

Plan Docente de una materia

“Laboratorio de Programación II”

I. Descripción y contextualización

<i>Identificación y características de la materia</i>				
<i>Denominación y código</i>	Laboratorio de Programación II			
<i>Curso y Titulación</i>	2º Curso. I.T. Informática de Sistemas			
<i>Área</i>	Lenguajes y Sistemas Informáticos.			
<i>Departamento</i>	Informática.			
<i>Tipo</i>	Obligatoria	Básico		
<i>Coefficientes</i>	Practicidad 4	Agrupamiento: 2		
<i>Duración ECTS (créditos)</i>	Segundo cuatrimestre		5.2 ECTS (130 h.)	
<i>Distribución ECTS (rangos)</i>	Grupo Grande: 10%	Seminario-Lab.: 30%	Tutoría ECTS: 5%	No presenciales: 55%
	13 horas	39 horas	6 horas	72 horas
<i>Descriptor (según BOE)</i>	Diseño, mantenimiento y desarrollo de programas. Técnicas de verificación y prueba de programas			
<i>Coordinador-Profesor/es</i>	(1) Halina Cwierz López			
<i>Tutorías complementarias (1)</i>	Despacho G-3	Ext. 2558	hccwierz@unex.es	
	Sujeta a horario			
<i>Tutorías complementarias (2)</i>				

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.

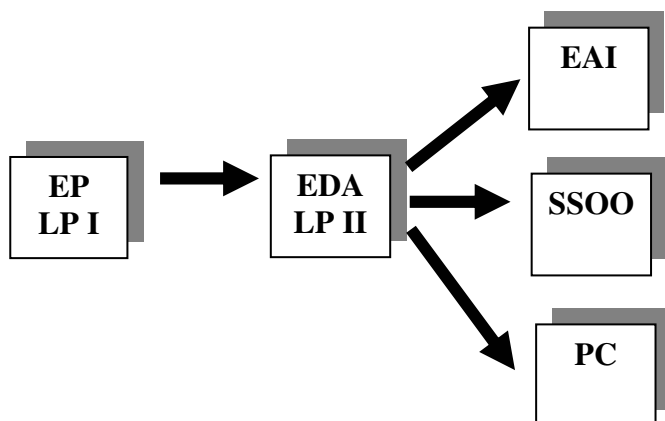
Contextualización curricular

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 II 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 II, como asignatura de segundo curso, segundo cuatrimestre, se apoya en Elementos de Programación, y Laboratorio de Programación I, asignaturas de primer curso, en la que el alumno toma por primera vez contacto con la programación. Asimismo, está estrechamente relacionada con la asignatura de segundo curso, Estructuras de Datos y Algoritmos. Con estas 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: 5, 7, 9, 14, 16, 17, 18, 21.

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 segundo curso sin haber superado o asistido a las asignaturas de Elementos de Programación y Laboratorio de Programación I. 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 solvente los problemas con los que se va encontrando.

II. Objetivos

<i>Relacionados con competencias académicas y disciplinares</i>		<i>Vinculación</i>
Descripción		<i>CET</i>
1.	Que el alumno sea capaz de diseñar programas empleando POO. Diseño e implementación de clases.	5, 7, 14, 21
2.	Que el alumno sea capaz de representar el diseño utilizado en una aplicación orientada a objetos, mediante un lenguaje de modelado.	5, 7, 14, 21
3.	Que el alumno sea capaz de abstraerse para adquirir los conocimientos necesarios para comprender y aplicar con fluidez las estructuras de datos adecuadas (dinámicas y estáticas).	5, 7, 14, 21
4.	Que el alumno sea capaz de resolver supuestos que supongan la implementación de programas de tamaño grande, con codificaciones correctas y eficientes, utilizando el paradigma orientado a objetos.	5, 7, 14, 21
5.	Que el alumno sea capaz de documentar (interna y externamente) de forma adecuada y correcta una aplicación propuesta.	5, 7, 9, 14, 21
6.	Preparar 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>CET</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. Programación orientada a objetos

