Modelo de Plan Docente de una materia

I. Descripción y contextualización

Identificación y características de la materia				
Denominación y código	Laboratorio de Programación			
Curso y Titulación	**1º de Ingeniería Informática**			
Área	Lenguajes y Sistemas Informáticos			
Departamento	*Informática*			
Tipo	*OB, OP, TR, LE		Troncal	
Coeficientes	Practicidad:4 (Medio-alto)		Agrupamiento: Medio-Bajo	
Duración ECTS (créditos)	**6 ECTS (150 horas)**		**Segundo Cuatrimestre**	
Distribución ECTS (rangos)	Grupo Grande:	Seminario-Lab.:	Tutoría ECTS:	No presenciales:
	15 horas	45 horas	0	90
Descriptores (según BOE)	Diseño de Algoritmos. Lenguajes de Programación. Diseño de Programas: Descomposición Modular y Documentación. Pruebas de Programas. Tipos Abstractos de Datos. Estructura de Datos y Algoritmos de Manipulación. Según Libro Blanco: Fundamentos y metodología de la programación Lenguajes de programación			
Coordinador-Profesor/es	**Pedro José Clemente Martín (coordinador)** **Juan Antonio Gil Prieto** **Maria de los Ángeles Mariscal Araújo** **Juan Carlos Preciado Rodríguez**			
Tutorías complementarias (1)	Despacho 10 (ext. 7806)	Pabellón de Informática	mariscal@unex.es	
Tutorías complementarias (2)	Despacho 11 (ext. 7807)	Pabellón de Informática	pjclemente@unex.es	
Tutorías complementarias (3)	Despacho 26 (ext. 7258)	Pabellón de Informática	jcpreciado@unex.es	
Tutorías complementarias (4)	Despacho 01 (ext. 7251)	Pabellón de Informática	jgil@infoibox.com	

Contextualización profesional

Conexión con los perfiles profesionales de la Titulación

Dentro de los Contenidos formativos comunes de la titulación de Ingeniería Informática se establecen con contenidos específicos del título. En estos contenidos se circunscribe la subcategoría Programación, en cuyo estudio se centra esta materia. En el libro blanco esta subcategoría debe tratar:

- Fundamentos y metodología de la programación
- Algoritmia
- Computabilidad
- Lenguajes de Programación
- Paradigmas de Programación
- Estructuras de datos

Son competencias fundamentales de esta asignatura:

- Fundamentos y metodología de la programación

Esta materia debe ser tratada independientemente del perfil profesional que pretenda alcanzarse, al ser materia de primer curso debe los objetivos fundamentales son que el alumno sepa analizar, diseñar e implementar pequeños programas, iniciándolo en la obtención de una visión crítica que deberá desarrollar posteriormente. También le dotaremos de las herramientas necesarias para la construcción de los programas.

En esta asignatura se acerca al alumno el desarrollo de software utilizando el paradigma orientado a objetos. En el proceso de desarrollo de software se seguirán las siguientes fases: análisis, diseño, implementación, pruebas y documentación de programas.

Esta materia debe ser tratada independientemente del perfil profesional que pretenda alcanzarse, al ser una materia fundamental dentro de la programación, en el que se profundizan los conocimientos y competencias que empezaron a desarrollarse en el primer curso, ampliando su estudio con algoritmos y estructuras de datos más complejos, lo que permitirá construir programas más grandes y más útiles.

Cualquier perfil profesional que desarrolle el estudiante posteriormente se verá influido por su conocimiento de las técnicas básicas y avanzadas de programación y su capacidad para analizar y resolver problemas, presentar y valorar alternativas, competencias desarrolladas ampliamente en esta asignatura.

Contextualización curricular

Esta es una asignatura anual de primer curso de introducción a la programación, por tanto se trata de una herramienta básica en la formación de cualquier titulado en Informática, independiente del perfil profesional que el alumno seleccione posteriormente. Por tanto, esta asignatura sirve como base de múltiples competencias tanto específicas como transversales de la titulación, así como base y requisito para muchas de las asignaturas impartidas posteriormente.

Esta asignatura está relacionada de forma directa con la asignatura de Elementos de Programación (1 curso, 1 cuatrimestre). Asignatura fundamental dentro de los contenidos específicos de la Ingeniería Informática.

También se relacionan con otras de las asignaturas restantes que forman parte de los contenidos específicos de la Ingeniería Informática (arquitectura de ordenadores, matemáticas, etc.) , ya que en casi todas se usa la programación de ordenadores para conseguir los objetivos formativos propuestos.

