

Apéndice A:

Plan de Calidad de la Unidad Docente de IS y OO

Se incluye íntegramente la versión 1.1 del Plan de Calidad de la Unidad Docente de Ingeniería del Software y Orientación a Objetos del Departamento de Informática y Automática de la Universidad de Salamanca para el bienio 1999-2000, cuya última modificación data del 20 de marzo de 2000 (por lo tanto existen algunas diferencias con respecto a las asignaturas presentadas en el presente Proyecto Docente e Investigador). Este plan de calidad ha sido elaborado por el Dr. Francisco José García Peñalvo, la Dra. María N. Moreno García, el Dr. José Rafael García-Bermejo Giner y Ana de Luis Reboredo.

A.1 Introducción

Hay muchas definiciones de calidad en la bibliografía, según la norma ISO 8402 (admitida como norma española UNE 66-001-92), la calidad se define como: “*Totalidad de características de un producto o servicio que le confieren su aptitud para satisfacer unas necesidades expresadas o implícitas*”.

El concepto de calidad es un objetivo fundamental para los directivos de una empresa junto a los dos parámetros clásicos de su gestión: el dinero y el tiempo. El mercado actual es

sumamente competitivo, donde no basta con producir masivamente los productos o servicios, vender es lo importante y sólo se produce un producto cuando se tiene la seguridad de aceptación por parte del cliente. Así, conseguir la satisfacción del cliente y conocer sus necesidades, para luego definir las en forma de requisitos a cumplir, se convierte en una necesidad imperiosa para las empresas que pretendan hacerse con un hueco en un determinado mercado.

En la vida cotidiana, la calidad representa las propiedades inherentes a un objeto, de forma que pueda ser comparado con otros objetos de su especie para determinar si es mejor, igual o peor. Calidad es sinónimo de bondad, excelencia o superioridad.

Para la comunidad universitaria, la calidad debe ser un objetivo tan importante como para la empresa. El profesor de Universidad debe ver en la calidad un camino hacia la excelencia en todas sus actividades: docencia, investigación y gestión.

En lo referente a la actividad investigadora, la calidad debe ser la guía para aumentar el conocimiento del investigador, y debe estar presente en todos los productos resultantes de la investigación.

La actividad docente por su parte puede asociarse más a una actividad contractual donde la Universidad es la empresa y los clientes son varios. El cliente más directo es el alumno, el producto que se le ofrece debe ser un currículo que cumpla unos requisitos acordes a los objetivos marcados por la titulación elegida. La sociedad es el cliente indirecto de la Universidad, donde el producto que se le ofrece son los alumnos formados y preparados para introducirse en el mundo real y rendir acorde a las necesidades de esa sociedad. Por último, el propio docente será a la vez mecanismo y cliente de la Universidad, al buscar por un lado la excelencia de conocimiento en una determinada materia y por otro lado la satisfacción personal conseguida cuando los alumnos salen cumpliendo los requisitos de conocimientos que la Universidad marca y la adecuación que la sociedad demanda.

Entre los cursos académicos 96-97 y 98-99 se han desarrollado una serie de actividades destinadas a la mejora continua en el ámbito de la docencia en los campos de la Ingeniería del Software y de la Orientación a Objetos, las cuales se han visto plasmadas en un conjunto de planes de calidad que se centraban en recoger los objetivos y los criterios de evaluación de los mismos para asignaturas concretas, cayendo la responsabilidad de poner en marcha estos planes en la persona que acometía esta tarea [García et al., 1999a].

Como complemento a esta actividad de carácter meramente individual, se estableció un mecanismo de evaluación externa, que permitiese al responsable de la misma obtener una mayor

información contrastada con la que poder realimentar el proceso de mejora continua [García et al., 1999b].

Sin embargo, estas iniciativas individuales no han influido de una forma clara en la obtención de una mejora global dentro de un colectivo (área, departamento, titulación...) al estar situadas en los límites de una sola asignatura.

Por este motivo, se va acometer la tarea de la realización de un plan de calidad para el bienio 1999-2001, pero no centrado en una única asignatura, sino que abarque una unidad docente completa, la unidad docente de Ingeniería del Software y Orientación a Objetos, que tiene competencias en dos titulaciones universitarias diferentes, aunque relacionadas: la *Ingeniería Técnica en Informática de Sistemas (Plan de 1997)* y la *Ingeniería Informática*, ambas impartidas en la Universidad de Salamanca.

El plan de calidad se estructura como sigue: en la sección dos se presenta y se justifica la unidad docente de Ingeniería del Software y Orientación a Objetos, especificando los objetivos perseguidos por esta unidad docente, las asignaturas que la forman, indicando sus interrelaciones, y estableciendo a groso modo como se reparten los objetivos de la unidad docente en las asignaturas. La tercera sección se centra en detallar de forma concreta los objetivos y las aportaciones de cada una de las asignaturas, con la misión específica, en esta primera aproximación para el bienio 1999-2001, de minimizar solapamientos de contenidos y de conocer explícitamente los contenidos y límites del resto de las asignaturas que conforman la unidad docente.

A.2 Unidad docente de Ingeniería del Software y Orientación a Objetos

La Ingeniería del Software es la parte de la disciplina informática que se ocupa de ofrecer una aproximación sistemática a la creación y mantenimiento de los sistemas software, ofreciendo un enfoque que rehuye las concepciones tradicionales, por las que el software se concibe como un producto artesano inmerso en un ambiente *mitológico*, casi *místico*, donde los sistemas software distan mucho de las necesidades reales del tejido social y empresarial actual.

En la aproximación buscada por la Ingeniería del Software se concibe al *ingeniero informático* como aquella persona que es capaz de aplicar sus conocimientos para la resolución de los problemas reales derivados de la concepción, explotación y mantenimiento de sistemas informáticos, donde la perspectiva no es la de pequeños sistemas, que pueden ser

responsabilidad de un único individuo, sino de los sistemas que requieren para su realización el trabajo conjunto y coordinado de diferentes personas.

En España, actualmente, los estudios universitarios de informática se organizan en ingenierías técnicas e ingenierías superiores, siendo una política acorde con los que defienden que la informática es una ingeniería [Buxton et al., 1976; Shaw y Tomayko, 1991; Leveson, 1997; Parnas, 1997]. Así, el cuerpo de conocimientos de un ingeniero informático debe estar formado por una sólida base en Lógica, Matemáticas y Ciencia de la Computación, aspectos propios de Ingeniería del Software y un conjunto de temas de carácter universal que completarán su currículo (economía, idiomas...).

La unidad docente de Ingeniería del Software y Orientación a Objetos del Departamento de Informática y Automática de la Universidad de Salamanca, se encarga de impartir la docencia relacionada con la Ingeniería del Software en las titulaciones que se discuten.

La inclusión de la orientación a objetos como parte de la unidad docente se debe a la gran repercusión de los métodos orientados a objetos en el desarrollo de sistemas software, lo cual lleva a hablar de una Ingeniería del Software Orientada a Objetos que, por las propias características intrínsecas de la tecnología de objetos, abarca con un mayor énfasis todas las fases del ciclo de vida del software.

A.2.1 Objetivos de la unidad docente

Los objetivos de la unidad docente se pueden clasificar en tres grandes apartados:

- Conceptos teóricos
- Aspectos prácticos
- Habilidades personales

A.2.1.1 Conceptos teóricos

- T1** Descripción de las actividades técnicas e ingenieriles que se llevan a cabo en el ciclo de vida de un producto software.
- T2** Descripción de los problemas, principios, métodos y tecnologías asociadas con la Ingeniería del Software.
- T3** Importancia de los requisitos en el ciclo de vida del software.
- T4** Elicitación, documentación, especificación y prototipado de los requisitos de un sistema software.
- T5** Especificaciones formales de requisitos.
- T6** Método de análisis/diseño estructurado.

- T7** Método de análisis/diseño orientado a objetos.
- T8** Diseño de la interfaz gráfica de usuario.
- T9** Estudio y comprensión de los fundamentos del diseño de sistemas software.
- T10** Gestión de proyectos software: definición de objetivos, gestión de recursos, estimación de esfuerzo y coste, planificación y gestión de riesgos.
- T11** Uso de métricas software para el apoyo a la gestión de proyectos software y aseguramiento de la calidad del software.
- T12** Conceptos, métodos, procesos y técnicas destinadas al mantenimiento y evolución de los sistemas software.
- T13** Conocimiento sobre el uso de la Ingeniería del Software en dominios de aplicación específicos.

A.2.1.2 Aspectos prácticos

- P1** Aplicación de forma práctica los conceptos teóricos sobre el desarrollo estructurado.
- P2** Aplicación de forma práctica los conceptos teóricos de Orientación a Objetos.
- P3** Aplicación de forma práctica los conceptos teóricos sobre gestión de proyectos.
- P4** Utilización de herramientas CASE para la gestión y desarrollo de sistemas software.
- P5** Programación orientada a objetos.
- P6** Realización de interfaces gráficas de usuario en diferentes plataformas.
- P7** Aprendizaje y manejo de forma práctica de plataformas, entornos de desarrollo, lenguajes de programación... de alta repercusión en el desarrollo de sistemas software en la actualidad.
- P8** Recolección de diferentes métricas en el desarrollo de sistemas software reales.
- P9** Construcción de sistemas software de entidad superior a una práctica de laboratorio, a ser posible partiendo de unas especificaciones reales elicítadas a *clientes y/o usuarios* reales.

A.2.1.3 Habilidades personales

- H1** Mejora de la expresión oral.
- H2** Mejora en la redacción de documentos técnicos.
- H3** Potenciación de la capacidad del alumno para la búsqueda de información (manejo de fuentes bibliográficas, Internet, foros de discusión...).
- H4** Capacitación de los alumnos para el trabajo en grupo.

A.2.2 Asignaturas de la unidad docente

La unidad docente de Ingeniería del Software y Orientación a Objetos se ha creado para dar cobertura a las titulaciones de Ingeniería Técnica en Informática de Sistemas (I.T.I.S) (Plan de 1997) e Ingeniero en Informática (I.I) (Plan de 1998) impartidas en la Universidad de Salamanca.

La primera de ellas es una titulación de primer ciclo (de tres cursos de duración), mientras que la segunda es una titulación de segundo ciclo (de dos cursos de duración), que tiene como condición de acceso la posesión del título de Ingeniero Técnico en Informática de Gestión o Ingeniero Técnico en Informática de Sistemas.

En estas titulaciones las asignaturas que mejor se ajustan a los objetivos marcados en la unidad docente se recogen en la Tabla A.1.

Asignatura	Titulación	Curso	Carácter	Créditos
Interfaces Gráficas	I.T.I.S	2º	Optativa	6 (3T + 3P)
Ingeniería del Software	I.T.I.S	3º	Obligatoria	6 (4,5T + 1,5P)
Programación Orientada a Objetos	I.T.I.S	3º	Optativa	6 (3T + 3P)
Proyecto	I.T.I.S	3º	Obligatoria	9 (9P)
Análisis de Sistemas	I.I	1º	Troncal	9 (6T + 3P)
Administración de Proyectos Informáticos	I.I	2º	Troncal	9 (6T + 3P)
Sistemas de Información	I.I	2º	Troncal	9 (9P)
Proyecto	I.I	2º	Troncal	6 (6P)

Tabla A.1. Asignaturas que componen la unidad docente

Las dependencias e interrelaciones entre estas asignaturas se muestran en la Figura A.1. En el establecimiento de estas dependencias se ha tenido en cuenta tanto el factor tiempo, que claramente establece el orden lógico en el que se van a cursar las asignaturas, como las aportaciones a la unidad docente de las mismas en forma de contenidos y objetivos a conseguir.

Además de las asignaturas propias de la unidad docente, se incluye la asignatura de **Diseño de Bases de Datos**. Esto se debe a que es la asignatura en la que se introducen los modelos de datos conceptuales y lógicos (diagramas entidad-interrelación y modelos relacionales típicamente), lo que supone una importante base, a la vez que una descarga, para la asignatura de Ingeniería del Software donde estos modelos serán utilizados de forma práctica sin necesidad de tener que incluirlos en la parte teórica de la asignatura.

