

1.- Datos de la Asignatura

Titulación	Ingeniería Técnica Informática de Gestión				
Centro	Escuela Politécnica Superior de Zamora				
Denominación	Ingeniería del Software II			Código	16915
Plan	2003		Ciclo		Curso 3º
Carácter ¹	T		Periodicidad ²	C1	
Créditos LRU	T	3	P	3	Créditos ECTS 6
Área	Lenguajes y Sistemas Informáticos				
Departamento	Informática y Automática				
Aula / Horario / grupo	P-117		Martes: 16:00h – 17:00h Miércoles: 16:00h – 17:00h	Grupo único.	
Laboratorio/ Horario / grupo					
Informática / Horario / grupo	Aula Informática 1		Miércoles: 09:30h - 11:30h Jueves: 12:30h - 14:30h		
Plataforma Virtual	Plataforma Eudored de la Universidad de Salamanca, basada en Moodle.				
	URL de acceso: http://eudored.usal.es/				

¹ Troncal, Obligatoria, Optativa (abreviatura T, B, O)

² Anual, 1º Cuatrimestre, 2º Cuatrimestre (A, C1, C2).

Datos del profesorado*

Profesor Responsable / Coordinador	Jaime Calvo Gallego			
Departamento	Informática y Automática			
Área	Lenguajes y Sistemas Informáticos			
Centro	Escuela Politécnica Superior de Zamora			
Despacho	Edif. Admin. 208	Grupo / s	Todos	
Horario de tutorías	Publicado en tablones			
URL Web	http://web.usal.es/~jaime.calvo			
E-mail	jaime.calvo@usal.es	Teléfono	(+34) 980545000, ext. 3636	

Profesor
Departamento
Área

Centro	-----		
Despacho	-----	Grupo / s	-----
Horario de tutorías	-----		
URL Web	-----		
E-mail	-----	Teléfono	-----

* Caso de que sea una asignatura impartida por más de un docente.

*Esta tabla se repetirá tantas veces como sea necesario, en el caso de que sean varios docentes los responsables de impartir la materia, dedicando una tabla para cada docente.

2.- Sentido de la materia en el plan de estudios*

Bloque formativo al que pertenece la materia
<p>La asignatura de Ingeniería del Software II se imparte en el primer cuatrimestre del tercer curso de la Ingeniería Técnica en Informática de Gestión, cuyo Plan de Estudios actual data de 2003. Esta asignatura se relaciona directamente con otras de este mismo Plan de Estudios, pero además con otras de la titulación de segundo ciclo Ingeniería en Informática.</p> <p>La asignatura Ingeniería del Software II está directamente vinculada con todas las asignaturas troncales y obligatorias tanto de los dos cursos precedentes como del curso en el cual se encuentra inmersa.</p>
Papel de la asignatura dentro del Bloque formativo y del Plan de Estudios.
<p>La asignatura Ingeniería del Software II está directamente vinculada con todas las asignaturas troncales y obligatorias tanto de los dos cursos precedentes como del curso en el cual se encuentra inmersa, cumpliendo un papel de continuación del plan de formación dentro del bloque formativo y del plan de estudios al que pertenece.</p>
Perfil profesional.
<p>Para un Ingeniero Técnico en Informática de Gestión es imprescindible tener unos conocimientos robustos de cómo desarrollar un sistema software, teniendo en cuenta todas las fases de su ciclo de vida, desde que surge la necesidad de realizar dicho sistema hasta su implementación.</p> <p>La Ingeniería del Software II como asignatura dentro del Plan de Estudios actual de la Ingeniería Técnica en Informática de Gestión, es troncal y consta de 6 créditos LRU (3 teóricos y 3 prácticos). Su cometido es presentar las actividades técnicas e ingenieriles que se llevan a cabo en el ciclo de vida de un producto software. Además, describe los problemas, métodos y tecnologías asociadas con la Ingeniería del Software, en concreto con el desarrollo de proyectos de software y específicamente, con las partes de gestión, control de la calidad, y evolución y mantenimiento del software, entre otras. Por otra parte, se debe afianzar al alumno en la sistematización de la creación del software mediante el uso de métodos de ingeniería del software, tanto desde el punto de vista estructurado, típicamente tomando el método de Yourdon como referencia principal, como desde un punto de vista orientado a objetos, tomando UML (Unified Modeling Language) y el Proceso Unificado como bases fundamentales. En esta asignatura es demasiado ambicioso afrontar en profundidad los métodos de análisis y diseño, por ello se enfoca mayormente desde el punto de vista del uso de métodos de ingeniería del software y el desarrollo de proyectos software; partiendo de la base adquirida en la asignatura de Ingeniería del Software I. Para la parte práctica de esta asignatura se debe hacer uso de herramientas CASE (Computer Aided/Assisted Software Engineering).</p>