*Contextualización personal**

Al tratarse de una materia de primer curso los alumnos proceden de niveles inferiores del sistema educativo actual, fundamentalmente de bachillerato tecnológico y de ciclos formativos superiores (FP-II).

En algunas ocasiones nos hemos encontrado con alumnos procedentes de otros itinerarios académicos, que muestran grandes deficiencias en cuestiones técnicas, lo que les impiden el seguimiento de las clases.

De forma general, los alumnos muestran una gran deficiencia en conocimientos elementales relacionados con las matemáticas, y nos gustaría destacar el bajo nivel que demuestran en la expresión escrita y oral. Ambas consideradas esenciales para un futuro Ingeniero en Informática. Sin embargo, bien es cierto que en los últimos años, las capacidades relacionadas con la informática a nivel de usuario ya están desarrolladas, lo que facilita el trabajo con nuestros alumnos.

Sería necesario que éstos alumnos, de primer curso, tuvieran más información acerca de la titulación en la que se encuentran, puesto que rara vez sus expectativas corresponden con el objetivo final de la misma. Es indispensable que los alumnos tengan una guía detallada de la universidad, de la titulación y de cada una de las materias, con sus objetivos, antes de realizar la matrícula. Pues hemos observado que la mayor dificultad es causada por su desorientación.

II. Objetivos

Relacionados con competencias académicas y disciplinares	*Vinculación*
1. Capacitar al alumno para llevar a cabo el desarrollo de software orientado a objetos siguiendo las fases de análisis, diseño, codificación, prueba, depuración y documentación de programas, utilizando una correcta metodología y estilo basado en el paradigma orientado a objetos.	2,11
2. Ser capaz de escribir aplicaciones de tamaño medio, correctas, eficientes, bien organizadas y bien documentadas.	2,11,10, 14,21
3. Capacitar al alumno para el correcto uso de las estructuras de datos y algoritmos básicos, así como su aplicación al diseño de programas.	2,11,14,21
4. Capacitar al alumno para afrontar con éxito las necesidades de programación que se presenten en asignaturas posteriores.	2,11,14,21
5. Desarrollar prácticas encaminadas a profundizar en la especificación, implementación y uso de TADs.	2,11
6. Enseñar los conceptos introductorios sobre análisis y diseño de una metodología de modelado orientada a objetos.	2,10,
7. Motivar y enseñar al alumno sobre la necesidad de documentar y probar adecuadamente el software desarrollado.	2, 14, 35
8. Enseñar el manejo de herramientas de desarrollo de software y de utilidades para depuración de programas.	10,12
9. Enseñar al alumno a escribir aplicaciones correctas, eficientes, bien organizadas y bien documentadas en un lenguaje de programación de alto nivel.	2,10
10. Capacitar al alumno para desarrollar aplicaciones que manejen de forma eficiente colecciones de datos a través de la especificación, implementación y uso de estructuras de datos secundarias (ficheros)	2,10

III. Contenidos

*Selección y estructuración de conocimientos generales**

SECUENCIACIÓN DE BLOQUES TEMÁTICOS Y TEMAS
Tema 1: Programación orientada a objetos
1.1. Conceptos básicos 1.2. Principios fundamentales de POO 1.3. Especificación, implementación y uso de clases en pseudocódigo 1.4. Representación en C++ 1.5. Comparación y Copia de Objetos. Sobrecarga de Operadores
Tema 2: Análisis y Diseño de Sistemas Orientado a Objetos
2.1.Introducción al proceso de desarrollo software 2.2. Análisis y diseño 2.2.1. Introducción 2.2.2. Notación de modelado UML 2.2.4. Diagramas de clases 2.2.5. Diagrama de interacción 2.2.6. Diagramas de estado y de actividad 2.3. Tareas en el análisis 2.3.1. Modelado conceptual de datos 2.3.2. Modelado de comportamiento 2.3.3. Modelado de estados 2.4. Tareas en el diseño 2.4.1. Clases de diseño 2.4.2. Diagramas de secuencia 2.4.3. Contratos de las operaciones 2.5. Más fases (siguientes temas) 2.5.1. Codificación 2.5.2. Pruebas 2.6. Documentación de Programas (siguientes temas)
Tema 3: Pruebas de Programas
3.1. Introducción 3.2. Técnicas de prueba 3.3. Estrategias de prueba sistemática 3.4. Depuración
Tema 4: Estructuras de almacenamiento secundario
4.1. Introducción. Concepto de flujo
4.2. Ficheros en C++ 4.2.1. Apertura y cierre 4.2.2. Detección del fin de fichero 4.2.3. Control de errores
4.3. Ficheros de texto

