran prestigio a lo largo de su carrera.

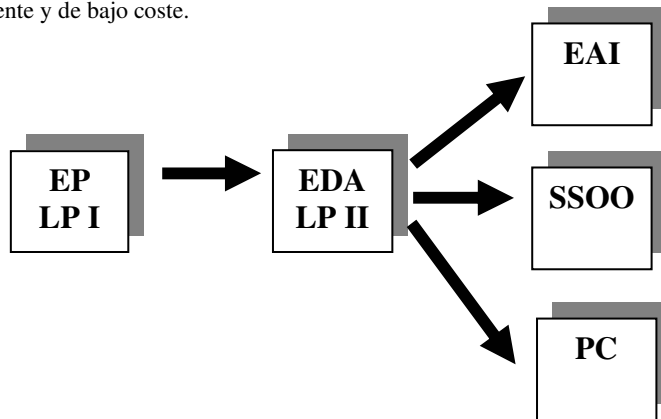
## Contextualización curricular

Conexión con las competencias genéricas y específicas del Título.

Para elaborar un programa hay que tener en cuenta el perfil del alumno al que va dirigido, su formación previa y el desarrollo posterior de su programa de estudios. Partir exclusivamente de unos contenidos preconcebidos no conduce a ningún resultado positivo. Igualmente hay que valorar la respuesta del alumno a las enseñanzas que se van impartiendo (las preguntas que formulan, cómo las formulan, los ejercicios que resuelven, las dudas que tienen, etc.), como un indicador dinámico de su nivel de asimilación. De aquí que el programa deba considerarse como un marco de referencia general, susceptible de ser dinámicamente ampliado o simplificado en puntos determinados de su desarrollo, según lo aconsejen las circunstancias de cada momento.

Para entender el papel que la asignatura Laboratorio de Programación I juega en la titulación, debemos ser conscientes de su conexión con otras asignaturas, el contexto en que se desarrolla. Para ello, mostramos en primer lugar las relaciones existentes entre esta asignatura y la titulación.

Laboratorio de Programación I, como asignatura de primer curso, segundo cuatrimestre, se apoya en Elementos de Programación, asignatura de primer curso, anual, en la que el alumno toma por primera vez contacto con la programación. Y está estrechamente relacionada con asignaturas posteriores: Estructuras de Datos y Algoritmos de segundo curso, anual y Laboratorio de Programación II, de segundo curso, segundo cuatrimestre. Con estas cuatro asignaturas se pretende dar al alumno un amplio conocimiento sobre programación eficiente y de bajo coste.



Otras conexiones:

Existen conexiones no descritas en los apartados anteriores, con asignaturas de tercer curso, en las que el alumno necesita una buena base de programación para llevarlas a cabo de forma adecuada, como es el caso de Estructuras y Almacenamiento de la Información, Sistemas Operativos y Programación Concurrente, en las cuales es imprescindible realizar unas prácticas usando todos los conocimientos que el alumno haya adquirido de programación.

Las competencias del título con las que se relaciona la asignatura de Laboratorio de Programación I son las siguientes:

- 1.- Ser capaz de evaluar los diferentes productos software/hardware del mercado.
- 2.- Ser capaz de elegir adecuadamente productos software/hardware dependiendo del problema que deba resolverse.
- 3.- Ser capaz de asesorar en la elección de productos a terceras partes.
- 4.- Ser capaz de analizar problemas y proponer soluciones técnicas basadas en equipamiento informático.
- 5.- Ser capaz de programar soluciones software a problemas.
- 6.- Ser capaz de administrar Sistemas de Información.
- 7.- Capacidad de evaluación técnica de tecnología de comunicaciones.
- 8.- Capacidad de elección y aplicación adecuada de tecnología de comunicaciones.
- 9.- Capacidad de integración de tecnologías de información y comunicaciones para dar respuesta a problemas.

## *Contextualización personal*

Tal como se describe en la sección anterior, el alumno llega a la asignatura que aquí se analiza con conocimientos de programación estructurada, conocimientos que son imprescindibles para entender lo que en esta asignatura se desarrolla.