Este planteamiento y concretamente los tópicos que en esta asignatura se desarrollan son fundamentales para la formación de cualquier ingeniero informática, sea cual sea su perfil académico y su futuro perfil profesional.

**Esta información se puede obtener, en la mayoría de los casos, en los libros blancos de la ANECA para cada titulación. http://www.aneca.es/modal_eval/conver_docs/titulos.html.*

3.- Recomendaciones previas*

Para cursar con garantías de éxito la asignatura, es requisito previo el haber cursado y aprobado todas las asignaturas troncales de primer curso y de segundo curso.
De la misma forma, es requisito recomendable el haber cursado y aprobado todas las asignaturas obligatorias de los cursos precedentes.

** Requisitos previos o mínimos que en algunas materias son necesarios para cursar la asignatura (asignaturas previas, conocimientos concretos, habilidades y destrezas determinadas,...)*

Datos Metodológicos

4.- Objetivos de la asignatura (Generales y Específicos)

Objetivos Generales:

- Capacidad de integrar los conocimientos y destrezas prácticas de las diferentes asignaturas del plan de estudio para resolver situaciones reales relacionadas con la Ingeniería Técnica Informática de Gestión, así como con otras disciplinas relacionadas.
- Reforzar el hábito de plantearse interrogantes. Ante un problema preguntarse por el número de soluciones, la relación entre ellas, cómo afectaría a las condiciones iniciales alguna modificación, etc.
- Capacidad de aplicar y relacionar, de forma autónoma, los contenidos de Ingeniería del Software de forma interdisciplinar.
- Adquirir una comprensión del método científico, a través de las diversas actividades realizadas en la asignatura de Ingeniería del Software II, y asimilar su importancia como manera de pensar y actuar en la labor de científico e ingeniero, fomentando la capacidad de abstracción y el espíritu crítico.
- Desarrollar la madurez necesaria en el proceso de abstracción para abordar problemas reales y plantear modelos y soluciones de forma razonada y correcta.
- Reforzar el hábito de desarrollar diferentes alternativas, cuestionando las características, riesgos y viabilidad de cada una, para cada problema planteado.
- Destrezas para la participación responsable: capacidad de coordinación, asistencia, contribuciones al grupo, etc.
- Capacidad de trabajar en equipo adquiriendo y mejorando las habilidades sociales y la inteligencia emocional.

- Comprometerse de forma ética con el trabajo, con el resto de los integrantes del grupo y consigo mismo.

Objetivos Específicos:

- Aplicar los conocimientos adquiridos mediante la resolución de problemas y/o una práctica obligatoria.
- Utilizar con fluidez herramientas de Ingeniería del Software.
- Adquirir y emplear un buen lenguaje formal, tanto oral como escrito, siendo riguroso en las explicaciones de cualquier proceso.
- Reforzar el conocimiento y utilización de la terminología utilizada en Ingeniería del Software.
- Adquirir un buen manejo de la bibliografía recomendada en la asignatura, de forma que se potencia la autosuficiencia a la hora de completar la formación.
- Comprender el ámbito de la Ingeniería del Software dentro de la Ingeniería Técnica Informática de Gestión y dentro de los perfiles profesionales.
- Tomar conciencia de las implicaciones del trabajo de ingeniero de software, tanto individualmente como formando parte de un equipo
- Reforzar la visión inicial del campo de la Ingeniería del Software.
- Tener una perspectiva global del proceso asociado al ciclo de vida de un producto software.
- Conocer los problemas, principios, métodos y herramientas propios de la Ingeniería del Software.
- Conocer los elementos básicos de un modelo objeto.
- Conocer la importancia y la influencia de los requisitos en el éxito de un proceso software.
- Aplicar las técnicas básicas de ingeniería de requisitos.
- Conocer los principios de un método de análisis y diseño orientado a objetos.
- Conocer los principios de un método de análisis y diseño estructurado.
- Adquirir la capacidad de crear documentaciones técnicas completas, correctas y legibles.