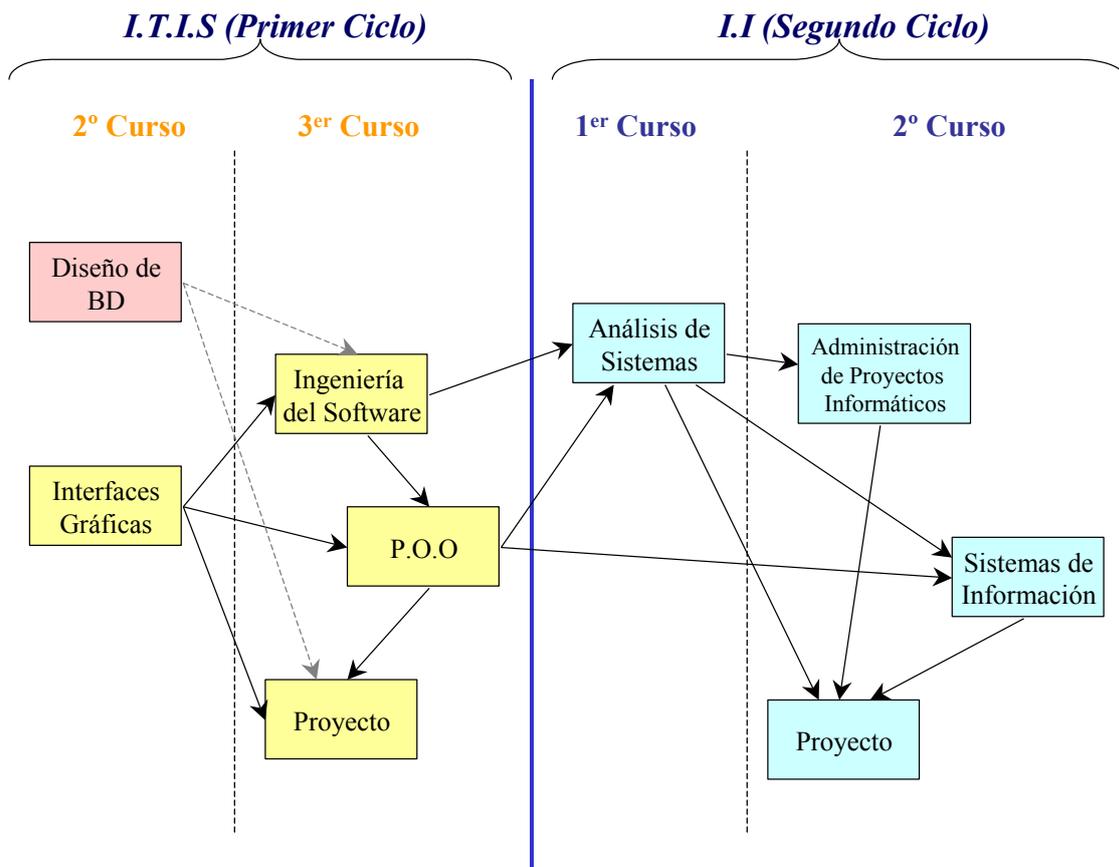


Figura A.1. Dependencias entre las asignaturas que forman la unidad docente

A.2.3 Reparto de los objetivos en las asignaturas de la unidad docente

	Obj. Teóricos	Obj. Prácticos	Hab. Personales
Interfaz Gráficas	T8	P6	H1, H2, H3
Ingeniería del Software	T1, T2, T3, T4, T6, T7, T9	P1, P2, P4	H1, H2, H3, H4
Programación Orientada a Objetos	T7, T9	P2, P5	H1, H2, H3, H4
Proyecto I.T.I.S		P9	H1, H2, H3
Análisis de Sistemas	T3, T4, T5, T7, T12, T13	P1, P2, P4	H1, H2, H3, H4
Administración de Proyectos Informáticos	T10, T11, T12	P4, P8	H1, H2, H3, H4
Sistemas de Información		P7	H1, H2, H3
Proyecto I.I		P8, P9	H1, H2, H3

Tabla A. 2. Reparto de los requisitos en las asignaturas de la unidad docente

A.3 Estudio de los objetivos de cada una de las asignaturas

A.3.1 Interfaces gráficas

El objetivo de esta asignatura es introducir a los alumnos de I.T.I.S en algunos sistemas de desarrollo de interfaces gráficas de frecuente uso en la actualidad. Se trata pues de un objetivo fundamentalmente práctico, ya que se busca la capacitación del alumno para implementar aplicaciones que incorporen los elementos habituales en las interfaces gráficas de usuario actuales.

Este objetivo se corresponde con los objetivos **T8, P6, H1, H2** y **H3** de la unidad docente.

A continuación, se detallan los prerequisites de esta asignatura, las líneas de acción a seguir en cada uno de ellos, el temario teórico/práctico de la asignatura, los criterios de evaluación y la bibliografía básica de consulta recomendada.

A.3.1.1 Ficha de la asignatura

Asignatura	<i>Interfaces gráficas (Optativa)</i>
Créditos	3T + 3P
Estudios	<i>I.T.I.S</i>
Plan	<i>B.O.E de 4-11-1997</i>
Curso	2º
Cuatrimestre	2º
Responsable	<i>Ana de Luis Reboredo (adeluis@usal.es)</i>
Página web de la asignatura	

Tabla A.3. Interfaces gráficas

A.3.1.2 Prerrequisitos

Es imprescindible que el alumno esté capacitado para el desarrollo de programas en lenguaje C. Estos conocimientos deben adquirirse en las asignaturas relacionadas con la programación en el primer y segundo curso **Algoritmia, Programación, Laboratorio de Programación y Estructuras de Datos.**

Sería muy conveniente que el alumno ya dispusiese de ciertos conocimientos de programación orientada a objetos, que le facilitaran el acceso a ciertas herramientas de creación de interfaces gráficas actuales. No obstante, con la actual distribución temporal de asignaturas,

esto no es muy frecuente. Por ello, en la asignatura se proporcionarán unos conocimientos básicos de programación orientada a objetos y, en concreto, del lenguaje C++ que permitan a los alumnos utilizar dichas herramientas.

A.3.1.3 Temario teórico/práctico

Presentación de la asignatura (1 Hora)

Unidad Docente I: La GUI de Windows. Programación SDK (11 Horas)

Tema 1. Introducción a la programación en Windows (1 Hora)

Tema 2. GDI. Fundamentos gráficos (2 Horas)

Tema 3. Eventos de entrada. Ratón y teclado (1 Hora)

Tema 4. Controles. Controles predefinidos (2 Horas)

Tema 5. Recursos (1 Hora)

Tema 6. Menús (1 Hora)

Tema 7. Cuadros de diálogo (2 Horas)

Tema 8. Intercambio de datos (1 Hora)

Unidad Docente II: Visual C++ (8 Horas)

Tema 8. Introducción a la programación orientada a objetos (2 Horas)

Tema 9. La biblioteca MFC (6 Horas)

Unidad Docente III: El entorno X Window (10 Horas)

Tema 10. Introducción. Arquitectura del sistema (2 Horas)

Tema 11. Conceptos básicos. Terminología (2 Horas)

Tema 12. Ventanas (2 Horas)

Tema 13. Entradas. El ratón y el teclado (2 Horas)

Tema 14. Gráficos (2 Horas)

Tabla A.4. Programa de teoría de Interfaces Gráficas (3 créditos)

Prácticas (30 Horas)

Realización de aplicaciones mediante programación SDK (12 Horas)

Realización de aplicaciones con Visual C++ (8 Horas)

Realización de aplicaciones con X Window (10 Horas)

Práctica obligatoria

Realización de una aplicación completa que incorpore la mayoría de los elementos estudiados.

Tabla A.5. Programa de prácticas de Interfaces Gráficas (3 créditos)

A.3.1.4 Líneas de acción

Para cada uno de los objetivos de la unidad docente que debe satisfacerse en esta asignatura se van a identificar las líneas de acción a seguir.

T8 Diseño de la interfaz gráfica de usuario.

Línea de acción	Cuándo	Quién	Medios	Evaluación
Presentación de los diferentes elementos que componen una GUI y sus distintos modos de implementación	Clases Teóricas	Responsable de la asignatura	Transparencias Bibliografía básica	Examen

P6 Realización de interfaces gráficas de usuario en diferentes plataformas.

Línea de acción	Cuándo	Quién	Medios	Evaluación
Realización de sucesivas aplicaciones que incorporen elementos de interfaces gráficas para Windows y X-Window	Clases prácticas	Responsable de la asignatura Alumnos	Aula de Informática Transparencias Bibliografía básica	Defensa de la práctica obligatoria Examen

H1 Mejora de la expresión oral.

Línea de acción	Cuándo	Quién	Medios	Evaluación
Defensa oral de la práctica obligatoria	Defensa de la práctica	Grupos de alumnos Responsable de la asignatura	Ordenador	Defensa de la práctica

H2 Mejora en la redacción de documentos técnicos.

Línea de acción	Cuándo	Quién	Medios	Evaluación
Documentación de la práctica obligatoria	Defensa de la práctica	Grupo de alumnos Responsable de la asignatura	Bibliografía	Defensa de la práctica

H3 Potenciación de la capacidad del alumnos para la búsqueda de información (manejo de fuentes bibliográficas, Internet, foros de discusión...).

Línea de acción	Cuándo	Quién	Medios	Evaluación
Completar las transparencias utilizadas en clase con la bibliografía recomendada	A lo largo de la asignatura	Alumnos	Biblioteca Recursos de Internet	Examen

A.3.1.5 Criterios de evaluación de la asignatura

- a) Parte de Teoría (50% de la nota final).
 - Un examen final en el mes de Junio.
 - Un examen final en el mes de Septiembre.
- b) Parte Práctica (50% de la nota final)
 - Desarrollo de una aplicación con una interfaz gráfica de usuario que incluya la mayoría de los elementos estudiados. Se realizará en grupos de un máximo de tres miembros.
 - Se realizará una defensa por grupos de dicho trabajo.

Tras la defensa, los diferentes miembros del grupo deberán responder a las preguntas que se les planteen de forma individualizada sobre el desarrollo de la aplicación y las posibles modificaciones de la misma.

La parte práctica se guardará hasta la convocatoria de septiembre, pero nunca para futuros cursos.

- c) Cada una de las partes se deberá aprobar por separado exigiéndose un mínimo de 5 puntos en cada una.

A.3.1.6 Bibliografía básica de referencia

 **Petzold, C.** “*Programación en Windows 95*”. McGraw-Hill, 1996.

Es la traducción del libro “*Programming Windows 95*” que a su vez es la continuación del clásico “*Programming Windows 3.0*” del mismo autor. Es el libro básico para la introducción en la programación SDK, imprescindible para comprender la programación en Windows. Explica detalladamente el funcionamiento de la API de Windows desarrollando de modo progresivo los distintos aspectos relacionados con la misma. Cuenta con multitud de ejemplos en C que están disponibles en CD. Se utilizará para todos los temas de la Unidad Docente I de la asignatura.

 **Reiss, L., Radin, J.** “*Aplique X Window*”. McGraw-Hill, 1993.

Se trata de la traducción del libro de los mismos autores “*X Window Inside & Out*”. Parte de una completa introducción al sistema y sus componentes para pasar a describir las funciones X Window ilustrándolas con programas completamente desarrollados. Será el libro que se utilizará en la Unidad Docente III de la asignatura.

 **Yao, P., Leinecker, R. C.** “*Todo Visual C++ 5*”. INFORBOOK'S, 1998.

Es la traducción del libro “*Visual C++ 5 Bible*” de los mismos autores. El contenido está dividido en 4 partes. La primera parte contiene una introducción a la programación en Windows en la que hace algunas consideraciones sobre la programación con la API de Microsoft Windows. A continuación, incluye una pequeña guía de uso de la herramienta de desarrollo Visual C++. La segunda parte es una introducción a la programación orientada a objetos y a las ampliaciones del lenguaje C++ sobre el lenguaje C. La tercera parte trata todos los aspectos relacionados con la librería MFC. La última parte incluye algunos temas de interés como, por ejemplo algunos formatos de archivos de imágenes o la gestión de memoria de Windows. Todo el libro está ilustrado con abundantes ejemplos que están disponibles en un CD adjunto.

A.3.2 Ingeniería del Software

El objetivo principal de esta asignatura es introducir a los alumnos de I.T.I.S en los principios fundamentales, métodos y herramientas para el desarrollo sistemático de proyectos informáticos. Adicionalmente, los alumnos deberán aplicar éstas trabajando en equipo.

Este objetivo se corresponde con los objetivos **T1, T2, T3, T4, T6, T7, T9, P1, P2, P4, H1, H2, H3** y **H4** de la unidad docente.

Para ver satisfacer estos objetivos, se va a detallar los prerequisites de esta asignatura, las líneas de acción a seguir en cada uno de ellos, el temario teórico/práctico de la asignatura, los criterios de evaluación y la bibliografía básica de consulta recomendada.

A.3.2.1 Ficha de la asignatura

Asignatura	<i>Ingeniería del software (obligatoria)</i>
Créditos	4,5T + 1,5P
Estudios	<i>I.T.I.S</i>
Plan	<i>B.O.E de 4-11-1997</i>
Curso	3º
Cuatrimestre	1º
Responsable	<i>Francisco José García Peñalvo (fgarcia@usal.es)</i>
Página web de la asignatura	http://tejo.usal.es/~fgarcia/docencia.html

Tabla A.6. Ingeniería del Software

A.3.2.2 Prerrequisitos

El alumno debe estar familiarizado con la teoría así como con la práctica del diseño y codificación en lenguajes procedurales (por ejemplo C). Estos conocimientos deben adquirirse en las asignaturas relacionadas con la programación en el primer y segundo curso **Algoritmia, Programación, Laboratorio de Programación y Estructuras de Datos**.