Uno de los problemas más frecuentes es que el alumno se matricula en esta asignatura de primer curso sin haber superado o asistido a la asignatura Elementos de Programación. En estos casos, el deficiente conocimiento sobre la programación estructurada interfieren en la adquisición de los conceptos nuevos sobre programación orientada a objetos.

Para hacer más atractiva la asignatura se proponen como prácticas a realizar por los alumnos paso a paso, el desarrollo de juegos (algunos disponibles en los móviles de última generación y otros juegos típicos como tres en raya, tableros con figuras, etc. para que el alumno desarrolle capacidad creativa y a su vez ponga en práctica todos los conocimientos adquiridos y solvete los problemas con los que se va encontrando.

## II. Objetivos

<i>Relacionados con competencias académicas y disciplinares</i>	<i>Vinculación</i>
Descripción	<i>CET<sup>i</sup></i>
1. Iniciar al alumno en el desarrollo de programas informáticos siguiendo una metodología de programación estructurada.	5, 7, 14, 21
2. Consolidar la metodología de diseño de programas estructurada e iniciar en la programación orientada a objetos.	5, 7, 14, 21
3. Iniciar al alumno en los conocimientos necesarios para comprender y aplicar con fluidez las estructuras de datos adecuadas.	5, 7, 14, 21
4. Permitir a los alumnos resolver supuestos que supongan la implementación de programas de tamaño medio, con codificaciones correctas y eficientes, utilizando el paradigma de la programación estructurada junto con la orientada a objetos.	5, 7, 14, 21
5. Preparar al alumno para la implementación de programas de gran tamaño y complejidad.	5, 7, 14, 21
6. Iniciar al alumno para su futura participación en proyectos de desarrollo de software en entornos profesionales.	5, 7, 9, 14, 21

<i>Relacionados con otras competencias personales y profesionales</i>	<i>Vinculación</i>
Descripción	<i>CG</i>
7. Formarse y actualizar conocimientos de forma continuada.	16,17
8. Trabajar en equipo.	9
9. Trabajar con constancia.	9
10. Resolver problemas con creatividad y confianza en los propios conocimientos.	18,21

### III. Contenidos

#### *Secuenciación de bloques temáticos y temas*

#### **Unidad temática I. Inicialización a la programación orientado a objetos.**

Tema 1. Introducción a la programación orientada a objetos.

Tema 2. Clases y métodos.

- 2.1. Terminología clases.
- 2.2. Clases.
- 2.3. Encapsulación.
- 2.4. Constructores
- 2.5. Destructores
- 2.6. Descomposición en módulos
- 2.7. Un array de objetos
- 2.8. Un objeto con un puntero interno
- 2.9. Punteros a clases
- 2.10. Ejercicios propuestos.

#### **Unidad temática II. Archivos y estructuras de datos estáticas.**

Tema 3. E/S, flujos y archivos

- 3.1. Introducción.
- 3.2. Abrir y cerrar archivos.
- 3.3. Escritura y lectura en archivos de texto
  - 3.3.1. Leer de un archivo de texto.
  - 3.3.2. Añadir datos a un archivo.
- 3.4. Apertura de múltiples archivos
- 3.5. Leer y escribir en archivos binarios.
  - 3.5.1. Detección de EOF.
- 3.6. Acceso aleatorio.
- 3.7. Ejercicios.
  - 3.7.1. Ejercicios resueltos.
  - 3.7.2. Ejercicios propuestos.

Tema 4. Pilas estáticas.

- 4.1. Pilas.
  - 4.1.1. Estructura del elemento de pila.
  - 4.1.2. Método "Crear pila".
  - 4.1.3. Método "Apilar".
  - 4.1.4. Método "Cima".
  - 4.1.5. Método "Borrar cima".
  - 4.1.6. Método "Pila llena".
  - 4.1.7. "Método "Pila Vacía".
- 4.2. Ejemplo completo de una pila estática.
  - 4.2.1. Fichero pila.h.
  - 4.2.2. Fichero pila.cpp.
  - 4.2.3. Fichero main.cpp.

Tema 5. Colas estáticas.