1.- Conceptos de la POO

- 1.1. Introducción. Clases y objetos.
 - 1.1.1. Motivaciones y conceptos fundamentales.
 - 1.1.2. Programación Orientada a Objetos como metodología.
- 1.2. Definición de una clase.
 - 1.2.1. Instancias de clases.
 - 1.2.2. Acceso a miembros de la clase: encapsulación.
 - 1.2.3. Métodos y mensajes.
 - 1.2.4. Métodos en línea y fuera de línea.
 - 1.2.5. Archivos de cabecera y clases.
- 1.3. Constructores y destructores.
- 1.4. Punteros a objetos. Creación de objetos dinámicos.
 - 1.4.1. Reserva dinámica de memoria para objetos.
 - 1.4.2. Métodos constructores.
- 1.5. Asignación de objetos dinámicos. Atributos dinámicos.
- 1.6. Destrucción de objetos dinámicos.
 - 1.6.1. Liberación de memoria de objetos.
 - 1.6.2. Métodos destructores.
- 1.7. Referencias.
- 1.8. Aplicación en C++.
- 1.9. Problemas resueltos.
- 1.10. Problemas propuestos

2.- Herencia

- 2.1. Introducción. Concepto de herencia.
 - 2.1.1. Clase base y clase derivada.
 - 2.1.2. Consideraciones de diseño.
- 2.2. Tipos de herencia.
 - 2.2.1. Herencia simple.
 - 2.2.2. Herencia múltiple.
 - 2.2.3. Herencia pública.
 - 2.2.4. Herencia privada.
 - 2.2.5. Herencia protegida.
- 2.3. Constructores y destructores en herencia.
- 2.4. Aplicación en C++.
- 2.5. Problemas resueltos.
- 2.6. Problemas propuestos.

3.- Polimorfismo

- 3.1. Introducción. Concepto de polimorfismo.
 - 3.1.1. Punteros de tipos derivados.
- 3.2. Constructores y funciones virtuales.
- 3.3. Destructores y funciones virtuales. Destructor virtual puro.
- 3.3. Polimorfismo. Uso y ventajas.
- 3.4. Ligadura dinámica frente a ligadura estática.
- 3.5. Métodos virtuales puros y tipos abstractos.
- 3.6. Aplicación en C++.
- 3.7. Problemas resueltos.
- 3.8. Problemas propuestos.

Unidad temática II. Análisis orientado a Objetos.

4.- Diagramas de clases.

- 4.1. Identificación de clases: análisis gramatical.
- 4.2. Identificación de atributos.
- 4.3. Identificación de operaciones.

Unidad temática III. Manejo de errores.

5.-Excepciones.

- 5.1. Introducción. Conceptos generales.
- 5.2. La gestión de errores.
 - 5.2.1. Excepciones. Lanzamiento de excepciones.
 - 5.2.2. Manejo de excepciones.
 - 5.2.3. Captura de excepciones. Relanzar excepciones.
 - 5.2.4. Restricción de excepciones permitidas.
 - 5.2.5. Excepciones definidas por el usuario.
 - 5.2.6. Excepciones en constructores y destructores.
- 5.3. Aplicación en C++.
- 5.4. Problemas resueltos.
- 5.5. Problemas propuestos.

Unidad temática IV. Programación genérica.

6.- Sobrecarga

- 6.1. Introducción. Concepto de sobrecarga.
- 6.2. Sobrecarga de funciones.
- 6.3. Sobrecarga de métodos.
- 6.4. Sobrecarga de operadores.
 - 6.4.1. Sobrecarga de operadores unarios.
 - 6.4.2. Sobrecarga de operadores binarios.
 - 6.4.3. Sobrecarga del operador de asignación.
 - 6.4.4. Sobrecarga como métodos y como funciones amigas.
- 6.5. Aplicación en C++.
- 6.6. Problemas resueltos.
- 6.7. Problemas propuestos.