El alumno debe tener los conocimientos y la práctica necesaria en la creación de modelos de información conceptuales y lógicos, en concreto, dominio del modelo entidad-relación, paso al modelo relacional y normalización. Estos conceptos se adquieren en la asignatura de segundo curso **Diseño de Bases de Datos** (asignatura troncal en el Plan de Estudios vigente).

Es deseable que el alumno conozca la problemática de las interfaces de usuario y esté familiarizado con la problemática de construir interfaces gráficas de usuario. Estos aspectos deben tratarse en la asignatura **Interfaces Gráficas** del segundo curso.

A.3.2.3 Temario teórico/práctico

Presentación de la asignatura (1 Hora)

Unidad Docente I: Conceptos básicos (6 Horas)

Tema 1. Introducción a la Ingeniería del Software (6 Horas)

Unidad Docente II: Paradigma estructurado de desarrollo (23 Horas)

Tema 2. Análisis y especificación de requisitos (11 Horas)

Tema 3. Análisis estructurado (2 Horas)

Tema 4. Diseño del software (6 Horas)

Tema 5. Diseño estructurado (4 Horas)

Unidad Docente III: Introducción al paradigma objetual (13 Horas)

Tema 6. Introducción a la Orientación a Objetos (6 Horas)

Tema 7. UML (6 Horas)

Tema 8. Visión general de la metodología OMT (1 Hora)

Unidad Docente IV: Miscelánea (2 Horas)

Tema 9. Herramientas CASE (2 Horas)

Tabla A.7. Programa teórico de Ingeniería del Software (4,5 créditos)

Talleres (10 Horas)

Práctica 1. Taller de modelado de datos. Repaso al modelo entidad-interrelación (2 Horas)

Práctica 2. Taller de modelado funcional (Enfoque Clásico)(2 Horas)

Práctica 3. Taller de modelado funcional (Yourdon) (4 Horas)

Práctica 4. Taller de orientación al objeto (2 Horas)

Laboratorio (2 Horas)

Práctica 5. Manejo de una herramienta CASE (2 Horas)

Práctica obligatoria

Práctica 6. Realización de una ERS y un prototipo de una aplicación software

Tabla A.8. Programa práctico de Ingeniería del Software (1,5 créditos)

Actividades Docentes Complementarias

- Seminarios impartidos sobre temas específicos
- Trabajos voluntarios realizados por los alumnos
- Conferencias invitadas
- *Workshop* de trabajos realizados por los alumnos sobre temas de Ingeniería del Software y objetos

En la Tabla A.9 se presenta la correspondencia existente entre el temario y los objetivos perseguidos en esta asignatura.

Elemento Docente	Objetivos
Tema 1	T1, T2, H3
Tema 2	T3, T4, H3
Tema 3	T6, H3
Tema 4	T9, H3
Tema 5	T6, H3
Tema 6	T3, T7, H3
Tema 7	T7, H3
Tema 8	T7, H3
Tema 9	P4, H3
Práctica 1	P1, H1, H2, H4
Práctica 2	P1, H1, H2, H4
Práctica 3	P1, H1, H2, H4
Práctica 4	P2, H1, H2, H4
Práctica 5	P4
Práctica 6	P1, P2, P4, H1, H2, H4

Tabla A.9. Correspondencia entre el temario teórico/práctico y los objetivos de la asignatura

La Unidad Docente IV: Miscelánea, es difícil de impartir en la realidad por limitaciones de tiempo. El Tema 9, dedicado a las herramientas CASE, puede introducirse a la vez que se realiza la Práctica 4.

A.3.2.4 Líneas de acción

Para cada uno de los objetivos de la unidad docente que debe satisfacerse en esta asignatura se van a identificar las líneas de acción a seguir. Cada línea de acción va a definirse mediante los cuatro apartados indicados en [García et al., 1999a]: *cuándo se lleva a cabo, quién es el responsable de realizarla, qué medios son necesarios y cómo se evaluarán los resultados de dicha línea de acción.*

T1 Descripción de las actividades técnicas e ingenieriles que se llevan a cabo en el ciclo de vida de un producto software.

T2 Descripción de los problemas, principios, métodos y tecnologías asociadas con la Ingeniería del Software.

Línea de acción	Cuándo	Quién	Medios	Evaluación
Presentación de la problemática del desarrollo de programas con calidad industrial, poniendo de manifiesto la necesidad de emplear técnicas de ingeniería	Unidad docente I	Responsable de la asignatura	Transparencias Bibliografía básica	Reuniones con los alumnos Examen
Presentación de ejemplos de empresas que hagan uso de métodos de Ingeniería del Software bien implantados	A lo largo de la asignatura Conferencias impartidas por parte de representantes del mundo empresarial	Ponentes invitados	Medios audiovisuales Contactos en la empresa	Opinión de los alumnos Opinión de otros profesores del Departamento
Estudio y comprensión de los diferentes paradigmas de ciclo de vida de un desarrollo software	Unidad docente I	Responsable de la asignatura	Transparencias Bibliografía básica	Examen

T3 Importancia de los requisitos en el ciclo de vida del software.

T4 Elicitación, documentación, especificación y prototipado de los requisitos de un sistema software.

Línea de acción	Cuándo	Quién	Medios	Evaluación
Presentación del papel protagonista de los requisitos y de las especificaciones dentro de un desarrollo software	Durante toda la asignatura, especialmente en la Unidad II, Unidad III y en la práctica de la asignatura	Responsable de la asignatura	Transparencias Bibliografía básica	Reuniones con los alumnos Examen
Distinción entre elicitación, documentación y especificación de requisitos	Unidad docente II. Se habla de requisitos de cliente y de desarrollador	Responsable de la asignatura	Transparencias Bibliografía básica	Defensa práctica Examen
Creación de un documento de ERS de un caso real, partiendo de unas entrevistas a los interesados	Práctica obligatoria de la asignatura	Clientes Alumnos Responsable	Clientes que se presten a ser entrevistados por los alumnos	Defensa de la práctica

T6 Método de análisis/diseño estructurado.

Línea de acción	Cuándo	Quién	Medios	Evaluación
Estudio de las principales técnicas y modelos para la especificación de requisitos y diseño en el paradigma estructurado	Unidad docente II	Responsable de la asignatura	Transparencias Bibliografía básica	Talleres prácticos Defensa de la práctica obligatoria Examen
Introducción de las extensiones de notación funcional para tiempo real	Unidad docente I	Responsable de la asignatura	Transparencias Bibliografía básica	Talleres prácticos Defensa de la práctica obligatoria Examen
Creación de un documento de ERS de un caso real, partiendo de unas entrevistas a los interesados (los alumnos pueden elegir el método estructurado para especificar los requisitos del desarrollador)	Práctica obligatoria de la asignatura	Cientes Alumnos Responsable de la asignatura	Cientes que se presten a ser entrevistados por los alumnos	Defensa de la práctica

T7 Método de análisis/diseño orientado a objetos.

Línea de acción	Cuándo	Quién	Medios	Evaluación
Estudio de las principales técnicas y modelos para la especificación de requisitos y diseño en el paradigma objetual	Unidad docente III	Responsable de la asignatura	Transparencias Bibliografía básica	Talleres prácticos Defensa de la práctica obligatoria Examen
Creación de un documento de ERS de un caso real, partiendo de unas entrevistas a los interesados (los alumnos pueden elegir el método objetual para especificar los requisitos del desarrollador)	Práctica obligatoria de la asignatura	Cientes Alumnos Responsable de la asignatura	Cientes que se presten a ser entrevistados por los alumnos	Defensa de la práctica

T9 Estudio y comprensión de los fundamentos del diseño de sistemas software.

Línea de acción	Cuándo	Quién	Medios	Evaluación
Ver los fundamentos de diseño como la base necesaria para obtener sistemas software de calidad, con independencia del paradigma de desarrollo empleado	Unidad docente II	Responsable de la asignatura	Transparencias Bibliografía básica	Talleres prácticos Defensa de la práctica obligatoria Examen

P1 Aplicar de forma práctica los conceptos teóricos sobre el desarrollo estructurado.

P2 Aplicar de forma práctica los conceptos teóricos de Orientación a Objetos.

Línea de acción	Cuándo	Quién	Medios	Evaluación
Realización de talleres en parte de las horas prácticas de la asignatura donde los alumnos se organizan en grupos para resolver un problema utilizando técnicas estructuradas y orientadas a objeto. Posteriormente uno de los grupos expondrá su solución y se debatirá sobre ella	Parte práctica de la asignatura	Grupos de alumnos Responsable de la asignatura	Enunciados con casos prácticos Transparencias	Informes realizados por los alumnos Comentarios de los alumnos
Creación de un documento de ERS de un caso real, partiendo de unas entrevistas a los interesados (los alumnos pueden elegir el método objetual para especificar los requisitos del desarrollador)	Práctica obligatoria de la asignatura	Cientes Alumnos Responsable de la asignatura	Cientes que se presten a ser entrevistados por los alumnos	Defensa de la práctica

P4 Utilización de herramientas CASE para la gestión y desarrollo de sistemas software.

Línea de acción	Cuándo	Quién	Medios	Evaluación
Introducción a las herramientas CASE	Unidad docente IV Clase práctica	Responsable de la asignatura Alumnos	Transparencias Bibliografía Herramientas CASE	Defensa de la práctica
Creación de un documento de ERS de un caso real, partiendo de unas entrevistas a los interesados (los alumnos pueden elegir el método objetual para especificar los requisitos del desarrollador)	Práctica obligatoria de la asignatura	Cientes Alumnos Responsable de la asignatura	Cientes que se presten a ser entrevistados por los alumnos	Defensa de la práctica

H1 Mejora de la expresión oral.

Línea de acción	Cuándo	Quién	Medios	Evaluación
Promover los debates sobre las soluciones presentadas en los talleres prácticos	Clases prácticas	Responsable de la asignatura Grupos de alumnos	Pizarra Medios audiovisuales	Estudio del seguimiento de los talleres Opinión alumnos
Defensa oral de la práctica obligatoria	Defensa de la práctica	Grupos de alumnos Responsable de la asignatura	Medios audiovisuales	Defensa de la práctica
Promover seminarios realizados por los alumnos como resultado de trabajos voluntarios	A lo largo de la asignatura	Grupos de alumnos	Bibliografía Recursos de Internet	Nota <i>extra</i> para esos trabajos voluntarios

H2 Mejora en la redacción de documentos técnicos.

Línea de acción	Cuándo	Quién	Medios	Evaluación
Realización de un informe técnico por cada uno de los talleres prácticos realizados. Estos informes quedarán publicados en la página web de la asignatura	En las 3 semanas siguientes a una clase práctica	Responsable de la asignatura Grupos de alumnos	Página web de la asignatura	Nota <i>extra</i> para estos trabajos voluntarios
Documentación de la práctica obligatoria	Defensa de la práctica	Grupos de alumnos Responsable de la asignatura	Medios audiovisuales	Defensa de la práctica
Memoria de los trabajos voluntarios	A lo largo de la asignatura	Grupos de alumnos	Bibliografía Recursos de Internet	Nota <i>extra</i> para estos trabajos voluntarios

H3 Potenciación de la capacidad del alumnos para la búsqueda de información (manejo de fuentes bibliográficas, Internet, foros de discusión...).

Línea de acción	Cuándo	Quién	Medios	Evaluación
Completar las transparencias utilizadas en clase con la bibliografía recomendada	A lo largo de la asignatura	Alumnos	Biblioteca	Examen
Promover seminarios realizados por los alumnos como resultado de trabajos voluntarios	A lo largo de la asignatura	Grupos de alumnos	Bibliografía Recursos de Internet	Nota <i>extra</i> para estos trabajos voluntarios

H4 Capacitar a los alumnos para el trabajo en grupo.

Línea de acción	Cuándo	Quién	Medios	Evaluación
La práctica obligatoria debe ser realizada en grupos de trabajo de 3 a 5 personas	Práctica obligatoria de la asignatura	Cientes Alumnos Responsable de la asignatura	Cientes que se presten a ser entrevistados por los alumnos	Defensa de la práctica
Realización de talleres en parte de las horas prácticas de la asignatura donde los alumnos se organizan en grupos para resolver un problema utilizando técnicas estructuradas y orientadas a objeto. Posteriormente uno de los grupos expondrá su solución y se debatirá sobre ella	Parte práctica de la asignatura	Grupos de alumnos Responsable de la asignatura	Enunciados con casos prácticos Transparencias	Informes realizados por los alumnos Comentarios de los alumnos

A.3.2.5 Criterios de evaluación de la asignatura

- a) Parte de Teoría (50% de la nota final)
- Un examen final en el mes de febrero.
 - Un examen final en el mes de septiembre.

b) Parte Práctica (50% de la nota final)

- ERS y prototipo realizado en grupos de trabajo.