- 5.1. Colas.
  - 5.1.1. Estructura del elemento de cola.
  - 5.1.2. Método “Crear cola”.
  - 5.1.3. Método “Encolar”.
  - 5.1.4. Método “Primero”.
  - 5.1.5. Método “Eliminar”.
  - 5.1.6. Método “Cola llena”.
  - 5.1.7. Método “Cola Vacía”.
- 5.2. Ejemplo completo de una cola estática.
  - 5.2.1. Fichero cola.h.
  - 5.2.2. Fichero cola.cpp.
  - 5.2.3. Fichero main.cpp.

**Unidad temática III. Gestión dinámica de memoria.**

Tema 6. Gestión dinámica.

- 6.1. Punteros.
- 6.2. Los operadores punteros.
- 6.3. Inicialización de punteros.
- 6.4. Puntero NULL.
- 6.5. Aritmética de punteros.
- 6.6. Introducción a los punteros const.
- 6.7. Indexación a punteros: direccionamientos indirecto múltiple.
- 6.8. Reserva dinámica de memoria.
- 6.9. Problemas con los punteros.
- 6.10. Punteros a arrays de caracteres.
- 6.11. Generación de un puntero a un array.
- 6.12. Arrays de punteros.
- 6.13. Tamaño de un array
- 6.14. Asignación dinámica de arrays.
- 6.15. Punteros a funciones.
- 6.16. Ejercicios.
  - 6.16.1. Ejercicios resueltos.
  - 6.16.2. Ejercicios propuestos.

**Unidad temática IV. Estructuras de datos dinámicas.**

Tema 7. Listas dinámicas.

- 7.1. Listas.
  - 7.1.1. Estructura del elemento de lista.
  - 7.1.2. Método “Crear lista”.
  - 7.1.3. Método “Insertar”
  - 7.1.4. Método “Iniciar”.
  - 7.1.5. Método “Leer”.
  - 7.1.6. Método “Avanza”.
  - 7.1.7. Método “Lista Vacía”.
  - 7.1.8. Método “Mostrar”.
  - 7.1.9. Método “Borrar lista”.
- 7.2. Ejemplo completo de una lista dinámica.
  - 7.2.1. Fichero lista.h.
  - 7.2.2. Fichero lista.cpp.
  - 7.2.3. Fichero main.cpp.

Tema 8. Pilas dinámicas.

- 8.1. Pilas.
  - 8.1.1. Estructura del elemento de pila.
  - 8.1.2. Método “Crear pila”.
  - 8.1.3. Método “Apilar”.
  - 8.1.4. Método “Cima”.
  - 8.1.5. Método “Desapilar”.
  - 8.1.6. Método “Pila llena”.
  - 8.1.7. “Método “Borrar pila”.
- 8.2. Ejemplo completo de una pila dinámica.
  - 8.2.1. Fichero pila.h.
  - 8.2.2. Fichero pila.cpp.
  - 8.2.3. Fichero main.cpp.

Tema 9. Colas dinámicas.

- 9.1. Colas.
  - 9.1.1. Estructura del elemento de cola.
  - 9.1.2. Método “Crear cola”.
  - 9.1.3. Método “Encolar”.
  - 9.1.4. Método “Primero”.
  - 9.1.5. Método “Desencolar”.
  - 9.1.6. Método “Cola llena”.
  - 9.1.7. Método “Borrar cola”.
- 9.2. Ejemplo completo de una cola dinámica.
  - 9.2.1. Fichero cola.h.
  - 9.2.2. Fichero cola.cpp.
  - 9.2.3. Fichero main.cpp.

*Interrelación*

Requisitos (Rq) y redundancias (Rd)		Tema	Procedencia
Programación imperativa.	Rq	1 - 9	Elementos de Programación (1º Anual)
Conocimiento de Lenguaje C.	Rq	1 - 9	Elementos de Programación (1º Anual)