4.3.1. Lectura y escritura 4.3.2. Avance de cursor 4.4. Ficheros Binarios 4.4.1. Lectura y escritura por bloques de bytes 4.4.2. Acceso aleatorio a ficheros 4.5. Ordenación y búsqueda en ficheros 4.6. Ejemplos y ejercicios.
Tema 5: Documentación de programas
5.1. Estilo de programación 5.2. Documentación interna 5.3. Documentación externa

Interrelación			
Requisitos (Rq) y redundancias (Rd)		Tema	*Procedencia*
Programación Imperativa	Rq	1, 2, 3, 4, 5	Elementos de Programación
Recursividad	Rq	5	Elementos de Programación
Análisis de algoritmos	Rq	7	Elementos de Programación
Tipos Abstractos de Datos. Uso de TAD lineales	Rq	7, 9, 10	Elementos de Programación
Clases de equivalencia	Rq	3	Algebra

En general, todos los temas tratados en las la asignatura de Elementos de Programación se utilizan y aplican directamente en esta asignatura.

Además, en muchos casos, se revisan de nuevo conceptos ya vistos ampliándolos, especificándolos con mayor formalismo, añadiendo más posibilidades, etc. Por eso, aunque en algunos casos pueda parecer que hay redundancia en los contenidos, realmente no se tratan de la misma forma, sino que se amplia la perspectiva.

IV. Metodología docente y plan de trabajo del estudiante

Actividades de enseñanza-aprendizaje				***Vinculación***		
Descripción y secuenciación de actividades	*Tipo*			*Duración*	*Tema*	*Objetivos*
	p/np	*grupo*	*Carácter*			
1. Presentación del Plan docente de la asignatura	P	GG	C-E	0,6	15-feb	
2. Encuesta sobre la procedencia de los alumnos y sus intereses	P	GG	C-E	0,4	15-feb	
Tema 1. Programación orientada objetos						
3. Lectura previa del resumen del tema	NP	GG	T	0,5	1	1
4. Exposición del detallada del tema	P	GG	T	3	1	1,2
5. Aplicación mediante ejercicios, tutorizados en el laboratorio, del contenido presentado	P	S	P	12	1	1,11
6. Desarrollo de ejercicios prácticos, e inicio del desarrollo del proyecto de programación que el alumno debe desarrollar	NP	S	T/P	15	1	1,4, 8, 9
Tema 2. Análisis y diseño orientado a objetos						
7. Lectura previa del resumen del tema	NP	GG	T	0,5	2	1,4, 6
8. Explicación, discusión y ejemplificación en clase	P	GG	T	0,5	2	1,4, 6
9. Resolución de problemas en clase	P	GG	T	1,5	2	6
10. ADOO aplicado a un proyecto de programación de años anteriores	P	S	P	3	2	1,4, 6, 8
11. Aplicación mediante ejercicios del contenido presentado, enlazando con el tema anterior	P	S	P	6	2	1,4, 6, 8
12. Resolución de un proyecto de programación donde integrar los conocimientos adquiridos. Resolución Fase 1	NP	S	T/P	15	1-2	1,4, 6, 8
Tema 3. Pruebas de programas						
13. Lectura previa del resumen del tema	NP	GG	T	0,5	3	1,4,6,7
14. Exposición de las técnicas de pruebas de programas	P	GG	T	1	3	1,4,6,7
15. Resolución de problemas sobre pruebas	P	S	P	3	3	1,4,6,7,8
16. Resolución de una práctica donde integrar los conocimientos adquiridos. Resolución Fase 2.	NP	T/P	T/P	15	1-2-3	1,4,6,7, 8
Tema 4. Estructuras de datos de almacenamiento secundario (Ficheros)						
17. Lectura previa del resumen del tema	NP	CG	T	0,5	4	1, 2, 3, 4, 5, 8, 9, 10
18. Descripción y clasificación básicas de las estructuras de almacenamiento secundario (ficheros)	P	CG	T	5	4	3,5,10
19. Resolución de ejercicios correspondientes al tema	P	S	P	18	4	3,5,10
20. Resolución de problemas	NP	T/P	P	5	4	Todos
21. Búsqueda y desarrollo de un Wiki con algoritmos de fusión, mezcla y ordenación de ficheros	NP	NP	T/P	3	5	
22. Resolución de una práctica donde integrar los conocimientos adquiridos .Realización Fase 3 del proyecto de programación	NP	T/P	T/P	12	1-2-3-4	Todos
Tema 5. Documentación de programas						
23. Lectura previa del resumen del tema	NP	GG	T	0,5	5	1, 7, 4, 6, 9
24. Exposición de las técnicas de documentación de programas	P	GG	T	1	5	1, 7, 9
25. Documentación externa del proyecto de programación que se ha desarrollado en fases durante el curso	NP	T/P	T/P	10	Todos	1,7, 4, 6, 9
Examen final						
26. Estudio y preparación del examen final	NP	I	C-E	15		Todos
27. Examen final teoría	P	GG	C-E	2		Todos
28. Examen final práctica	P	S	C-E	3		Todos
				150		