7.- Plantillas (Templates)

- 7.1. Introducción. Genericidad.
- 7.2. Fundamentos teóricos.
- 7.3. Plantillas de funciones.
 - 7.3.1. Definición de una plantilla de función.
 - 7.3.2. Ejemplos de plantilla de funciones.
- 7.4. Plantillas de clases.
 - 7.4.1. Definición de una plantilla de clases.
 - 7.4.2. Instanciación de una plantilla de clases.
- 7.5. Plantillas frente a polimorfismo.
- 7.6. Aplicación en C++.
- 7.7. Problemas resueltos.
- 7.8. Problemas propuestos.

8.- Biblioteca de plantillas estandar. (STL)

- 8.1. Introducción a la STL. Conceptos clave.
- 8.2. Contenedores e iteradores.
- 8.3. Contenedores y algoritmos.
- 8.4. Aplicación en C++.
- 8.5. Problemas resueltos.
- 8.6. Problemas propuestos.

Unidad temática V. Documentación de programas**9.- Introducción y Documentación interna.**

- 9.1. Introducción a la documentación de programas.
- 9.2. Documentación interna.
 - 9.2.1. Estilo de programación.

10.- Documentación externa.

- 10.1. Manual del Usuario.
 - 10.1.1. Descripción General del Sistema.
 - 10.1.2. Instalación del Sistema.
 - 10.1.3. Puesta en marcha del Sistema.
 - 10.1.4. El Entorno de Trabajo.
 - 10.1.5. Guía del Operador del Sistema.
 - 10.1.6. Lista de errores.
- 10.2. Manual del Programador.
 - 10.2.1. Descripción del Manual del Programador.
 - 10.2.2. Especificaciones lógicas del sistema.
 - 10.2.3. Estructuras de Datos.
 - 10.2.4. Algoritmos.
 - 10.2.5. Descripción por Módulos.
 - 10.2.6. Etapas de desarrollo.
 - 10.2.7. Diagramas de Clase.
 - 10.2.8. Diagramas de secuencia.
 - 10.2.9. Diagramas de actividad.
 - 10.2.10. Futuras modificaciones.
 - 10.2.11. Listados.
- 10.3. Problemas resueltos.
- 10.4. Problemas propuestos.

Interrelación

Requisitos (Rq) y redundancias (Rd)		Tema	Procedencia
Estructuras de Datos y Algoritmos	Rq	1 a 8	Estructuras de Datos y Algoritmos (2° Anual)
Programación imperativa	Rq	1 a 8	Elementos de Programación (1° Anual)
Conocimiento de Lenguaje C	Rq	1 a 8	Laboratorio de Programación I (1°)