Se realizará una defensa individualizada de dicho trabajo.

La parte práctica se guardará hasta la convocatoria de septiembre, pero nunca para futuros cursos.

c) Entre 0,5 y 0,75 puntos sumados a la nota de la parte práctica por la defensa y posterior redacción de un informe sobre los supuestos tratados en los talleres prácticos.

d) Trabajos voluntarios presentados (de 0,5 a 1,5 puntos sumados a la nota conseguida en los apartados anteriores, siempre que en los apartados anteriores se obtenga la calificación mínima exigida)

e) Fórmula para la obtención de la calificación final.

Si (Teoría $\geq 4,75$) y (Práctica ≥ 5.0)

$$\text{Nota Final} = (\text{Teoría} * 0,5) + ((\text{Práctica} + \text{Nota Talleres}) * 0,5) + \text{Nota trabajos}$$

Sino

\emptyset

Fin si

A.3.2.6 Bibliografía básica de referencia

En este apartado se va a incluir aquellas referencias que el alumno debe consultar para completar el material que el responsable facilita para seguir las clases (transparencias). Estos títulos han sido elegidos en función de tres factores: *su adecuación a los contenidos, su disponibilidad en la biblioteca y el idioma (primando, si fuera posible títulos en español)*.

Existen otras referencias que se han manejado por el responsable de la asignatura para preparar los temas, pero éstas, al ser más específicas, no son incluidas en este apartado, aunque aparecen citadas en las transparencias y son recomendadas como otras lecturas en muchas ocasiones.

 **Booch, G., Rumbaugh, J., Jacobson, I.** “*El Lenguaje Unificado de Modelado*”. Addison-Wesley, 1999.

Este libro es la traducción del libro de estos mismos autores “*The Unified Modeling Language User Guide*”, Addison-Wesley, 1999. De la trilogía de libros publicados por los tres máximos responsables del lenguaje modelado UML, éste es el que se corresponde con la explicación detallada de la notación de este lenguaje de modelado. Es un libro de consulta obligada para el tema 6, aunque su primera parte de introducción, especialmente el capítulo 1, se puede utilizar en el primer tema de la parte teórica.

-  **Meyer, B.** “*Construcción de Software Orientado a Objetos*”. 2ª Edición. Prentice Hall, 1999.

Corresponde a la traducción del libro “*Object-Oriented Software Construction*”, 2nd Edition, Prentice-Hall, 1997. Esta segunda edición es más que una simple actualización de la primera edición, con una ampliación de conceptos más que considerable. Aunque es un libro que se adecua más a una asignatura de programación orientada a objetos, los contenidos de los tres primeros capítulos se adaptan muy bien a la asignatura de Ingeniería del Software, especialmente el Capítulo 3 dedicado a la modularidad, que es una referencia importante para el Tema 4.

-  **Piattini, M. G., Calvo-Manzano, J. A., Cervera, J., Fernández, L.** “*Análisis y Diseño Detallado de Aplicaciones Informáticas de Gestión*”. Ra-ma, 1996.

Libro de introducción a la Ingeniería del Software, centrado en la construcción de software de gestión. Consta de tres partes: Sistemas de Información (capítulos 1 y 2), El Proceso de Desarrollo de Software (capítulos 3-16) y Tecnología (capítulos 17-19).

Es un libro de consulta que se puede utilizar en todos los temas de la parte teórica de la asignatura, aunque con una mayor profusión en los temas 1, 2, 4 y 5.

-  **Pressman, R. S.** “*Ingeniería del Software: Un Enfoque Práctico*”. 4ª Edición. McGraw-Hill, 1998.

Es uno de los mejores libros de referencia y consulta para la práctica totalidad de los temas relacionados con la Ingeniería del Software. Corresponde a la traducción del libro del mismo autor “*Software Engineering: A Practitioner’s Approach*”, 4th Edition, McGraw-Hill, 1997. En este texto se realiza una amplia exposición de prácticamente todos los temas relacionados con la Ingeniería del Software desde la doble perspectiva del producto y el proceso. Es de destacar el conjunto de referencias tanto bibliográficas como de recursos disponibles en Internet de cada uno de los temas tratados. Se considera un texto imprescindible de referencia para esta asignatura. En este sentido hay que señalar que en el prólogo de esta edición española se indica que explica todos los temas incluidos en la asignatura o materia troncal Ingeniería del Software de los Planes de Estudios de las universidades españolas.

En lo que respecta a la asignatura, se adecua como una referencia especialmente interesante en los temas 1, 2, 4, 5, 7 y 8.

-  **Rational Software Corporation.** “*The Unified Modeling Language Documentation Set 1.1*”. <http://www.rational.com>. September 1997.

Esta documentación corresponde al texto completo de la propuesta de UML realizada por Rational al OMG. Esta documentación consta de un metamodelo formal, una notación y su semántica. En el metamodelo se realiza una exposición de los conceptos e ideas, de la sintaxis y de la semántica de UML. En la notación se describen los diferentes tipos de diagramas: de casos de uso, de clases, de secuencias, de colaboración, de estados, de actividad, de componentes y de despliegue. En el documento de semántica se establece la semántica de los elementos de UML utilizando un subconjunto de la notación. Esta documentación es básica para el seguimiento de la Unidad Didáctica III de la asignatura.

-  **Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., Lorensen, W.** “*Modelado y Diseño Orientados a Objetos. Metodología OMT*”. 2ª Reimpresión. Prentice Hall, 1998.

Corresponde a la traducción del libro de los mismos autores “*Object Oriented Modeling and Design*”, Prentice Hall, 1991. En el libro se presenta una de las primeras metodologías orientadas al objeto que fueron desarrolladas, OMT. En esta metodología de desarrollo de sistemas se considera que el modelado orientado al objeto consta de tres partes: el modelo objeto, el modelo funcional y el modelo dinámico. A pesar de ser una metodología superada en muchos aspectos, desde el punto de vista de la docencia a impartir tiene como ventaja que toma algunas de las herramientas de las metodologías clásicas, por ejemplo el DFD para el elaborar el modelo funcional, lo que permite realizar una transición suave desde los métodos clásicos a los de la Orientación al Objeto. La tercera parte del libro aborda los aspectos de paso del diseño a la implementación en relación con los lenguajes de programación y las bases de datos relacionales. Incluye la descripción de tres casos de estudio. Este libro se utiliza principalmente como referencia para los temas 1 y 6.

-  **Sommerville, I.** “*Software Engineering*”. 5th Edition, Addison-Wesley, 1996.

Esta es la última edición de uno de los libros más difundidos sobre Ingeniería del Software. Explica un amplio conjunto de técnicas y muestra como pueden ser aplicadas en la práctica de la Ingeniería del Software. Los temas están agrupados de acuerdo a las actividades del proceso de desarrollo y mantenimiento del software. Respecto a las versiones anteriores hay que destacar la incorporación de temas como re-ingeniería o la ampliación de temas como CASE y gestión y evolución del software. El autor, profesor de la asignatura de Ingeniería del Software en una universidad inglesa, se ha preocupado de adaptar los contenidos a la recomendación curricular ACM/IEEE-CS91 para los módulos SE2 a SE5. Además, puede obtenerse material suplementario de apoyo a la docencia en Internet. El nivel de algunos de los capítulos es superior al exigido a los alumnos de I.T.I.S, pero se considera uno de los libros imprescindibles de consulta en la materia.

 **Yourdon, E.** “*Análisis Estructurado Moderno*”. Prentice-Hall Hispanoamericana, 1993.

Corresponde a la traducción del texto del mismo autor “*Modern Structured Analysis*”, Prentice-Hall, 1989. En él, Yourdon revisa las teorías del análisis estructurado desarrolladas por DeMarco en el año 1979. Es un libro con un diseño muy didáctico y adaptado a un curso de análisis estructurado. Tiene algunos capítulos, que en su momento constituyeron un cierto avance en las metodologías estructuradas y que, a pesar del paso de los años, mantienen su vigencia. Es de destacar el capítulo en el que se realiza un estudio general de sistemas, la propuesta de los diferentes modelos a realizar de un sistema y los casos prácticos desarrollados en los apéndices. Sigue constituyendo uno de los libros de referencia obligada para los cursos de Ingeniería del Software. En esta asignatura es una referencia complementaria en los temas 1 y 2, siendo la referencia fundamental del tema 3.

A.3.3 Programación Orientada a Objetos

El principal objetivo de esta asignatura es transmitir al alumno los conocimientos necesarios para el diseño e implementación de aplicaciones software bajo el paradigma objetual.

Aunque se va a utilizar un lenguaje orientado a objetos (C++) para que el alumno pueda llevar al terreno práctico los conceptos transmitidos en esta asignatura, se pretende una exposición de los mismos desde un plano más abstracto, más centrado en el diseño, de forma que el alumno pueda aplicarlos en su vida laboral con independencia del entorno de desarrollo orientado a objetos con el que se encuentre.

Este objetivo general se corresponde con los objetivos concretos **T7, T9, P2, P5, H1, H2, H3 y H4** de la unidad docente.

Para ver satisfacer dichos objetivos, se va a detallar los prerrequisitos de esta asignatura, las líneas de acción a seguir en cada uno de ellos, el temario teórico/práctico de la asignatura, los criterios de evaluación y la bibliografía básica de consulta recomendada.

A.3.3.1 Ficha de la asignatura

Esta asignatura se considera una extensión de la asignatura Ingeniería del Software, en la que se amplía la Unidad Docente III de ésta con aspectos de diseño y programación orientada objetos. De hecho, la asignatura de Programación Orientada a Objetos se imparte en el segundo cuatrimestre del tercer curso de I.T.I.S, después de la finalización de la asignatura de Ingeniería del Software.

Asignatura	<i>Programación orientada a objetos (optativa)</i>
Créditos	3T + 3P
Estudios	I.T.I.S
Plan	B.O.E de 4-11-1997
Curso	3º
Cuatrimestre	2º
Responsable	Francisco José García Peñalvo (fgarcia@usal.es)
Página web de la asignatura	http://tejo.usal.es/~fgarcia/docencia.html

Tabla A.10. Programación Orientada a Objetos

A.3.3.2 Prerrequisitos

El alumno debe estar familiarizado con la teoría así como con la práctica del diseño y codificación en lenguajes procedurales (por ejemplo C). Estos conocimientos deben adquirirse en las asignaturas relacionadas con la programación en el primer y segundo curso **Algoritmia, Programación, Laboratorio de Programación y Estructuras de Datos**.

Es deseable que el alumno conozca la problemática de las interfaces de usuario y esté familiarizado con la problemática de construir interfaces gráficas de usuario. Estos aspectos deben tratarse en la asignatura **Interfaces Gráficas** del segundo curso.

El alumno debe conocer los fundamentos básicos que rigen un proyecto software, estar familiarizado con la elicitación y especificación de requisitos y manejar los principios de la Ingeniería del Software orientada a objetos, en concreto el lenguaje de modelado UML y conocimientos generales de una metodología orientada a objetos del tipo OMT. Todos estos conceptos le son transmitidos al alumno en la asignatura de tercer curso **Ingeniería del Software**.

A.3.3.3 Temario teórico/práctico

Presentación de la asignatura (1 Hora)

Unidad Docente I: Conceptos básicos (4 Horas)

Tema 1. Lenguajes de programación orientados a objetos (2 Horas)

Tema 2. Orientación a objeto y reutilización del software (2 Horas)

Unidad Docente II: Diseño orientado a objetos básico (14 Horas)

Tema 3. Técnicas básicas de diseño orientado a objetos (8 Horas)

Tema 4. Genericidad (4 Horas)

Tema 5. Manejo de excepciones (2 Horas)

Unidad Docente III: Diseño orientado a objetos avanzado (11 Horas)

Tema 6. Patrones de diseño (8 Horas)

Tema 7. Diseño por contrato (3 Horas)

Tabla A.11. Programa de teoría de Programación Orientada a Objetos (3 créditos)

C++ (18 Horas)

Tema C++ 1. Modelo objeto en C++ (10 Horas)

Tema C++ 2. STL (8 Horas)

Taller (2 Horas)

Tarjetas CRC

Prácticas libres (10 Horas)**Práctica obligatoria**

Realización de una aplicación utilizando técnicas orientadas a objeto

Tabla A.12. Programa de prácticas de Programación Orientada a Objetos (3 créditos)**Actividades Docentes Complementarias**

- Seminarios impartidos sobre temas específicos
- Trabajos voluntarios realizados por los alumnos
- Conferencias invitadas
- *Workshop* de trabajos realizados por los alumnos sobre temas de Ingeniería del Software y objetos

En la Tabla A.13 se presenta la correspondencia existente entre el temario y los objetivos perseguidos en esta asignatura.