En la distribución del tiempo para el alumno se han tenido en cuenta las actividades marcadas en la tabla anterior.
Para el cálculo del tiempo que dedica el profesor, se ha estimado un grupo de 60 personas para los grupos grandes, GG, de 20 para los seminarios y de 5 personas para las tutorías.
Para el cálculo de la carga del profesor en horas no presenciales, se ha seguido con los parámetros expuestos en la guía extensa proporcionada por el Vicerrectorado:

- Revisión del plan docente a principio de curso : 2 horas
- Revisar y prepara los resúmenes y otros materiales de prácticas y apoyo: 3 horas
- Completar la evaluación de los trabajos dirigidos o grupos autorizados: 30 minutos por grupo.
- Corregir exámenes: 1 hora por alumno, teniendo en cuenta que se realiza un examen final, 2 exámenes parciales y se debe corregir el proyecto de programación desarrollado.

Dos sesiones de revisión de resultado y elaboración de las actas: 2 horas

- Clases de teoría: 30 minutos cada una
- Clases de prácticas: 45 minutos cada una, no repetida.

Como resultado podemos observar que para un grupo de 60 alumnos, en esta asignatura de 6 créditos ECTS distribuidos en (1,5 de Teoría y 4,5 de Prácticas), necesitamos que un profesor tenga una dedicación de 304 horas. Si tenemos en cuenta que la asignatura es cuatrimestral, suponemos entonces que son 20 las semanas, tenemos una dedicación de 15,2 horas/semana.

Distribución del tiempo (ECTS)			*Dedicación del alumno*		*Dedicación del profesor*	
Distribución de actividades		*Nº alumnos*	*H. presenciales*	*H. no presenc.*	*H. presenciales*	*H. no presenc.*
Grupo grande (Más de 20 alumnos)	Coordinac./evaluac. (I)	60	1		1	19
	Teóricas (II y III)	60	14	35	14	7
	Prácticas (IV, V y VI)	60	0			0
	Subtotal	**60**	**15**	**35**	**15**	**26**
Seminario-Laboratorio (6-20 alumnos)	Coordinac./evaluac. (I)	20	3	0	9	45
	Teóricas (II y III)	20	0	0	0	0
	Prácticas (IV, V y VI)	20	42	55	126	63
	Subtotal	**20**	**45**	**55**	**135**	**108**
Tutoría ECTS (1-5 alumnos)	Coordinac./evaluac. (I)	5				
	Teóricas (II y III)	5				
	Prácticas (IV, V y VI)	5				
	Subtotal	**5**				
Tutoría comp. y preparación de ex. (VII)		1				20
Totales			**60**	**90**	**150**	**154**

*Otras consideraciones metodológicas**

Recursos y metodología de trabajo en las actividades presenciales

Grupo grande:
Ante el número de alumnos matriculados en la titulación, hasta ahora muy elevados, y debido a los pocos recursos con los que cuentan los centros en los que se imparte, no es posible prescindir de la clase magistral como metodología de trabajo.

Buscando que la interacción con los alumnos, y posibilitar el debate dentro de las clases, se ofrecerá a los alumnos, antes de comenzar cada tema un resumen del mismo. Estos resúmenes deberán ser leídos por el alumno antes de ir a clase, lo que permitirá el debate en el aula, y que las clases puedan centrarse en los conceptos claves.