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	0,5	1-10	Todos
2. Encuesta de conocimientos previos	GG	C-E	0,5	1-10	Todos
3. Lectura previa del tema	NP	T	1	1	Todos
4. Exposición general de los conceptos de POO	GG	T-P	2	1	Todos
5. Resolución de problemas cortos sobre el computador, enfocados a la resolución de la práctica propuesta.	S	T-P	3	1	Todos
6. Aplicación en C++ de supuestos prácticos de la actividad anterior	NP	P	3	1	Todos
7. Implementación de clases y su aplicación.	S	T-P	3	1	1,5,6 y 9
8. Asistencia a tutorías complementarias para aclaración de dudas.	Tut	T-C-E	1	1	Todos
9. Realización de problemas propuestos en las tutorías complementarias.	NP	P-C-E	2	1	1,5,6,7,8 y 9
10. Lectura previa del tema	NP	T	1	2	Todos
11. Exposición del concepto y tipos de Herencia	GC	T-P	1	2	1,5,6 y 9
12. Solución de problemas en el computador, enfocados a la práctica.	S	T-P	3	2	1,5,6 y 9
13. Aplicación en C++ de supuestos prácticos de la actividad anterior	NP	P	3	2	Todos
14. Asistencia a tutorías complementarias para aclaración de dudas.	Tut	P-C-E	1	2	Todos
15. Realización de problemas propuestos en las tutorías complementarias.	NP	P-C-E	1	2	Todos
16. Lectura previa del tema	NP	T	1	3	Todos
17. Exposición del concepto de Polimorfismo y su aplicación	GC	T-P	1	3	1,5,6 y 9
18. Resolución de problemas cortos sobre el computador, enfocados a la resolución de la práctica propuesta.	S	T-P	3	3	Todos
19. Aplicación en C++ de supuestos prácticos de la actividad anterior	NP	P	3	3	Todos
20. Asistencia a tutorías complementarias para aclaración de dudas.	Tut	P-C-E	1	3	Todos
21. Realización de problemas propuestos en las tutorías complementarias.	NP	P-C-E	1	3	Todos
22. Estudio y preparación del examen parcial práctico	NP	P-C-E	5	1-3	Todos
23. Realización del examen parcial práctico	S	P-C-E	3	1-3	Todos
24. Lectura previa del tema	NP	T	1	4	Todos
25. Identificación de clases y su representación mediante diagramas	GC	T-P	1	4	1,2,6 y 9
26. Resolución de problemas cortos sobre el computador, enfocados a la resolución de la práctica propuesta.	S	T-P	3	4	Todos
27. Aplicación en C++ de supuestos prácticos de la actividad anterior	NP	P	3	4	Todos
28. Estudio y preparación del examen parcial teórico	NP	T-C-E	2	1-4	Todos
29. Realización del examen parcial teórico	GC	T-C-E	1	1-4	Todos
30. Lectura previa del tema	NP	T	1	5	Todos
31. Gestión de errores, manejo de excepciones	GC	T-P	1	5	1,2,4,6 y 9
32. Resolución de problemas cortos sobre el computador, enfocados a la resolución de la práctica propuesta.	S	P	3	5	Todos
33. Aplicación en C++ de supuestos prácticos de la actividad anterior	NP	P	2	5	Todos
34. Asistencia a tutorías complementarias para aclaración de dudas.	Tut	P-C-E	1	5	Todos
35. Realización de problemas propuestos en las tutorías complementarias.	NP	P-C-E	1	5	Todos
36. Lectura previa del tema	NP		1	6	Todos
37. Exposición del concepto de sobrecarga y su aplicación	GC	T-P	1	6	1,2,4,6 y 9
38. Resolución de problemas cortos sobre el computador, enfocados a la resolución de la práctica propuesta.	S	P	3	6	Todos
39. Aplicación en C++ de supuestos prácticos de la actividad anterior	NP	P	3	6	Todos
40. Asistencia a tutorías complementarias para aclaración de dudas.	Tut	P-C-E	1	6	Todos
41. Realización de problemas propuestos en las tutorías complementarias.	NP	P-C-E	2	6	Todos
42. Lectura previa del tema	NP	T	1	7	Todos
43. Exposición del concepto de genericidad, plantillas de funciones y de clases	GC	T-P	1	7	1,2,3,4,6 y 9
44. Resolución de problemas cortos sobre el computador, enfocados a la resolución de la práctica propuesta.	S	P	3	7	Todos
45. Aplicación en C++ de supuestos prácticos de la actividad anterior	NP	P	3	7	Todos
46. Lectura previa del tema	NP		1	8	Todos
47. Exposición de la biblioteca de plantillas estándar.	GC	T-P	1	8	1,2,4,6 y 9