## 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>Tipo<sup>ii</sup></i>		<i>D<sup>iii</sup></i>	<i>Tema</i>	<i>Objet.</i>
1. Presentación de la asignatura	GG	C-E	0,5	1-9	Todos
2. Encuesta de conocimientos previos	GG	C-E	0,5	1-9	Todos
3. Sesión de prácticas con resolución de ejemplos cortos implementando tipos abstractos de datos.	S	T-P	3	1	1, 7, 8, 9, 10
4. Aplicación en C++ de supuestos prácticos de la actividad anterior	NP	P	3	1	1, 7, 8, 9, 10
5. Lectura previa del tema	NP	T	2	1	1, 7, 8, 9, 10
6. Exposición general de los conceptos de programación orientada a objetos.	GG	T-P	2	1	1, 7, 8, 9, 10
7. Sesión de prácticas con resolución de ejemplos cortos implementando tipos abstractos de datos.	S	T-P	3	1	1, 7, 8, 9, 10
8. Aplicación en C++ de supuestos prácticos de la actividad anterior	NP	P	3	1	1, 7, 8, 9, 10
9. Sesión de prácticas con resolución de ejemplos cortos implementando tipos abstractos de datos.	S	T-P	3	1	1, 7, 8, 9, 10
10. Aplicación en C++ de supuestos prácticos de la actividad anterior	NP	P	3	1	1, 7, 8, 9, 10
11. Asistencia a tutorías complementarias para aclaración de dudas.	Tut	T-C-E	1	1	1, 7, 8, 9, 10
12. Realización de problemas propuestos en las tutorías complementarias.	NP	P-C-E	2	1	1, 7, 8, 9, 10
13. Lectura previa del tema	NP	T	1	2	1, 2, 4, 7, 8, 9, 10
14. Exposición del concepto de clases y objetos.	GC	T-P	1,5	2	1, 2, 4, 7, 8, 9, 10
15. Solución de problemas en el computador, enfocados a la práctica utilizando orientación a objetos.	S	T-P	3	1-2	1, 2, 4, 7, 8, 9, 10
16. Aplicación en C++ de supuestos prácticos de la actividad anterior	NP	P	3	1-2	1, 2, 4, 7, 8, 9, 10
17. Lectura previa del tema	NP	T	1	3	1, 2, 4, 7, 8, 9, 10
18. Exposición de conocimientos para tratamiento de ficheros en C++.	GC	T-P	0,5	3	1, 2, 4, 7, 8, 9, 10
19. Resolución de problemas cortos sobre el computador, enfocados a la resolución de la práctica propuesta combinando todos los conocimientos previos.	S	T-P	3	1-3	1, 2, 4, 7, 8, 9, 10
20. Aplicación en C++ de supuestos prácticos de la actividad anterior	NP	P	3	3	1, 2, 4, 7, 8, 9, 10
21. Asistencia a tutorías complementarias para aclaración de dudas.	Tut	P-C-E	1	1, 2, 3	1, 2, 4, 7, 8, 9, 10
22. Realización de problemas propuestos en las tutorías complementarias.	NP	P-C-E	1	1, 2, 3	1, 2, 4, 7, 8, 9, 10
23. Lectura previa del tema	NP	T	1	4	1, 2, 3, 4, 7, 8, 9, 10
24. Exposición de conocimientos de estructuras estáticas (Pila estática).	GC	T-P	1	4	1, 2, 3, 4, 7, 8, 9, 10
25. Resolución de problemas cortos sobre el computador, enfocados a la resolución de la práctica propuesta.	S	T-P	3	1-4	1, 2, 3, 4, 7, 8, 9, 10
26. Aplicación en C++ de supuestos prácticos de la actividad anterior	NP	P	3	1-4	1, 2, 3, 4, 7, 8, 9, 10
27. Lectura previa del tema	NP	T	1	5	1, 2, 3, 4, 7, 8, 9, 10
28. Exposición de conocimientos de estructuras estáticas (Cola estática).	GC	T-P	1	5	1, 2, 3, 4, 7, 8, 9, 10