Elemento Docente	Objetivos
Tema 1	T7
Tema 2	T7
Tema 3	T7, T9, P2
Tema 4	T7, T9, P2
Tema 5	T7, T9, P2
Tema 6	T7, T9, P2
Tema 7	T7, T9, P2
Tema C++ 1	P5
Tema C++ 2	P5
Taller CRC	P2, H1, H2, H3, H4
Práctica obligatoria	P2, P5, H1, H2, H3, H4

Tabla A.13. Correspondencia entre el temario teórico/práctico y los objetivos de la asignatura**A.3.3.4 Líneas de acción**

Para cada uno de los objetivos de la unidad docente que debe satisfacerse en esta asignatura se van a identificar las líneas de acción a seguir. Cada línea de acción va a definirse mediante los

cuatro apartados indicados en [García et al., 1999a]: *cuándo se lleva a cabo, quién es el responsable de realizarla, qué medios son necesarios y cómo se evaluarán los resultados de dicha línea de acción.*

T7 Método de análisis/diseño orientado a objetos.

Línea de acción	Cuándo	Quién	Medios	Evaluación
Desarrollo de aplicaciones bajo el paradigma objetual, centrando la atención en las fases de diseño e implementación	Clases teóricas y prácticas	Responsable de la asignatura	Bibliografía Transparencias Lenguaje de programación orientado a objetos	Examen teórico Prácticas Opinión de los alumnos

T9 Estudio y comprensión de los fundamentos del diseño de sistemas software.

Línea de acción	Cuándo	Quién	Medios	Evaluación
Aplicación de principios y técnicas del diseño orientado a objetos	Parte teórica	Responsable de la asignatura	Transparencias Bibliografía básica	Examen Práctica obligatoria
Introducción de técnicas avanzadas de diseño orientado a objetos	Parte teórica	Responsable de la asignatura	Transparencias Bibliografía básica	Examen Práctica obligatoria

P2 Aplicar de forma práctica los conceptos teóricos de Orientación a Objetos.

Línea de acción	Cuándo	Quién	Medios	Evaluación
Realización de una práctica obligatoria en la que se cree una aplicación utilizando técnicas orientadas a objeto	Parte práctica de la asignatura	Grupos de alumnos	Caso práctico	Defensa de la práctica

P5 Programación orientada a objetos.

Línea de acción	Cuándo	Quién	Medios	Evaluación
Estudio de un lenguaje orientado a objetos	Clases prácticas	Responsable de la asignatura Alumnos	Bibliografía Pequeños supuestos prácticos	Defensa de la práctica
Realización de una práctica obligatoria en la que se cree una aplicación utilizando técnicas orientadas a objeto	Parte práctica de la asignatura	Grupos de alumnos	Caso práctico	Defensa de la práctica

H1 Mejora de la expresión oral.

Línea de acción	Cuándo	Quién	Medios	Evaluación
Defensa oral de la práctica obligatoria	Defensa de la práctica	Grupos de alumnos Responsable de la asignatura	Medios audiovisuales	Defensa de la práctica
Promover seminarios realizados por los alumnos como resultado de trabajos voluntarios	A lo largo de la asignatura	Grupos de alumnos	Bibliografía Recursos de Internet	Nota <i>extra</i> para estos trabajos voluntarios

H2 Mejora en la redacción de documentos técnicos.

Línea de acción	Cuándo	Quién	Medios	Evaluación
Documentación de la práctica obligatoria	Defensa de la práctica	Grupos de alumnos Responsable de la asignatura	Medios audiovisuales	Defensa de la práctica
Memoria de los trabajos voluntarios	A lo largo de la asignatura	Grupos de alumnos	Bibliografía Recursos de Internet	Nota <i>extra</i> para estos trabajos voluntarios

H3 Potenciación de la capacidad del alumnos para la búsqueda de información (manejo de fuentes bibliográficas, Internet, foros de discusión...).

Línea de acción	Cuándo	Quién	Medios	Evaluación
Completar las transparencias utilizadas en clase con la bibliografía recomendada	A lo largo de la asignatura	Alumnos	Biblioteca	Examen
Promover seminarios realizados por los alumnos como resultado de trabajos voluntarios	A lo largo de la asignatura	Grupos de alumnos	Bibliografía Recursos de Internet	Nota <i>extra</i> para estos trabajos voluntarios

H4 Capacitar a los alumnos para el trabajo en grupo.

Línea de acción	Cuándo	Quién	Medios	Evaluación
La práctica obligatoria debe ser realizada en grupos de trabajo de 2 ó 3 personas	Práctica obligatoria de la asignatura	Alumnos Responsable de la asignatura	Supuestos prácticos	Defensa de la práctica

A.3.3.5 Criterios de evaluación de la asignatura

a) Parte de Teoría (50% de la nota final)

- Un examen final.

b) Parte Práctica (50% de la nota final)

- Realización de un supuesto práctico por parejas.

Se realizará una defensa de dicho trabajo.

La parte práctica se guardará hasta la convocatoria de septiembre, pero nunca para futuros cursos.

- c) Trabajos voluntarios presentados (de 0,5 a 1,5 puntos sumados a la nota conseguida en los apartados anteriores, siempre que en los apartados anteriores se obtenga la calificación mínima exigida)
- d) Fórmula para la obtención de la calificación final.

Si (Teoría $\geq 4,75$) y (Práctica ≥ 5.0)

$$\text{Nota Final} = (\text{Teoría} * 0,5) + (\text{Práctica} * 0,5) + \text{Nota trabajos}$$

Sino

\emptyset

Fin si

A.3.3.6 Bibliografía básica de referencia

En este apartado se va a incluir aquellas referencias que el alumno debe consultar para completar el material que el responsable facilita para seguir las clases (transparencias). Estos títulos han sido elegidos en función de tres factores: *su adecuación a los contenidos, su disponibilidad en la biblioteca y el idioma (primando, si fuera posible títulos en español)*.

Existen otras referencias que se han manejado por el responsable de la asignatura para preparar los temas, pero éstas, al ser más específicas, no son incluidas en este apartado, aunque aparecen citadas en las transparencias y son recomendadas como otras lecturas en muchas ocasiones.

 **Booch, G., Rumbaugh, J., Jacobson, I.** “*El Lenguaje Unificado de Modelado*”. Addison-Wesley, 1999.

Este libro es la traducción del libro de estos mismos autores “*The Unified Modeling Language User Guide*”, Addison-Wesley, 1999. De la trilogía de libros publicados por los tres máximos responsables del lenguaje modelado UML, éste es el que se corresponde con la explicación detallada de la notación de este lenguaje de modelado.

Es un libro de consulta para repasar o aclarar cualquier duda que pudiera surgir sobre UML a lo largo del desarrollo de la asignatura.

 **Deitel, H. M., Deitel, P. J.** “*Como Programar en C/C++*”. 2ª Ed. Prentice Hall, 1995.

Traducción del libro de los mismos autores “*C How to Program*”, 2nd Edition, Prentice-Hall, 1994. Es un libro con dos partes bien diferenciadas, los primeros catorce capítulos están dedicados al lenguaje C, mientras que los siete últimos a C++. Es por tanto un libro de referencia general que, además de tratar temas de C++, sirve como consulta para posibles dudas sobre lenguaje C.

-  **Gamma, E., Helm, R., Johnson, R., Vlissides, J.** “*Design Patterns. Elements of Reusable Object-Oriented Software*”. Addison-Wesley, 1995.

Este libro es un clásico del diseño orientado a objeto y el máximo responsable de la difusión de los patrones de diseño en la comunidad de la Orientación a Objetos. El libro es un catálogo de 23 patrones de diseño divididos en tres categorías: patrones de creación, patrones estructurales y patrones de comportamiento.

Es la referencia más importantes de la parte teórica de la asignatura, así como el texto base del tema 5.

-  **Glass, G., Schuchert, B.** “*The STL <Primer>*”. Prentice-Hall, 1996.

Libro que hace un recorrido muy detallado por la biblioteca de plantillas estándar de C++ STL (*Standard Template Library*). El libro está dividido en cinco secciones: *Presentación de STL, Utilizando STL, Catálogo de las clases de STL, Catálogo de Algoritmos y Apéndices*.

Este libro sirve como referencia complementaria en el tema 3 de la parte teórica dedicado a la genericidad, y como referencia base en el tema 2 de prácticas dedicado a la biblioteca STL.

-  **Meyer, B.** “*Construcción de Software Orientado a Objetos*”. 2ª Edición. Prentice Hall, 1999.

Corresponde a la traducción del libro “*Object-Oriented Software Construction*”, 2nd Edition, Prentice-Hall, 1997. Esta segunda edición es más que una simple actualización de la primera edición, con una ampliación de conceptos más que considerable. Es un libro sobre el cual se podía construir una asignatura de programación orientada a objetos, especialmente si elige Eiffel como lenguaje de programación para la parte práctica.

En la asignatura de Programación Orientada a Objetos va a ser una referencia de consulta más que una guía de los contenidos, aunque es la referencia clave en el tema 6.

-  **Rational Software Corporation.** “*The Unified Modeling Language Documentation Set 1.1*”. <http://www.rational.com>. September 1997.

Esta documentación corresponde al texto completo de la propuesta de UML realizada por Rational al OMG. Esta documentación consta de un metamodelo formal, una notación y su semántica. En el metamodelo se realiza una exposición de los conceptos e ideas, de la sintaxis y de la semántica de UML. En la notación se describen los diferentes tipos de diagramas: de casos de uso, de clases, de secuencias, de colaboración, de estados, de actividad, de componentes y de despliegue. En el documento de semántica se establece la semántica de los elementos de UML utilizando un subconjunto de la notación.

Aunque el estudio de UML no es un objetivo de esta asignatura, sino de la asignatura de Ingeniería del Software, va a ser el lenguaje de modelado que se utilice para representar los diseños a discutir en la asignatura.

-  **Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., Lorensen, W.** “*Modelado y Diseño Orientados a Objetos. Metodología OMT*”. Prentice Hall, 2ª reimpresión, 1998.

Corresponde a la traducción del libro de los mismos autores “*Object Oriented Modeling and Design*”, Prentice Hall, 1991. En el libro se presenta una de las primeras metodologías orientadas al objeto que fueron desarrolladas, OMT. En esta metodología de desarrollo de sistemas se considera que el modelado orientado al objeto consta de tres partes: el modelo objeto, el modelo funcional y el modelo dinámico. A pesar de ser una metodología superada en muchos aspectos, desde el punto de vista de la docencia a impartir tiene como ventaja que toma algunas de las herramientas de las metodologías clásicas, por ejemplo el DFD para el elaborar el modelo funcional, lo que permite realizar una transición suave desde los métodos clásicos a los de la Orientación al Objeto. La tercera parte del libro aborda los aspectos de paso del diseño a la implementación en relación con los lenguajes de programación y las bases de datos relacionales. Incluye la descripción de tres casos de estudio.

Este libro constituye una referencia de consulta válida en todos los temas de la parte teórica de la asignatura, especialmente por su excelente exposición del modelo objeto de OMT.

-  **Stroustrup, B.** “*El Lenguaje de Programación C++*”. 3ª Ed. Addison-Wesley. 1998.

Este libro es la traducción de la obra del mismo autor “*The C++ Programming Language*”, 3rd Edition, Addison-Wesley, 1997. Es el libro de referencia por excelencia cuando de C++ se trata, al ser el autor del mismo el padre de este lenguaje de programación orientado a objetos. En el momento de escribir esta tercera edición el proceso de estandarización de C++ se encontraba en su recta final, no previéndose cambios significativos, por lo que se puede considerar que el libro contiene una presentación de C++ actualizada y validada por el comité de estandarización de este lenguaje. El libro se divide en 6 partes: *Introducción* (capítulos 1-3), *Tipos Predefinidos en C++* (capítulos 4-9), *Programación Orientada a Objetos y Genérica con C++* (capítulos 10-15), *Biblioteca C++ Estándar* (capítulos 16-22), *Diseño y Desarrollo de Software* (capítulos 23-25) y *Apéndices*.

Es un libro que se ajusta perfectamente a la práctica de la asignatura, en especial las tres primeras partes.

A.3.4 Proyecto I.T.I.S.

La realización del proyecto de final de carrera se convierte en una de las primeras oportunidades con las que cuenta un estudiante de una Ingeniería Técnica en Informática para aplicar de forma integrada los conocimientos teóricos y prácticos obtenidos en el conjunto de asignaturas cursadas en la carrera.

El objetivo principal de éste casa con los objetivos **P9**, **H1**, **H2** y **H3** de la unidad docente (en los casos en que el proyecto se lleve a cabo por más de una persona tendría sentido incluir el objetivo **H4**).

En un proyecto de final de carrera el cometido y la bibliografía será establecida por el tutor o tutores del mismo, aunque como guía de referencia general para la realización y documentación de los proyectos final de carrera se tiene [García et al., 2000], guía que se ajusta especialmente a los proyectos que tienen como cometido la creación de un sistema software.