5.- Contenidos

Unidad Didáctica I: Conceptos básicos

Tema 0: Sumario de la asignatura

- 0.1. Motivación.
- 0.2. Enfoque de ingeniería en el desarrollo del software.
- 0.3. Docencia de la Ingeniería del Software II.
- 0.4. Asignatura Ingeniería del Software II en ITIG.
 - 0.4.1. Objetivos.
 - 0.4.2. Temario.
 - 0.4.3. Evaluación.
 - 0.4.4. Bibliografía básica recomendada.
 - 0.4.5. Otras fuentes.
 - 0.4.6. Enlaces.
 - 0.4.7. Herramientas CASE.
 - 0.4.8. Tutorías.
- 0.5. Prerrequisitos: Conocimiento de los conceptos desarrollados en la asignatura Ingeniería del Software I.

Tema 1. Modelo Objeto. Una descripción de UML

- 1.1. Introducción a la orientación a objetos
- 1.2. Modelo objeto
- 1.3. ¿Qué es UML?
- 1.4. Historia de UML
- 1.5. Visión global de UML

- 1.6. Vista estática
- 1.7. Vista de interacción
- 1.8. Vista de casos de uso
- 1.9. Vista de máquina de estados
- 1.10. Vista de actividad
- 1.11. Vistas físicas

Tema 2. Introducción a la ingeniería del software asistida por computador

- 2.1. Introducción.
- 2.2. Clasificación de herramientas CASE.
- 2.3. Componentes de una herramienta.
- 2.4. Integración.

Unidad Didáctica II: Gestión de proyectos

Tema 3. Gestión de proyectos

- 3.1. Introducción.
- 3.2. Gestión de un proyecto de construcción del software.
- 3.3. Actividades para la planificación de un proyecto de construcción del software.

Unidad Didáctica III: Control de calidad del software

Tema 4. Control de calidad del software

- 4.1. Introducción
- 4.2. Conceptos de calidad.
- 4.3. Tendencias de la calidad.
- 4.4. Garantía de calidad del software.
- 4.5. Revisiones del software.
- 4.6. Revisiones técnicas formales.
- 4.7. Estándares de calidad.
- 4.8. El plan de calidad del software.

Unidad Didáctica IV: Evolución y mantenimiento del software

Tema 5. Evolución y mantenimiento del software

- 5.1. Introducción
- 5.2. Actividades de mantenimiento.
- 5.3. El proceso de mantenimiento.
- 5.4. Garantía de calidad del software.
- 5.5. Problemas del mantenimiento.
- 5.6. Costes de mantenimiento.
- 5.7. Herramientas y técnicas.
- 5.8. Ingeniería inversa y reingeniería.

6.- Competencias a adquirir*

Competencias Específicas. (En relación a los conocimientos, habilidades. y actitudes: conocimientos destrezas, actitudes...)

Competencias Específicas Unidad Didáctica I: Conceptos básicos

- Conocer el alcance de un producto software. Razonar acerca de la calidad del software.
- Entender el método de resolución de problemas que sustenta la Ingeniería del Software.
- Diferenciar las perspectivas propias del dominio de problema y del dominio de la solución.
- Entender el concepto de proceso software y conocer los principales modelos de proceso o paradigmas de ciclo de vida del software. Razonar acerca de su necesidad en la producción de software de calidad.
- Entender la noción de metodología y conocer los diferentes tipos existentes.
- Conocer la estructura de un proceso iterativo e incremental, que esté dirigido por casos de uso.
- Conocer el concepto y alcance de las herramientas CASE.
- Comprender los elementos que conforman un modelo objeto.
- Conocer y comprender el lenguaje de modelado unificado (UML), con especial atención a las vistas estática y de interacción.
- Ser capaz de aplicar las vistas estáticas y de interacción en modelos concretos y de complejidad baja y media, correspondientes a diversos artefactos software.
- Conocer y comprender la vista de casos de uso de UML.
- Ser capaz de aplicar la vista de casos de uso en proyectos software de complejidad baja y media.