29.	Resolución de problemas cortos sobre el computador, enfocados a la resolución de la práctica propuesta.	S	P	3	1-5	1, 2, 3, 4, 7, 8, 9, 10
30.	Aplicación en C++ de supuestos prácticos de la actividad anterior	NP	P	3	1-5	1, 2, 3, 4, 7, 8, 9, 10
31.	Asistencia a tutorías complementarias para aclaración de dudas.	Tut	P-C-E	1	1-5	1, 2, 3, 4, 7, 8, 9, 10
32.	Realización de problemas propuestos en las tutorías complementarias.	NP	P-C-E	1	1-5	1, 2, 3, 4, 7, 8, 9, 10
33.	Lectura previa del tema	NP	T	2	6	Todos
34.	Gestión de memoria en C++	GC	T-P	2	6	Todos
35.	Resolución de problemas cortos sobre el computador, enfocados a la resolución de la práctica propuesta.	S	P	3	1-6	Todos
36.	Aplicación en C++ de supuestos prácticos de la actividad anterior	NP	P	3	1-6	Todos
37.	Estudio y preparación del examen parcial práctico	NP	P-C-E	5	1-6	Todos
38.	Realización del examen parcial práctico	S	P-C-E	3	1-6	Todos
39.	Lectura previa del tema	NP	T	1	7	Todos
40.	Exposición de conocimientos de estructuras dinámicas (Lista dinámica).	GC	T-P	2	7	Todos
41.	Asistencia a tutorías complementarias para aclaración de dudas.	Tut	P-C-E	1	1-7	Todos
42.	Realización de problemas propuestos en las tutorías complementarias.	NP	P-C-E	2	1-7	Todos
43.	Lectura previa del tema	NP	T	1	8	Todos
44.	Exposición de conocimientos de estructuras dinámicas (Pila dinámica).	GC	T-P	1	8	Todos
45.	Resolución de problemas cortos sobre el computador, enfocados a la resolución de la práctica propuesta.	S	P	3	1-7	Todos
46.	Aplicación en C++ de supuestos prácticos de la actividad anterior	NP	P	3	1-7	Todos
47.	Lectura previa del tema	NP	T	1	9	Todos
48.	Exposición de conocimientos de estructuras dinámicas (Cola dinámica).	GC	T-P	1	9	Todos
49.	Resolución de problemas cortos sobre el computador, enfocados a la resolución de la práctica propuesta.	S	P	3	1-7	Todos
50.	Aplicación en C++ de supuestos prácticos de la actividad anterior	NP	P	3	7	Todos
51.	Asistencia a tutorías complementarias para aclaración de dudas.	Tut	P-C-E	1	1-9	Todos
52.	Realización de problemas propuestos en las tutorías complementarias.	NP	P-C-E	1	1-9	Todos
53.	Realización del examen teórico	GC	T-C-E	1	1-9	Todos
54.	Resolución de problemas cortos sobre el computador, enfocados a la resolución de la práctica propuesta.	S	P	3	1-9	Todos
55.	Aplicación en C++ de supuestos prácticos de la actividad anterior	NP	P	3	1-9	Todos
56.	Estudio y preparación del examen parcial práctico	NP	P-C-E	5	1-9	Todos
57.	Realización del examen parcial práctico	S	P-C-E	3	1-9	Todos
58.	Estudio y preparación del examen final.	NP	P-C-E	8	1-9	Todos
59.	Examen final	S	C-E (I)	3	1-9	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	1	0	2	20
	Teóricas (II y III)	60	12	11	11	22
	Prácticas (IV, V y VI)	60	-	-	-	-
	Subtotal	60	13	11	13	42
Seminario-Laboratorio (6-20 alumnos)	Coordinac./evaluac. (I)	20	-	-	-	-
	Teóricas (II y III)	20	5	-	5	15
	Prácticas (IV, V y VI)	20	28	33	28	40
	Subtotal	20	33	33	33	55
Tutoría ECTS (1-5 alumnos)	Coordinac./evaluac. (I)	5	-	-	-	12
	Teóricas (II y III)	5	1	-	12	-
	Prácticas (IV, V y VI)	5	4	7	48	-
	Subtotal	5	5	7	60	12
Tutoría comp. y preparación de ex. (VII)		1	10	19	36	20
Totales			61 (2,4 ECTS)	70 (2,8 ECTS)	142	129