A.3.5 Análisis de Sistemas

El propósito fundamental de esta asignatura es que los alumnos del segundo ciclo de Ingeniería en Informática profundicen en los conocimientos adquiridos en la asignatura *Ingeniería del Software* de primer ciclo, especialmente en lo que se refiere a métodos de análisis y especificación de requisitos de sistemas orientados a objetos y a técnicas formales de especificación.

Los objetivos de la unidad docente en que se desglosa son los siguientes:

T3, T4, T5, T6, T7, T12, T13, P1, P2, P4, H1, H2, H3 y H4.

Seguidamente se expondrán los prerrequisitos de esta asignatura, necesarios para conseguir los objetivos mencionados, las líneas de acción a seguir en cada uno de ellos, el temario teórico/práctico de la asignatura, los criterios de evaluación y la bibliografía básica de consulta que se recomienda a los alumnos.

A.3.5.1 Ficha de la asignatura

Asignatura	<i>Análisis de Sistemas (troncal)</i>
Créditos	<i>6T + 3P</i>
Estudios	<i>Ingeniería en Informática (2º ciclo)</i>
Plan	<i>B.O.E de 1-7-1999</i>
Curso	<i>1º</i>
Cuatrimestre	<i>1º y 2º (anual)</i>
Responsable	<i>María N. Moreno García (mmg@usal.es)</i>
Página web de la asignatura	http://lisisu02.usal.es/~mmoreno/analisis.html

Tabla A.14. Análisis de Sistemas

A.3.5.2 Prerrequisitos

Además de los prerrequisitos que se establecen en la asignatura Ingeniería del Software, de esta unidad docente, se deben tener conocimientos teóricos y prácticos de los objetivos de la Ingeniería del Software, de los modelos de ciclo de vida del software más utilizados y de los

métodos de análisis y diseño estructurado. Asimismo sería conveniente conocer las bases del desarrollo de software orientado a objetos.

Las asignaturas que tratan estos aspectos son **Ingeniería del Software y Programación Orientada a Objetos**, que se imparten en 3º de ITIS.

A.3.5.3 Temario teórico/práctico

Presentación de la asignatura (1 hora)

Unidad Docente I: Definición del proceso de software (7 horas)

Tema 1. Modelos avanzados de ciclo de vida (3 Horas)

Tema 2. Prototipos (4 Horas)

Unidad Docente II: Métodos de desarrollo (36 horas)

Tema 3. Métodos estructurados. Técnicas avanzadas (6 Horas)

Tema 4. Métodos orientados a objetos (10 Horas)

Tema 5. El lenguaje UML y el proceso unificado (10 Horas)

Tema 6. Técnicas formales de especificación (10 Horas)

Unidad Docente III: Desarrollo de sistemas especiales (12 horas)

Tema 7. Sistemas de tiempo real (6 horas)

Tema 8. Otros sistemas (6 horas)

Unidad Docente IV: Evolución del software (4 horas)

Tema 9. Evolución y mantenimiento del software (4 horas)

Tabla A.15. Programa de teoría de Análisis de Sistemas (6 créditos)

Laboratorio: uso de herramientas CASE (30 horas)

Práctica 1. Análisis y Diseño estructurado de un sistema (15 horas)

Práctica 2. Análisis y Diseño orientado a objetos (15 horas)

Tabla A.16. Programa de prácticas de Análisis de Sistemas (3 créditos)

Actividades Docentes Complementarias

- Seminarios impartidos sobre temas específicos
- Trabajos voluntarios realizados por los alumnos
- Conferencias invitadas
- *Workshop* de trabajos realizados por los alumnos sobre temas de Ingeniería del Software y objetos

En la Tabla A.17 se presenta la correspondencia existente entre el temario y los objetivos perseguidos en esta asignatura.

Elemento Docente	Objetivos
Tema 1	T3, T4, H3
Tema 2	T3, T4, H3
Tema 3	T6, H3
Tema 4	T7, H3
Tema 5	T7, H3
Tema 6	T5, H3
Tema 7	T13, H3
Tema 8	T13, H3
Tema 9	T12, H3
Práctica 1	P1, H1, H2, H4
Práctica 2	P2, H1, H2, H4

Tabla A.17. Correspondencia entre el temario teórico/práctico y los objetivos de la asignatura

A.3.5.4 Líneas de acción

En este apartado se comentarán las líneas de acción a seguir para la consecución de cada uno de los objetivos de la unidad docente. Cada línea de acción va a definirse mediante los cuatro apartados indicados en [García et al., 1999a]: *cuándo se lleva a cabo, quién es el responsable de realizarla, qué medios son necesarios y cómo se evaluarán los resultados de dicha línea de acción.*

T3 Importancia de los requisitos en el ciclo de vida del software.

T4 Elicitación, documentación, especificación y prototipado de los requisitos de un sistema software.

Línea de acción	Cuándo	Quién	Medios	Evaluación
Mostrar la relevancia de los requisitos al estudiar modelos avanzados de ciclo de vida.	En el tema 1 y en las prácticas de la asignatura	Responsable de la asignatura	Transparencias Bibliografía básica	Defensa práctica Examen
Estudiar el prototipado de sistemas como modelo de ciclo de vida y como técnica de verificación y validación de requisitos	En el tema 2 y en las prácticas de la asignatura	Responsable de la asignatura	Transparencias Bibliografía básica	Defensa práctica Examen
Realización del análisis y especificación de requisitos de un sistema haciendo uso de herramientas CASE.	Prácticas de la asignatura	Clientes Alumnos Responsable	Clientes que se presten a ser entrevistados por los alumnos	Defensa de la práctica

T5 Especificaciones formales de requisitos.

Línea de acción	Cuándo	Quién	Medios	Evaluación
Estudio de las principales técnicas formales de especificación de sistemas, incluyendo la especificación de aspectos dinámicos y los métodos formales orientados a objetos	En el tema 6	Responsable de la asignatura	Transparencias Bibliografía básica	Ejercicios prácticos Examen

T7 Métodos de análisis/diseño orientado a objetos.

Línea de acción	Cuándo	Quién	Medios	Evaluación
Profundización en los métodos orientados a objetos. Estudio del Lenguaje Unificado de Modelado y del Proceso Unificado	Unidad docente II	Responsable de la asignatura	Transparencias Bibliografía básica	Ejercicios prácticos Defensa de la práctica Examen
Realización del análisis y diseño de un sistema orientado a objetos con la ayuda de una herramienta CASE	Práctica 2 de la asignatura	Cientes Alumnos Responsable de la asignatura	Cientes que se presten a ser entrevistados por los alumnos	Defensa de la práctica

T12 Conceptos, métodos, procesos y técnicas destinadas al mantenimiento y evolución de los sistemas software.

Línea de acción	Cuándo	Quién	Medios	Evaluación
Estudiar los procesos relacionados con la modificación de un producto software después de su entrega.	Tema 9	Responsable de la asignatura	Transparencias Bibliografía básica	Examen

T13 Conocimiento sobre el uso de la Ingeniería del Software en dominios de aplicación específicos.

Línea de acción	Cuándo	Quién	Medios	Evaluación
Estudio de las particularidades que presenta el desarrollo de sistemas de tiempo real y de las técnicas más apropiadas para el modelado de este tipo de sistemas	Unidad docente III	Responsable de la asignatura	Transparencias Bibliografía básica	Examen
Análisis de las características de los sistemas distribuidos y su relación con la aplicación de métodos de Ingeniería del Software.	Unidad docente III	Responsable de la asignatura	Transparencias Bibliografía básica	Examen
Examen de otros sistemas no convencionales como sistemas expertos, sistemas para el soporte a las decisiones, etc.	Unidad docente III	Responsable de la asignatura	Transparencias Bibliografía básica	Examen

- P1** Aplicar de forma práctica los conceptos teóricos sobre el desarrollo estructurado.
- P2** Aplicar de forma práctica los conceptos teóricos de Orientación a Objetos.
- P4** Utilización de herramientas CASE para la gestión y desarrollo de sistemas software.

Línea de acción	Cuándo	Quién	Medios	Evaluación
Con la ayuda de herramientas CASE, los alumnos realizarán el análisis y diseño de sendos sistemas de software aplicando, respectivamente, un método estructurado y otro orientado a objetos.	Clases prácticas	Responsable de la asignatura Alumnos Clientes	Transparencias Bibliografía Herramientas CASE Clientes que se presten a ser entrevistados por los alumnos	Defensa de la práctica

- H1** Mejora de la expresión oral.

Línea de acción	Cuándo	Quién	Medios	Evaluación
Promover los debates sobre las soluciones presentadas en los ejercicios realizados en clase	Clase de teoría	Responsable de la asignatura Grupos de alumnos	Pizarra Medios audiovisuales	Estudio del seguimiento de los ejercicios Opinión alumnos
Defensa oral de las prácticas	Defensa de la práctica	Grupos de alumnos Responsable de la asignatura	Medios audiovisuales	Defensa de la práctica
Promover seminarios realizados por los alumnos como resultado de trabajos voluntarios	A lo largo de la asignatura	Grupos de alumnos	Bibliografía Recursos de Internet	Nota <i>extra</i> para estos trabajos voluntarios

- H2** Mejora en la redacción de documentos técnicos.

Línea de acción	Cuándo	Quién	Medios	Evaluación
Propuesta de ejercicios de aplicación de los conceptos teóricos	Durante las clases de teoría	Responsable de la asignatura Grupos de alumnos	Transparencias Pizarra	Evaluación de los ejercicios entregados
Documentación de las prácticas	Defensa de las prácticas	Grupos de alumnos Responsable de la asignatura	Medios audiovisuales	Defensa de la práctica
Memoria de los trabajos voluntarios	A lo largo de la asignatura	Grupos de alumnos	Bibliografía Recursos de Internet	Nota <i>extra</i> para estos trabajos voluntarios

H3 Potenciación de la capacidad del alumno para la búsqueda de información (manejo de fuentes bibliográficas, Internet, foros de discusión...).

Línea de acción	Cuándo	Quién	Medios	Evaluación
Completar las transparencias utilizadas en clase con la bibliografía recomendada	A lo largo de la asignatura	Alumnos	Biblioteca	Examen
Promover seminarios realizados por los alumnos como resultado de trabajos voluntarios	A lo largo de la asignatura	Grupos de alumnos	Bibliografía Recursos de Internet	Nota <i>extra</i> para estos trabajos voluntarios

H4 Capacitar a los alumnos para el trabajo en grupo.

Línea de acción	Cuándo	Quién	Medios	Evaluación
Realización de las prácticas en grupos de 2 personas	Durante las clases prácticas	Cientes Alumnos Responsable de la asignatura	Cientes que se presten a ser entrevistados por los alumnos	Defensa de la práctica

A.3.5.5 Criterios de evaluación de la asignatura

a) Parte de Teoría

- Un examen final en el mes de junio.
- Un examen final en el mes de septiembre.

b) Parte Práctica

- En junio la evaluación de esta parte de la asignatura se realizará mediante la valoración de los trabajos presentados. La nota será la media de las obtenidas en cada uno de los trabajos prácticos. La defensa de dichos trabajos se realizará de forma individualizada.
- En septiembre se realizará un examen en el que el alumno tendrá que resolver un supuesto práctico con la ayuda de una herramienta CASE.

c) Trabajos voluntarios presentados (de 0,5 a 1,5 puntos sumados a la nota conseguida en los apartados anteriores, siempre que en los apartados anteriores se obtenga la calificación mínima exigida).

d) En el caso de haber superado la parte teórica y la práctica, se aplicará la siguiente fórmula para la obtención de la calificación final:

$$\text{Nota Final} = ((\text{Teoría} * 0,6) + (\text{Práctica} * 0,3)) * 10/9 + \text{Nota trabajos}$$

A.3.5.6 Bibliografía básica de referencia

En el apartado de bibliografía se recogen aquellas referencias que pueden ayudar al alumno a completar los conocimientos impartidos en las clases. La elección de las mismas, como ocurre en otras asignaturas de la unidad, se ha basado en tres aspectos: *su adecuación a los contenidos, su disponibilidad en la biblioteca y el idioma (primando, si fuera posible títulos en español)*.

- 📖 **Booch, G., Rumbaugh, J., Jacobson, I.** “*El Lenguaje Unificado de Modelado*”. Addison-Wesley, 1999.

El texto en castellano corresponde a la traducción del libro de estos mismos autores “*The Unified Modeling Language User Guide*”, Addison-Wesley, 1999. Es uno de los libros de la trilogía publicada por los creadores del lenguaje modelado UML, en él se proporciona una referencia de uso de las características específicas de UML, centrándose fundamentalmente la notación gráfica. Es indispensable su consulta en el tema 4 y fundamentalmente en el 5.