48.	Resolución de problemas cortos sobre el computador, enfocados a la resolución de la práctica propuesta.	S	P	3	8	Todos
49.	Aplicación en C++ de supuestos prácticos de la actividad anterior	NP	P	3	8	Todos
50.	Lectura previa del tema	NP		1	9	Todos
51.	Documentación interna de programas.	GC	T-P	1	9	Todos
52.	Corrección de la actividad anterior.	S	P	3	9	Todos
53.	Aplicación sobre los programas de la práctica propuesta.	NP	P	3	9	Todos
54.	Lectura previa del tema	NP	T	1	10	Todos
55.	Documentación externa de programas.	GC	T-P	1	10	Todos
56.	Corrección de la actividad anterior. Explicación y discusión en clase.	S	P	3	10	Todos
57.	Aplicación sobre los programas de la práctica propuesta.	NP	P	3	10	Todos
58.	Estudio y preparación del examen parcial práctico	NP	P-C-E	5	1-10	Todos
59.	Realización del examen parcial práctico	S	P-C-E	3	1-10	Todos
60.	Estudio y preparación del examen final.	NP	P-C-E	8	1-10	Todos
61.	Examen final	S	C-E (I)	3	1-10	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)	80	1	0	2	10
	Teóricas (II y III)	80	11	10	11	22
	Prácticas (IV, V y VI)	80	-	-	-	-
	Subtotal	80	12	10	13	32
Seminario- Laboratorio (6-20 alumnos)	Coordinac./evaluac. (I)	20	-	-	-	-
	Teóricas (II y III)	20	5	-	20	10
	Prácticas (IV, V y VI)	20	28	33	112	40
	Subtotal	20	33	33	132	50
Tutoría ECTS (1-5 alumnos)	Coordinac./evaluac. (I)	5	-	-	-	16
	Teóricas (II y III)	5	1	-	16	-
	Prácticas (IV, V y VI)	5	4	7	64	-
	Subtotal	5	5	7	80	16
Tutoría comp. y preparación de ex. (VII)		1	10	20	36	20
Totales			60 (2,4 ECTS)	70 (2,8 ECTS)	261	118

*Otras consideraciones metodológicas**

Recursos y metodología de trabajo en las actividades presenciales

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).

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.

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

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).

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.

V. Evaluación

<i>Criterios de evaluación*</i>		<i>Vinculación*</i>	
Descripción		<i>Objetivo</i>	<i>CCⁱⁱ</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 tipo test donde se evaluará los conceptos teóricos explicados a lo largo del curso.	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 de laboratorio.	Todos		5%

<i>Actividades e instrumentos de evaluación</i>		
Seminarios y Tutorías ECTS	<ul style="list-style-type: none"> • Valoración de la asistencia y participación en las actividades teóricas y prácticas. • Registro y valoración de las actividades prácticas entregadas por el alumno. 	10% (NR) ^v
Exámenes parciales	<ul style="list-style-type: none"> • Examen teórico tipo test. • 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. • 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. 	10% (NR) ^v 80%
Examen final	<ul style="list-style-type: none"> • Modificación a realizar en la práctica propuesta. 	80%

VI. Bibliografía

Bibliografía de apoyo seleccionada

Cómo programar en C++. Segunda edición.
H. Deitel & P. Deitel. Prentice Hall, 1999

Estructura de Datos, Algoritmos y Programación Orientada a Objetos.
Gregory L. Heileman. Ed. McGraw-Hill, 1998

Fundamentals of Data Structures in C++.
E. Horowitz, S. Sahni, D. Mehta. Computer Science Press, 1997

Estructura de Datos. Algoritmos, Abstracción y Objetos.
Luis Joyanes Aguilar, Ignacio Zahonero Martínez. Ed. McGraw-Hill, 1998

Estructura de Datos. Libro de Problemas
Luis Joyanes Aguilar, Ignacio Zahonero Martínez, Matilde Fernández Azuela, Lucas Sánchez García. Ed. McGraw-Hill, 1999

Programación en C++. Algoritmos, Estructura de Datos y Objetos.
Luis Joyanes Aguilar. Ed. McGraw-Hill, 2000

Borland C++. Manual de referencia.
H. Schildt. McGraw Hill, 1999

El lenguaje de Programación C++.
Bjarne Stroustrup. Addison-Wesley, 1998

*Bibliografía o documentación de ampliación, sitios web...**

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

Códigos.-

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

ⁱⁱ *CC*: Criterios de Calificación (ponderación del criterio de evaluación en la calificación cuantitativa final).