<i>Otras consideraciones metodológicas*</i>
<b>Recursos y metodología de trabajo en las actividades presenciales</b>
<p>Las actividades expositivas cumplen 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 facilita 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 pueda dedicar más tiempo a “mediar” en ese proceso de reelaboración (y no tanto a “transmitir” información).</p> <p>Para facilitar la integración y el registro de la información nueva que se explica en clase, los alumnos deberán aplicar esos conocimientos nuevos a la práctica que se propone, debatiendo las posibles soluciones que se aporten, considerando ventajas e inconvenientes de cada una de ellas.</p>

## V. Evaluación

<i>Criterios de evaluación*</i>	<i>Vinculación*</i>	
Descripción	<i>Objetivo</i>	<i>CC<sup>iv</sup></i>
Para superar la asignatura el alumno deberá realizar una práctica desarrollada de forma <i>individual</i> , en lenguaje C++. La práctica debe entregarse debidamente resuelta y documentada en la fecha indicada. La documentación debe consistir en el <i>Manual del Usuario</i> y <i>Manual del Programador</i> (que incluirá el código fuente), entregándose también por duplicado en disco los ficheros fuente de la práctica y modificación.	Todos	
Evaluación		
Examen teórico que consistirá en contestar un test sobre los conocimientos teóricos desarrollados a lo largo de la asignatura.	Todos	10 %
Examen práctico que consistirá en realizar una modificación sobre el ordenador, a la práctica entregada por el alumno.	Todos	80 %
Estructuración, resolución y presentación de la práctica.	Todos	5 %
Asistencia y participación en las sesiones prácticas en laboratorio.	Todos	5 %

<i>Actividades e instrumentos de evaluación</i>		
Seminarios y Tutorías ECTS	<ul style="list-style-type: none"> <li>• Valoración de la asistencia y participación en las actividades teóricas y prácticas.</li> <li>• Registro y valoración de las actividades prácticas entregadas por el alumno.</li> </ul>	10% (NR) <sup>v</sup>
Exámenes parciales	<ul style="list-style-type: none"> <li>• Examen teórico tipo test.</li> <li>• Modificación realizada en laboratorio por los alumnos sobre la primera parte de la práctica propuesta. Si se obtiene la calificación de apto en esta evaluación, permitirá a los alumnos poder presentarse a la evaluación contemplada en el siguiente punto.</li> <li>• Modificación realizada en laboratorio por los alumnos sobre la práctica propuesta completa. Superando esta prueba, los alumnos habrán superado la asignatura y no necesitan presentarse al examen final.</li> </ul>	10 % (NR) <sup>vi</sup>  80%
Examen final	<ul style="list-style-type: none"> <li>• Modificación a realizar en la práctica propuesta.</li> </ul>	(80%)

## VI. Bibliografía

<i>Bibliografía de apoyo seleccionada</i>
<p>Herbert Schildt. “Borland C++. Manual de Referencia”. McGraw-Hill, 1997.</p> <p>Walter Savitch. “Resolución de problemas con C++. 2ª Edición “. Prentice Hall, 2000.</p> <p>Adam Drozdek. “Data Structures and Algorithms in C++. 2ª Edición”. Thomson Learning.</p> <p>Deitel y Deitel. “Como programar C++”. Prentice Hall, 1999.</p>

Moshe J. Augenstein, Aaron M. Tenenbaum. "Estructuras de Datos con C y C++".  
Prentice Hall, 1997.

Bjarne Stroustrup. "El lenguaje de Programación C++".  
Addison Wesley. 2002.

Luis Joyanes. "Fundamentos de la Programación". Mc-Graw Hill.  
Tercera edición. 2003.

Ricardo Peña Marí, "Diseño de Programas. Formulismo y Abstracción".  
Prentice Hall, 2005.

### *Bibliografía o documentación de lectura obligatoria\**

Resumen de cada uno de los temas (elaborado por el profesor).

#### **Códigos.-**

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

<sup>ii</sup> *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).

<sup>iii</sup> *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).

<sup>iv</sup> *CC: Criterios de Calificación* (ponderación del criterio de evaluación en la calificación cuantitativa final).

<sup>v</sup> *NR:* actividad "no recuperable" o que no permite evaluación extraordinari

(\*) Apartados no obligatorios

<sup>vi</sup> *NR:* actividad "no recuperable" o que no permite evaluación extraordinari

(\*) Apartados no obligatorios