- 📖 **Graham, I.** “*Métodos Orientados a Objetos*”. 2ª Ed. Addison-Wesley/Díaz de Santos, 1996.

Traducción del libro del mismo autor “*Object-Oriented Methods*” 2nd ed. Addison-Wesley, 1994. En un libro introductorio a la Orientación al Objeto. Cubre de forma amplia y correcta todos los aspectos de la Orientación al Objeto: conceptos básicos, lenguajes de programación, bases de datos, aplicaciones y métodos. En los capítulos de métodos de análisis y de diseño realiza una exposición y un examen de los de mayor difusión, cubriendo prácticamente todas las escuelas existentes. Incorpora un capítulo interesante relacionado con la inteligencia artificial y la teoría de los conjuntos difusos. Se considera un buen libro de referencia donde se puede adquirir una panorámica amplia de la Orientación al Objeto.

- 📖 **Hatley, D. J., Pirbhai, I. A.** “*Strategies for Real-Time System Specification*”. Dorset House Pub. Co., 1987.

Se describen los métodos para la especificación de los requisitos y el diseño de sistemas de tiempo real basados en computadora. Los métodos propuestos integran las herramientas del análisis estructurado, fundamentalmente el DFD, para la construcción del modelo de requisitos y del modelo arquitectónico. Partiendo de la base de los DFD propone una herramienta de modelado para los procesos de control los CFD (*Control Flow Diagrams*), y otra para la representación de la configuración física del sistema los ACF (*Architecture Flow Diagrams*). Los métodos y la construcción de modelos están apoyados por un ejemplo que sirve de hilo conductor a lo largo del libro. Aunque el método propuesto para la asignatura de Ingeniería del Software I va a seguir más la propuesta de Ward & Mellor, este libro tiene aportaciones muy interesantes en el modelado de sistemas de tiempo real, sobre todo en el modelado arquitectónico.

-  **Henderson-Sellers, B.** “*A Book of Object-Oriented Knowledge*”. 2nd ed. Prentice Hall PTR, 1997.

Es una introducción básica al enfoque orientado al objeto en la Ingeniería del Software. El libro describe las ideas básicas relacionadas con el análisis, el diseño y la implementación e incluye un buen conjunto de referencias para el estudio con profundidad de cada uno de esos temas. Aporta en cada tema el diseño de las transparencias que pueden ser utilizadas en la explicación del mismo. Es una buena presentación de los conceptos y los métodos de la Orientación al Objeto.

-  **Jacobson, I.; Booch, G., Rumbaugh, J.** “*The Unified Software Development Process*”. Addison-Wesley Object Technology Series, 1999.

Otro de los libros de la trilogía de los creadores de UML, en el que se ofrece una referencia completa del proceso de desarrollo unificado que se adapta a UML. Su consulta puede ser beneficiosa en el estudio de los temas 4 y 5.

-  **Meyer, B.** “*Construcción de Software Orientado a Objetos*”. 2^a Edición. Prentice Hall, 1999.

Esta traducción del libro “*Object-Oriented Software Construction*”, 2nd Edition, Prentice-Hall, 1997 en su segunda edición supone una extensa revisión respecto a la primera. Se considera una referencia importante para el Tema 4.

-  **Piattini, M. G., Calvo-Manzano, J. A., Cervera, J., Fernández, L.** “*Análisis y Diseño Detallado de Aplicaciones Informáticas de Gestión*”. Ra-ma. 1996.

Libro recomendado en la asignatura Ingeniería del Software de primer ciclo que puede ser útil también en el segundo para repasar y profundizar en algunos aspectos relacionados con: *Sistemas de Información* (capítulos 1 y 2), *El Proceso de Desarrollo de Software* (capítulos 3-16) y *Tecnología* (capítulos 17-19) que se tratan en las dos primeras unidades docentes de la asignatura.

-  **Pressman, R. S.** “*Ingeniería del Software: Un Enfoque Práctico*”. 4^a Edición. McGraw-Hill. 1998.

Este libro, aunque ya ha sido manejado ampliamente por los alumnos en primer ciclo, puede ser igualmente útil en las asignaturas de Ingeniería del Software de segundo ciclo ya que abarca la mayoría de los temas relacionados con esta materia, tratándolos con bastante profundidad. Cada tema se complementa además con un gran número de referencias de otras fuentes de información sobre el tema, incluyendo recursos disponibles en Internet. Corresponde a la traducción del libro del mismo autor “*Software Engineering: A Practitioner’s Approach*”, 4th Edition, McGraw-Hill, 1997. Se considera un texto imprescindible de referencia para esta asignatura siendo una referencia interesante en la práctica totalidad de los temas.

- 📖 **Sommerville, I.** “*Software Engineering*”. 5th Edition, Addison-Wesley, 1996.

En esta última edición de la obra de Sommerville, éste realiza una extensa revisión de la anterior edición incluyendo nuevos temas y modificando sustancialmente algunos de los existentes como los dedicados a la tecnología CASE y evolución del software. Los contenidos se adaptan a la recomendación curricular ACM/IEEE-CS91 para los módulos SE2 a SE5. Se considera un libro de consulta importante para la práctica totalidad de la asignatura.

- 📖 **Rational Software Corporation.** “*OMG Unified Modeling Language Specification. Version 1.3*”. <http://www.rational.com>. Abril, 1999.

Esta documentación corresponde a la última versión del estándar UML en la que se hace una revisión de la versión 1.1, enfocada fundamentalmente a la resolución de inconsistencias y clarificación de ambigüedades. Esta documentación incluye un apartado dedicado a la semántica de UML en el que se presenta la semántica de los elementos de los modelos de objetos estáticos y dinámicos, además se describe de una manera semi-formal el metamodelo. En otra sección del documento denominada *guía de la notación UML* se ofrece la notación propuesta para la representación gráfica de los modelos, compuesta de nueve tipos de diagramas diferentes. La consulta de esta documentación es imprescindible en el tema 5.

- 📖 **Rumbaugh, J., Jacobson, I., Booch, G.** “*The Unified Modeling Language. Reference Manual*”. Addison-Wesley Object Technology Series, 1999.

Manual completo de referencia de UML que completa la trilogía de libros sobre UML de los tres autores. En el libro se expone en primer lugar la forma de llevar a cabo el modelado y los diagramas utilizados en cada una de las vistas del sistema. Posteriormente aparece en orden alfabético la descripción de todos los componentes del lenguaje. El libro puede ser de gran ayuda en el estudio de los temas 4 y 5, especialmente en éste último.

- 📖 **Yourdon, E.** “*Análisis Estructurado Moderno*”. Prentice-Hall Hispanoamericana, 1993.

La versión en castellano de este libro corresponde a la traducción del texto del mismo autor “*Modern Structured Analysis*”, Prentice-Hall, 1989. En él se describe el método propuesto por Yourdon que se basa en las ideas de DeMarco en el año 1979. Recomendado como referencia importante en la asignatura de Ingeniería del Software de primer ciclo, puede ser de ayuda en esta asignatura para el estudio del tema 3.

A.3.6 Administración de Proyectos Informáticos

El propósito fundamental de esta asignatura es proporcionar los conocimientos y capacidades prácticas necesarias para llevar a cabo las actividades de Ingeniería del Software relacionadas con los aspectos de gestión y control de proyectos. Dichas tareas comienzan antes del inicio de cualquier actividad técnica y continúan a lo largo de todo el ciclo de vida del software. El

objetivo final de dichas actividades es obtener un producto software final de calidad y fiable desarrollado mediante un proceso eficiente y productivo.

Los objetivos de la unidad docente que se pretenden conseguir en esta asignatura son los siguientes:

T10, T11, T12, P3, P8, H1, H2, H3 y H4.

Los apartados que se exponen a continuación siguen la pauta marcada en las asignaturas precedentes.

A.3.6.1 Ficha de la asignatura

Asignatura	<i>Administración de Proyectos Informáticos (troncal)</i>
Créditos	<i>6T + 3P</i>
Estudios	<i>Ingeniería en Informática (2º ciclo)</i>
Plan	<i>B.O.E de 1-7-1999</i>
Curso	<i>2º</i>
Cuatrimestre	<i>1º y 2º (anual)</i>
Responsable	<i>María N. Moreno García (mmg@usal.es)</i>
Página web de la asignatura	http://lisisu02.usal.es/~mmoreno/api.html

Tabla A.18. Administración de Proyectos Informáticos

A.3.6.2 Prerrequisitos

Para lograr los objetivos anteriores es fundamental que el alumno haya adquirido previamente conocimientos sobre el proceso de desarrollo, ya que dichas nociones son básicas para la comprensión y asimilación de los conceptos y métodos que se tratan en esta asignatura. Los conocimientos requeridos se adquieren en las asignaturas **Ingeniería del Software** que se imparte en 3º de ITIS y **Análisis de Sistemas** perteneciente a primer curso del segundo ciclo.

A.3.6.3 Temario teórico/práctico

Presentación de la asignatura (1 hora)

Tema 1. Visión general de la administración de proyectos (6 horas)

Tema 2. Medición del software (11 horas)

Tema 3. Métodos de estimación y gestión del riesgo (10 horas)

Tema 4. Planificación temporal de proyectos (12 horas)

Tema 5. Gestión de la calidad (15 horas)

Tema 6. Gestión de configuraciones (5 horas)

Tabla A.19. Programa teórico de la asignatura Administración de Proyectos Informáticos (6 créditos)

Laboratorio: uso de herramientas automatizadas (30 horas)

Práctica 1. Aplicación de métricas (4 horas)

Práctica 2. Estimación de coste y esfuerzo de un proyecto (8 horas)

Práctica 3. Planificación temporal del proyecto (18 horas)

Tabla A.20. Programa práctico de la asignatura Administración de Proyectos Informáticos (3 créditos)

Actividades Docentes Complementarias

- Seminarios impartidos sobre temas específicos
- Trabajos voluntarios realizados por los alumnos
- Conferencias invitadas

En la Tabla A.21 se presenta la correspondencia existente entre el temario y los objetivos perseguidos en esta asignatura.

Elemento Docente	Objetivos
Tema 1	T10, T11, T12, H3
Tema 2	T11, H3
Tema 3	T10, H3
Tema 4	T10, H3
Tema 5	T10, T11, H3
Tema 6	T12, H3
Práctica 1	P8, H1, H2, H4
Práctica 1	P3, H1, H2, H4
Práctica 2	P3, H1, H2, H4

Tabla A.21. Correspondencia entre el temario teórico/práctico y los objetivos de la asignatura

A.3.6.4 Líneas de acción

Como en las asignaturas anteriores, seguidamente se comentarán todos los componentes de las líneas de acción que contribuirán a la consecución de los objetivos.

T10 Gestión de proyectos software: definición de objetivos, gestión de recursos, estimación de esfuerzo y coste, planificación y gestión de riesgos.

T11 Uso de métricas software para el apoyo a la gestión de proyectos software y aseguramiento de la calidad del software.

Línea de acción	Cuándo	Quién	Medios	Evaluación
Ofrecer una visión general de las actividades de gestión, planificación y seguimiento de proyectos	En el tema 1	Responsable de la asignatura	Transparencias Bibliografía básica	Examen
Mostrar el alcance de las métricas del software y las dos vertientes de su aplicación: evaluación y predicción	En el tema 2 y en las prácticas de la asignatura	Responsable de la asignatura Alumnos	Transparencias Bibliografía básica	Defensa de la práctica Examen
Estudiar y aplicar los conceptos y métodos relacionados con las actividades descritas en el primer tema.	Todos los temas de la asignatura Prácticas de la asignatura	Responsable de la asignatura Alumnos	Transparencias Bibliografía básica	Defensa de las prácticas Examen

T12 Conceptos, métodos, procesos y técnicas destinadas al mantenimiento y evolución de los sistemas software.

Línea de acción	Cuándo	Quién	Medios	Evaluación
Examinar los conceptos y tareas asociados a la gestión de configuraciones, acentando la importancia de asegurar la calidad y la consistencia de los cambios.	Tema 6	Responsable de la asignatura	Transparencias Bibliografía básica	Examen

P3 Aplicar de forma práctica los conceptos teóricos sobre gestión de proyectos.

Línea de acción	Cuándo	Quién	Medios	Evaluación
Utilización de herramientas automatizadas de gestión para la planificación y la estimación de costes.	En las clases prácticas	Responsable de la asignatura Alumnos	Medios audiovisuales Herramientas de gestión. Documentación de trabajos realizados el curso anterior	Defensa de las prácticas

P8 Recolección de diferentes métricas en el desarrollo de sistemas software reales.

Línea de acción	Cuándo	Quién	Medios	Evaluación
Utilización de herramientas automatizadas para la aplicación de métricas.	En las clases prácticas	Responsable de la asignatura Alumnos	Herramientas de gestión. Documentación de trabajos realizados el curso anterior	Defensa de las prácticas

H1 Mejora de la expresión oral.