Al tratarse de una asignatura de primer curso, se tratarán conceptos básicos que deberán ser asimilados para poder cursar adecuadamente el resto de asignaturas de cursos posteriores de la titulación.

La clase expositiva no es suficiente en esta materia. La reflexión, la capacidad de creación y la de aportar ideas son características fundamental y propias de la misma, por ello la clase de desarrollará en un entorno de continuo debate, de preguntas y respuestas, donde es necesario una participación activa del alumno, contestando a las preguntas formuladas o bien resolviendo los problemas

propuestos.

En un grupo grande es fácil que los alumnos eviten la participación activa, por lo que en las actividades dedicadas a la resolución de problemas, se pretende la creación de grupos de trabajo, (de tres o cuatro personas), para que una vez que hayan resuelto problemas propuestos, un portavoz pueda mostrarlo en clase.

Seminarios:
Los seminarios se desarrollarán en grupos de 20 personas en el aula de informática. Los alumnos se agruparán en equipos de dos personas, que se mantendrá durante todo el cuatrimestre, fomentando así el trabajo en grupo.

Se aplicará una metodología de *Aprendizaje Basado en Problemas*, fomentando así una de las capacidades que deben adquirirse: el análisis y la proposición de una solución.

Es imprescindible que estos trabajos se apoyen en un trabajo realizado de forma no presencial, y que será propuesto por el profesor y resulto parcialmente antes de llegar al aula.

Es importante que el profesor dirija al alumno para ayudarle a encontrar la solución más óptima, pero a medida que el curso avanza, la intervención del profesor será menor, buscando la creatividad y la autosuficiencia del alumno a la hora de proporcionar soluciones.

El trabajo desarrollado en las aulas se apoyará del desarrollo parcial de un proyecto de programación, en el que se le propondrá al alumno una práctica que deberá desarrollar en tres fases. Cada una de las fases deberá ser integrada con lo hecho anteriormente, además de documentada.

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

Toda la metodología propuesta se basa en un aprendizaje mixto semipresencial, donde la asistencia a las clases de grupo grande, y a los seminarios es obligatoria, pero ambas deben apoyarse en el trabajo no presencial realizado por el alumno, antes y después de ir al aula.

Además del estudio individual, se proponen al alumno una serie de actividades que les ayuden a aumentar su participación y a fomentar el estudio activo y continuo durante toda la duración del curso. Entre estas actividades el alumno contará con un sistema de autoevaluación con cuestionarios on-line sobre cada uno de los temas tratados, que permitirán al alumno comprobar si los conocimientos obtenidos.

Se habilitará una página web en la que se incluirán todas las referencias bibliográficas, en principio básicas, y que los alumnos irán incluyendo.

Para que los alumnos se sientan motivados, utilizaremos una herramienta web que permita la elaboración de baterías de preguntas tipo test, que pueden ser utilizadas por los alumnos como pruebas de autoevaluación. Los resultados pueden ser almacenados o no, pero su mayor ventaja radica en la posibilidad de que los alumnos obtengan una información continuada sobre su nivel de conocimientos.

Recursos y metodología de trabajo para los alumnos que no han alcanzado los requisitos

Puesto que toda la metodología propuesta se basa en un aprendizaje mixto semipresencial y cualquier tipo de asistencia debe apoyarse en el trabajo no presencial realizado por el alumno, antes y después de ir al aula en este caso se establecerán mecanismos globales y concretos para los alumnos que no han alcanzado los requisitos. En este se sentido se replanteará el trabajo propuesto en clase para adaptarlo al tiempo y circunstancia necesaria con el fin de que el alumno alcance en un periodo de recuperación los requisitos establecidos.

En este caso también se dispondrá de las actividades de fomento de estudio y de un sistema de autoevaluación mediante cuestionarios sobre cada uno de los temas tratados, que permitirán al alumno comprobar si los conocimientos obtenidos.

Con el fin de plantear un sistema de evaluación similar, de nuevo y para que los alumnos se sientan motivados, utilizaremos una herramienta diseñada por algunos de los profesores implicados en la asignatura, denominada SECSI “Sistema de Evaluación Continua y Seguimiento por Internet”, que permite la elaboración de baterías de preguntas tipo test, que pueden ser utilizadas por los alumnos como pruebas de autoevaluación. Los resultados pueden ser almacenados o no, pero su mayor ventaja radica en la posibilidad de que los alumnos obtengan una información continuada sobre su nivel de conocimientos.