Competencias Específicas Unidad Didáctica II: Gestión de proyectos

- Conocer los conceptos implicados en la gestión de un proyecto de construcción del software.
- Ser capaz de trazar las actividades para la planificación de un proyecto de construcción del software.

Competencias Específicas Unidad Didáctica III: Control de calidad del software

- Conocer los conceptos implicados en la calidad del software.
- Conocer las tendencias de los sistemas de calidad.
- Conocer los métodos para garantizar de calidad del software.
- Resaltar la importancia de las revisiones del software y de las revisiones técnicas formales.
- Conocer los estándares de calidad.
- Ser capaz de entender y aplicar planes de calidad del software.

Competencias Específicas Unidad Didáctica IV: Evolución y mantenimiento del software

- Tomar conciencia de la evolución y mantenimiento del software, y ser capaz de trazar planificaciones para tales fines.
- Ser capaz de planificar actividades de mantenimiento.
- Conocer las líneas principales actuación en el proceso de mantenimiento software, de asegurar la calidad del mismo, así como una planificar una posible acción preventiva de los problemas que pudieran surgir.
- Ser capaz de plasmar y planificar los costes de mantenimiento.
- Conocer las herramientas y técnicas.
- Introducir el desarrollo de la capacidad de para efectuar procesos de ingeniería inversa y reingeniería.

Transversales: (Competencias Instrumentales: <cognitivas, metodológicas, tecnológicas o lingüísticas>; Competencias Interpersonales <individuales y sociales>; o Competencias Sistémicas. <organización, capacidad emprendedora y liderazgo>

Competencias instrumentales

Dentro de las competencias instrumentales se distinguen entre habilidades cognitivas, capacidades metodológicas, destrezas tecnológicas y destrezas lingüísticas.

1. Habilidades cognitivas

Se distinguirán unas habilidades cognitivas generales y otras agrupadas por cada unidad didáctica.

Generales

- Conocer y entender las relaciones que mantiene la asignatura con el resto de asignaturas de la titulación a partir de la introducción explícita de ejemplos y comentarios, aprovechando puntos relevantes del temario.
- Conocer y comprender algunas de las capacidades, aptitudes y conocimientos que la asignatura aporta para el desarrollo de los diversos perfiles profesionales.
- Ser capaz de crear documentaciones técnicas completas, correctas y legibles.

2. Capacidades metodológicas

- Ser capaz de tomar decisiones de manera razonada.
- Tener capacidad de análisis y síntesis.
- Ser capaz de manejar bibliografía relacionada con la Ingeniería del Software.
- Ser capaz de documentar la solución a un problema dado en las diferentes fases del ciclo de vida con los diferentes niveles de abstracción y aproximación que esto requiere.
- Ser capaz de aplicar correctamente y manejar con comodidad los diferentes elementos de los que dispone un lenguaje de modelado concreto, ya sea orientado a objetos u orientado a procesos.
- Ser capaz de modelar un sistema de información desde las diferentes perspectivas que exige un determinado proceso.

3. Destrezas tecnológicas

- Habilidades básicas de navegación por la Web y uso del resto de servicios de red para la obtención y manejo de la información relacionada con la asignatura.
- Manejar las operaciones básicas y de administración de diferentes sistemas operativos para la instalación de las diferentes herramientas CASE a utilizar en la asignatura.
- Manejar con fluidez diferentes herramientas CASE.
- Manejo avanzado de un sistema de procesamiento de textos para la realización de los informes de las prácticas y la documentación de la práctica obligatoria.

4. Destrezas lingüísticas

- Adquirir y utilizar con fluidez un buen lenguaje científico, tanto oral como escrito, siendo riguroso en las explicaciones de cualquier proceso dentro de la asignatura.

- Conocer y utilizar la terminología usual de la Ingeniería del Software, tanto en español como en inglés.

Competencias interpersonales

Las competencias interpersonales se dividen en competencias para las tareas colaborativas y competencias relativas al compromiso con el trabajo.

1. Competencias para tareas colaborativas

- Ser capaz de realizar de trabajar en equipo para resolver los problemas de modelado enunciados en los talleres/laboratorios prácticos de la asignatura.
- Ser capaz de presentar en público una solución a un problema planteado y mantener un debate con el resto de la clase sobre la solución planteada, para así buscar colaborativamente la mejor solución al problema.
- Ser capaz de realizar y defender la práctica obligatoria de la asignatura en equipo.

2. Compromiso con el trabajo

- Se ha de definir un plan de trabajo en el que el volumen de trabajo de todos los miembros del equipo sea similar.
- Una vez finalizado el trabajo, todos los miembros del grupo deben conocer en profundidad todo el desarrollo realizado.
- Se debe cumplir el plazo de entrega de los trabajos.
- Se debe adquirir un compromiso ético entre todos los componentes del grupo.

Competencias sistémicas

Las competencias sistémicas hacen referencia a la integración de las capacidades cognitivas, destrezas prácticas y disposiciones.

- Capacidad de aplicar los conocimientos, métodos y herramientas vistos en la asignatura de Ingeniería del Software II a situaciones y problemas concretos del área de la Ingeniería Informática y de otras disciplinas relacionadas.
- Capacidad de aprender y aplicar, de forma autónoma e interdisciplinar, nuevos conceptos y métodos relacionados con cada asignatura.
- Motivación por la calidad y por la creatividad.
- Capacidad de adoptar el proceso marcado por el método científico y de ingeniería en el planteamiento y realización de trabajos diversos, tanto a nivel académico como profesional.
- Capacidad de asimilación y adaptación a la evolución del estado del arte en el ámbito de la Ingeniería del Software y de la Ingeniería Informática como profesión.

**Según la clasificación establecida por la ANECA, esta tabla puede ser más adecuada para las asignaturas que ya están adaptadas al modelo del EEES. En los documentos recogidos por la ANECA para cada titulación, se especifican las competencias tanto específicas como transversales o genéricas. Esta relación de competencias se puede consultar en: http://www.aneca.es/modal_eval/conver_docs_titulos.html*

7.- Metodologías

Los futuros planes de estudio que se definan dentro del EEES se encaminan a potenciar una mayor creatividad en las aulas, más que aprender por aprender conocimientos, cambiando así el centro de atención hacia el alumno más que el aprendizaje en sí mismo. La labor del profesor se transforma de esta manera en conseguir que el alumno aprenda a aprender en lugar de seguir siendo meros transmisores de conocimiento.

El alumno en este modelo también ve como su rol, normalmente pasivo, se ha de transformar en uno mucho más activo que lo haga partícipe de este proceso formativo, teniendo que:

- Convertirse en el responsable de su aprendizaje, debiendo gestionar y controlar la forma en que se produce.
- Desarrollar un espíritu crítico y una actitud abierta ante los cambios científico-técnicos de su especialidad, con curiosidad intelectual y rigor científico.
- Fomentar actitudes y adquirir técnicas para trabajar en equipo con eficacia.
- Basar en principios deontológicos su futura actividad profesional.
- Valorar el sentido humanístico de la ciencia y de la técnica como resultado del esfuerzo de generaciones anteriores, que debe servir para resolver los problemas del hombre actual y futuro.
- Estimular el perfeccionamiento profesional y la formación continua a lo largo de la vida.

Siendo congruentes con lo anterior, y con carácter general, el Ingeniero Técnico en Informática de Gestión debe estar capacitado para aprender a conocer, hacer, convivir y ser, en su ámbito personal, profesional y social, de acuerdo con lo recogido en el informe de la UNESCO sobre las perspectivas de la educación en el siglo XXI.