Recursos y metodología de trabajo para desarrollar competencias transversales

El mecanismo de propuesta de prácticas y ejercicios complejos se basa en abstraer del mundo real problemas que se dan en la actualidad del desarrollo de software a nivel empresarial, salvando naturalmente las diferencias y a una escala mucho menor. En este sentido podemos plantear problemas en escala reducida de los que los alumnos pueden encontrar hoy en día en el desarrollo de software y además dirigir la resolución de dichos problemas en la misma dirección.

La metodología de trabajo se basa en establecer equipos ficticios de desarrollo donde uno de los integrantes de dicho equipo será el responsable de que el trabajo encomendado a dicho equipo llegue a buen puerto. Este responsable de equipo será además quién trate con el profesor las dudas y problemas planteados en el desarrollo de cada trabajo. El liderazgo de cada equipo recaerá en cada ejercicio en un integrante diferente por cada trabajo propuesto.

V. Evaluación

*Criterios de evaluación**	*Vinculación**	
Descripción	*Objetivo*	*CC*[i]
Capacidad de escribir ordenada y claramente sus ideas Capacidad para analizar adecuadamente el enunciado de un proyecto	1,2,3,8	15% 40%
Ser capaz de analizar y diseñar una correcta solución basada en POO	1,3 4,5, 6,7,9,10	
Capacidad de evaluación y emisión de juicios de valor sobre algoritmos propuestos y escritos	4,6	10%
Habilidad para manejar el software de desarrollo de aplicaciones propuesto	8,7	10%
Ser capaz de entender software escrito previamente y adaptarlo a sus necesidades	6,7, 10	5%
Aplicar adecuadamente los algoritmos básicos para la manipulación de estructuras de datos y ficheros	11	20%
		100%

Actividades e instrumentos de evaluación		
Grupo Grande Seminarios, y no presencial	• [Participación en resolución y presentación de problemas en clase] • Participación en las actividades on-line (foro, wiki, cuestionarios, etc.) • Correcciones de problemas • Propuesta de problemas, preguntas de test, etc. • Búsqueda de documentación. • Participación activa en clase, etc.	*10%* *NR*
Proyecto	• Elaboración de un proyecto de programación en tres fases, cada una de ellas independientes y que deberá ser analizada y desarrollada, y posteriormente integrada. Esta tarea incluye buena parte del contenido impartido en la asignatura, incluyendo aspectos como el análisis y diseño del problema, implementación, documentación y pruebas • Un examen práctico donde el alumno demostrará las habilidades adquiridas	*50%*
Seminarios	• Trabajos tutorizados durante los seminarios	*15* *NR*
Examen final	• Se tratarán en este examen los contenidos presentados en la asignatura:	*25%*
		100%+ (5%)

VI. Bibliografía

Bibliografía de apoyo seleccionada

- Walter Savitch , /Resolución de problemas con C++. 2ª edición. /Prentice Hall, 2000
- Luis Joyanes, /Fundamentos de programación. Algoritmos, estructuras de datos y objetos. 3ª edición. / McGraw-Hill, 2003
- Luis Joyanes, /Fundamentos de programación. Libro de problemas. 2ª edición. /McGraw-Hill, 2003
- H.M. Deitel y P.J. Deitel, /Como programar en C/C++. 4ª ed./ Prentice Hall, 2003
- H. Schildt, /C++ Manual de referencia/. McGraw-Hill, 95.
- J. Castro y otros, /Curso de programación/. McGraw-Hill, 93
- Luis Joyanes, /Programación en C++. Algoritmos, estructuras de datos y objetos. /McGraw-Hill, 2000
- Pedro José Clemente y Julia González. Metodología de programación: enfoque práctico. Manuales de la Uex Nº 38. Universidad de Extremadura. Servicios de Publicaciones. ISBN 84-7723-652-6, 2005.

*Bibliografía o documentación de lectura obligatoria**

Documentación de las prácticas elaborada por los profesores de la asignatura http://epcc.unex.es dentro de la titulación de Ingeniería de Informática asignatura Laboratorio de Programación, en la sección de descarga.

Noticias interesantes relacionadas con las asignaturas y anunciadas en la web.

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

Referencias bibliográficas actualizadas para cada tema, sitios web, etc, aparecerán en la web de la asignatura.

---

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

[v] *NR*: actividad "no recuperable" o que no permite evaluación extraordinaria.

(*) Apartados no obligatorios.