Para caminar en este sentido, el modelo educativo que se va a seguir en la asignatura de Ingeniería del Software II tiene en la clase magistral un elemento importante, pero ya no exclusivo, en la transmisión de conocimiento. Este tipo de enseñanza se va a complementar con otros procesos entre los que cabe destacar las prácticas basadas en enseñanza colaborativa y organizadas en talleres/laboratorios. Concretamente las actividades que se proponen son las siguientes:

- *Clases de teoría con apoyo de material audiovisual.* En estas clases se presentarán los contenidos básicos de un cierto tema. Las clases comenzarán con una breve introducción de los contenidos que se pretenden transmitir en la clase, así como con un breve comentario a los conceptos vistos en clases anteriores y que sirven de enlace a los que se pretenden desarrollar. El desarrollo de la clase se llevará a cabo con medios audiovisuales, textos, transparencias, etc. que permitan un adecuado nivel de motivación e interés en los alumnos. Se debe intentar motivar a los alumnos a intervenir en cualquier momento en las clases para hacer éstas más dinámicas y facilitar el aprendizaje. Es importante intentar terminar la exposición con las conclusiones más relevantes del tema tratado.
- *Talleres/laboratorios de prácticas.* Las clases prácticas estarán dedicadas a la resolución colaborativa de problemas de modelado, pudiendo dar lugar a la ejecución de algún proyecto sobre algún caso real propuesto.
- *Práctica obligatoria.* Los grupos de prácticas deberán realizar una especificación de requisitos del software (completada con algunas partes de diseño) sobre un tema libre (previamente acordado con profesor y contando con el visto bueno de éste para poder empezar) que ellos han debido de cerrar. Esa práctica tendrá que defenderse en grupo y/o de forma individual.
- *Entrega de ejercicios.* Un alumno individualmente puede entregar ejercicios resueltos por él, cuyos enunciados debe encontrar en la bibliografía (sin que estén resueltos). Éstos serán discutidos con el profesor, bien en horas de tutoría, o bien, en función del interés que considere el profesor que puede repercutir sobre el resto de alumnos, presentados

por el alumno a la clase con la correspondiente discusión posterior.

- *Tutorías.* El alumnado tiene a su disposición las tutorías en las que puede consultar cualquier duda relacionada con los contenidos, organización y planificación de la asignatura, bien de forma presencias o bien de forma virtual. Las tutorías pueden ser individualizadas, pero se admite tutorías grupales para resolver problemas relacionados con las actividades a realizar en grupo.
- *Página web.* Se convierte en el vehículo de comunicación principal y registro de información de la asignatura, sin exclusión de los medios tradicionales.

8.- Previsión de Técnicas (Estrategias) Docentes*

Opcional para asignaturas de cualquier curso			
	Horas presenciales	Horas no presenciales	Horas totales
Clases magistrales			
Clases prácticas			
Seminarios			
Exposiciones y debates			
Tutorías			
Actividades no presenciales			
Preparación de trabajos			
Otras actividades			
Exámenes			
TOTAL			

*Esta tabla está pensada para aquellas asignaturas que **no** han sido planificadas teniendo en cuenta los créditos ECTS.

8.- Previsión de Técnicas (Estrategias) Docentes*

Opcional para asignaturas de 1er curso				
	Horas presenciales.	Horas no presenciales.	Horas de trabajo autónomo del alumnos	Horas totales
Clases magistrales	22		33	55
Clases prácticas	16		24	40
Seminarios	7		7	14
Exposiciones y debates	8		8	16
Tutorías	2		2	4
Actividades no presenciales		2		2
Preparación de trabajos	2		10	12
Otras actividades				0
Exámenes	2		4	6
Revisión exámenes	1			1
				0
TOTAL	60	2	88	150

*Para las asignaturas cuya estructura y organización se haya realizado en base a los créditos ECTS.

Libros de consulta para el alumno

Amescua Seco, A.; et al. *Análisis y Diseño Estructurado y Orientado a Objetos de Sistemas de Información*. McGraw-Hill, 2003.

Dawson, C.W.; Martín, G. *El Proyecto Fin de Carrera en Ingeniería Informática*. Prentice Hall, 2002.

Booch, G., Rumbaugh, J., Jacobson, I. *El Lenguaje Unificado de Modelado*. 2ªEd. Addison Wesley, 2004.

Jacobson, I., Booch, G., Rumbaugh, J. *El Proceso Unificado de Desarrollo de Software*. Addison-Wesley, 2000.

Naveda, J. F.; Sidman, S. B. *Real-World Software Engineering Problems*. IEEE Computer Society, 2006.

Piattini, M.G.; et al. *Análisis y Diseño de Aplicaciones Informáticas de Gestión*. Ra-Ma, 2004.

Pressman, R. S. *Ingeniería del Software: Un Enfoque Práctico*. 6ª Edición. McGraw-Hill, 2006.

Rumbaugh, J.; Jacobson, I.; Booch, G. *El Lenguaje Unificado de Modelado. Manual de Referencia*. 2ªEd. Addison Wesley, 2007.

Schach, S. R. *Ingeniería del Software Clásica y Orientada a Objetos*. 6ªEd. McGraw-Hill, 2006.

Sommerville, I. *Ingeniería del Software*. 7ª Edición, Addison-Wesley, 2005.

Stevens, P.; Pooley, R. *Utilización de UML en Ingeniería del Software con Objetos y Componentes*. Addison-Wesley, 2007.

Otras referencias bibliográficas, electrónicas o cualquier otro tipo de recurso.

Dargan, P. A. *Open Systems and Standards for Software Products Development*. Artech House, 2005.

ESA Board for Software Standardisation and Control (BSSC), ESA Software Engineering Standards. European Space Agency. 1991.

Eric Braude, *Software Engineering. An Object-Oriented Perspective*. John Wiley & Sons. 2001.

Eckel, B. *Thinking in Patterns with Java*. MindView Inc. <http://www.mindview.net/>

IEEE Std. 1058-1987, *IEEE Standard for Software Project Management Plan*.

IEEE Std. 729-1983, *IEEE Standard Glossary of Software Engineering Terminology*.

IEEE Std. 1175.3-2004, *IEEE Standard for CASE Tool Interconnections-Reference Model for Specifying Software Behavior*.

IEEE Std.1063-2001, *IEEE standard for software user documentation*.

International Journals on Software Engineering:

- IEEE Transactions on Software Engineering.
- ACM Transactions on Software Engineering and Methodology.
- Empirical Software Engineering.

- International Journal of Software Engineering and Knowledge Engineering.

Gamma, E., *Design Patterns, Elements of Reusable Object-Oriented Software*. Addison-Wesley. 1995.

Object Management Group *Unified Modeling Language Specification*. 2003. <http://www.omg.org/>

10.- Evaluación

Consideraciones Generales

Además de los exámenes ordinarios, se podrá tener en cuenta para la evaluación todo tipo de cuestiones, problemas, ejercicios, talleres, prácticas, y cualquier otro tipo de aportación que considere oportuno y plantee el profesor, lo cual queda a criterio del profesor.

Criterios de evaluación

Examen escrito de conocimientos teóricos y ejercicios prácticos.

Entrega obligatoria y exposición de prácticas, trabajos y/o proyectos de prácticas.

Será imprescindible aprobar la teoría y las prácticas independientemente para poder aprobar la asignatura, esto implica obtener una calificación igual o superior a cinco puntos (5) sobre diez (10) en cada una de las partes.

Instrumentos de evaluación

Examen escrito de conocimientos teóricos y ejercicios prácticos.

Entrega obligatoria y exposición de prácticas, trabajos y/o proyectos de prácticas.

Será imprescindible aprobar la teoría y las prácticas independientemente para poder aprobar la asignatura, esto implica obtener una calificación igual o superior a cinco puntos (5) sobre diez (10) en cada una de las partes.

Además de los exámenes ordinarios, se podrá tener en cuenta para la evaluación todo tipo de cuestiones, problemas, ejercicios, talleres, prácticas, y cualquier otro tipo de aportación que considere oportuno y plantee el profesor, lo cual queda a criterio del profesor.

Recomendaciones para la evaluación.

--

Recomendaciones para la recuperación.

--