Línea de acción	Cuándo	Quién	Medios	Evaluación
Fomentar la participación en las clases de problemas	A lo largo del curso	Responsable de la asignatura Alumnos	Pizarra	Participación de los alumnos
Defensa oral de las prácticas	Defensa de la práctica	Alumnos	Medios audiovisuales	Defensa de la práctica
Promover seminarios realizados por los alumnos como resultado de trabajos voluntarios	A lo largo de la asignatura	Grupos de alumnos	Bibliografía Recursos de Internet	Nota <i>extra</i> para estos trabajos voluntarios

H2 Mejora en la redacción de documentos técnicos.

Línea de acción	Cuándo	Quién	Medios	Evaluación
Propuesta de ejercicios de aplicación de los conceptos teóricos	Durante las clases de teoría	Responsable de la asignatura alumnos	Transparencias Pizarra	Evaluación de los ejercicios entregados
Documentación de las prácticas	Defensa de las prácticas	Alumnos Responsable de la asignatura	Herramientas de gestión Bibliografía	Defensa de la práctica
Memoria de los trabajos voluntarios	A lo largo de la asignatura	Grupos de alumnos	Bibliografía Recursos de Internet	Nota <i>extra</i> para estos trabajos voluntarios

H3 Potenciación de la capacidad del alumnos para la búsqueda de información (manejo de fuentes bibliográficas, Internet, foros de discusión...).

Línea de acción	Cuándo	Quién	Medios	Evaluación
Completar las transparencias utilizadas en clase con la bibliografía recomendada	A lo largo de la asignatura	Alumnos	Biblioteca	Examen
Promover seminarios realizados por los alumnos como resultado de trabajos voluntarios	A lo largo de la asignatura	Grupos de alumnos	Bibliografía Recursos de Internet	Nota <i>extra</i> para estos trabajos voluntarios

H4 Capacitar a los alumnos para el trabajo en grupo.

Línea de acción	Cuándo	Quién	Medios	Evaluación
Realización de las prácticas en grupos de 2 personas	Durante las clases prácticas	Alumnos Responsable de la asignatura	Documentación de trabajos realizados el curso anterior	Defensa de la práctica

A.3.6.5 Criterios de evaluación de la asignatura

a) Parte de Teoría

- Un examen final en el mes de junio.
- Un examen final en el mes de septiembre.

b) Parte Práctica

- En junio la evaluación de esta parte de la asignatura se realizará mediante la valoración de los trabajos presentados. La nota será la media de las obtenidas en cada uno de los trabajos prácticos. La defensa de dichos trabajos se realizará de forma individualizada.

- En septiembre se realizará un examen en el que el alumno tendrá que resolver un supuesto práctico con la ayuda de una herramienta.

c) Trabajos voluntarios presentados (de 0,5 a 1,5 puntos sumados a la nota conseguida en los apartados anteriores, siempre que en los apartados anteriores se obtenga la calificación mínima exigida).

d) En el caso de haber superado la parte teórica y la práctica, se aplicará la siguiente fórmula para la obtención de la calificación final:

$$\text{Nota Final} = ((\text{Teoría} * 0,6) + (\text{Práctica} * 0,3)) * 10/9 + \text{Nota trabajos}$$

A.3.6.6 Bibliografía básica de referencia

Siguiendo la misma línea que en el resto de las asignaturas de la unidad docente, la bibliografía que aquí se incluye servirá de complemento al material suministrado para el seguimiento de las clases. Los criterios seguidos para su elección son los mismos que en los casos anteriores.

 **Burnett, K.** “*The Project Management Paradigm*”. Springer, 1998.

Este texto proporciona un enfoque de la gestión de proyectos basado en la combinación de aspectos humanos y principios metodológicos para conseguir el éxito del proyecto. Es interesante la visión que ofrece sobre los conceptos, principios y métodos de estimación y gestión de riesgos. El capítulo que dedica a la gestión de la calidad es bastante completo, por lo que puede ser muy útil en el estudio del tema 5.

 **Cos, M.** “*Teoría General del Proyecto*”. Síntesis, 1997.

El primer volumen de este texto cubre los conceptos y métodos de organización de proyectos, estimación, planificación y control. El último capítulo está dedicado al plan de calidad del proyecto.

-  **Fenton, N. E., Pfleeger, S. L.** “*Software Metrics. A Rigorous & Practical Approach*”. PWS, 1997.

Excelente obra sobre métricas del software que cubre diversos aspectos de su aplicación como son la evaluación y predicción del tamaño y funcionalidad del software, de la productividad, de costes y esfuerzos y de la calidad del producto y del proceso de desarrollo. En el se recogen gran número de métricas existentes en bibliografía así como algunas propuestas por los mismos autores del libro. Los primeros capítulos dedicados a los fundamentos de la medición y a la forma de realizar medidas eficaces son muy recomendables. Imprescindible en el estudio de los temas 2, 3 y 5.

-  **McConnell, S.** “*Desarrollo y Gestión de Proyectos Informáticos*”. McGraw Hill, 1997.

Traducción de la primera edición en inglés de “*Rapid Development*”. Este libro, dirigido principalmente a las estrategias de desarrollo rápido, dedica el capítulo 4 a explicar de manera clara las bases del desarrollo de software. Tiene también temas dedicados a la estimación y gestión de riesgos. Los capítulos 11, 12 y 13 constituyen una buena referencia sobre los aspectos a considerar en la formación de equipos de trabajo (motivación, características, estructura...), al igual que el capítulo 15 en el apartado de herramientas de mejora de la productividad. La tercera parte del libro, dedicada a “métodos recomendables” cubre la práctica totalidad de los temas de la asignatura.

-  **Pressman, R. S.** “*Ingeniería del Software: Un Enfoque Práctico*”. 4ª Edición. McGrawHill. 1998.

Este clásico de la Ingeniería del Software constituye una referencia obligada en casi todos los temas de la asignatura. El capítulo 3 trata de forma global los aspectos de gestión del proyecto. La medición de software aparece en apartados sueltos incluidos en diferentes capítulos y en un tema relacionado con el proceso software que está dedicado casi completamente a la medición. El resto de los temas de la asignatura los contempla también de forma detallada en diferentes capítulos.

-  **Puig, J.** “*Proyectos Informáticos. Planificación, Desarrollo y Control*”, Paraninfo, 1994.

El texto de Puig puede ser de utilidad para obtener una visión global de las actividades clásicas de gestión de proyectos que se tratan de una manera concisa y clara. El libro utiliza, en algunos casos, notaciones y terminología obsoleta y no incluye técnicas actuales, por lo que necesitaría una revisión.

 **Quang, P., Goñi, J.** “*Dirección de Proyectos Informáticos*”. Eyrolles, 1994.

En este libro se contemplan los métodos clásicos de organización de equipos, estimación, planificación, seguimiento y control de calidad de un proyecto. Además se ofrecen una serie de recomendaciones obtenidas de experiencias prácticas. El libro proporciona una visión superficial de los temas de esta asignatura, por lo que se aconseja su uso únicamente en el tema de introducción.

 **Romero, C.** “*Técnicas de Programación y Control de Proyectos*”. Pirámide, 1997.

Describe de forma detallada tres métodos clásicos de planificación temporal (PERT, CPM y ROY) partiendo de la explicación previa de los conceptos sobre teoría de grafos que son la base de tales métodos. Puede ser de gran ayuda en el estudio del tema 4.

 **Sommerville, I.** “*Software Engineering*”. 5th Edition. Addison-Wesley, 1996.

La quinta edición de esta obra sobre ingeniería del software incorpora nuevos capítulos sobre temas de gestión (personal, calidad...) y mejora los dedicados a la tecnología CASE y evolución del software. Estos cambios lo convierten un libro de consulta indispensable de la asignatura.

A.3.7 Sistemas de Información

El propósito fundamental de esta asignatura es la aplicación práctica de los principios de la Programación Orientada a Objetos estudiados en otras asignaturas. Los objetivos de la unidad docente que se pretenden conseguir en esta asignatura son los siguientes:

P7, H1, H2, H3 y H4.

Los apartados que se exponen a continuación siguen la pauta marcada en las asignaturas precedentes.

A.3.7.1 Ficha de la asignatura

Asignatura	<i>Sistemas de Información (troncal)</i>
Créditos	0T + 9P
Estudios	<i>Ingeniería en Informática (2º ciclo)</i>
Plan	<i>B.O.E de 1-7-1999</i>
Curso	2º
Cuatrimestre	1º
Responsable	José Rafael García-Bermejo Giner (coti@usal.es)
Página web de la asignatura	http://acebuche.usal.es

Tabla A.22. Sistemas de Información

A.3.7.2 Prerrequisitos

Para lograr los objetivos anteriores es fundamental que el alumno haya adquirido previamente conocimientos sobre la realización de proyectos software, tanto en la asignatura de **Ingeniería del Software** del tercer curso del primer ciclo, como en la asignatura de **Análisis de Sistemas** del primer curso del segundo ciclo.

Además, sería aconsejable que el alumno tuviera conocimientos prácticos de **Programación Orientada a Objetos**, del tercer curso del primer ciclo.

A.3.7.3 Temario práctico

Presentación de la asignatura (1 hora)

Tema 1. Conceptos básicos de orientación a objetos (9 Horas)

Tema 2. El lenguaje Java. Java frente a C/C++ (20 Horas)

Tema 3. Applets (15 Horas)

Tema 4. Creación de aplicaciones en Java (15 Horas)

Tema 5. Realización de un supuesto práctico (30 Horas)

Tabla A.23. Programa de prácticas (9 Créditos)

Actividades Docentes Complementarias

- Seminarios impartidos sobre temas específicos.
- Trabajos voluntarios realizados por los alumnos.
- Conferencias invitadas.

A.3.7.4 Líneas de acción

P7 Aplicar de forma práctica los conceptos teóricos sobre gestión de proyectos.

Línea de acción	Cuándo	Quién	Medios	Evaluación
Búsqueda e instalación del entorno de desarrollo más adecuado y reciente para su uso en la asignatura	En la fase inicial del curso	Responsable de la asignatura Alumnos	Internet Revistas Bibliografía	
Cambio de mentalidad para la adecuada aplicación del paradigma objetual	A partir del tema 2	Responsable de la asignatura	Numerosos ejercicios prácticos	Práctica en dos ocasiones a lo largo del curso y examen teórico

H1 Mejora de la expresión oral.

Línea de acción	Cuándo	Quién	Medios	Evaluación
Defensa en seminarios de algún tema teórico o ejercicio práctico	A lo largo del curso	Responsable de la asignatura Alumnos	Audiovisuales	Participación de los alumnos

H2 Mejora en la redacción de documentos técnicos.

Línea de acción	Cuándo	Quién	Medios	Evaluación
Extracción y documentación de las especificaciones	En el desarrollo de los trabajos prácticos	Alumnos	Enunciados de los problemas	Evaluación de los ejercicios entregados

H3 Potenciación de la capacidad del alumnos para la búsqueda de información (manejo de fuentes bibliográficas, Internet, foros de discusión...).

Línea de acción	Cuándo	Quién	Medios	Evaluación
Búsqueda de información actualizada en Internet	A lo largo de la asignatura	Alumnos	Internet Revistas	Resultados de los ejercicios prácticos

H4 Capacitar a los alumnos para el trabajo en grupo.

Línea de acción	Cuándo	Quién	Medios	Evaluación
Construcción de aplicaciones modulares, realizando los distintos grupos los módulos componentes	Durante las clases prácticas	Alumnos Responsable de la asignatura	Enunciado Entorno de desarrollo	Por la interfaz y la implementación de los módulos

A.3.7.5 Criterios de evaluación de la asignatura

a) Parte Práctica

- En diciembre y en febrero se examinan los ejercicios realizados.
- En febrero se realiza un examen escrito.
- En septiembre se realizará un examen escrito.

c) Se aplica la siguiente fórmula para la obtención de la calificación final:

$$\text{Nota Final} = (\text{Ejercicios entregados} * 0,5) + (\text{Examen} * 0,5)$$

A.3.7.6 Bibliografía básica de referencia

Siguiendo la misma línea que en el resto de las asignaturas de la unidad docente, la bibliografía que aquí se incluye servirá de complemento al material suministrado para el seguimiento de las clases.

 **Jaworski, J.** “Java 1.2 Unleashed”. Sams, 1998.

- 📖 **Cadenhead, R.** “*Teach Yourself JAVA 1.2 in 24 hours*”. Sams, 1999.
- 📖 **Lemay, L., Cadenhead, R.** “*Teach Yourself Java 2 Platform in 21 Days: Professional Reference Edition*”. Sams Teach Yourself in 21 Days Series. Sams, 1999.
- 📖 **SUN Microsystems.** “*Java™ Standard Edition Platform Documentation*”. <http://java.sun.com/docs/index.html>. [Última vez visitado, 16-3-2000]. 2000.
- 📖 **SUN Microsystems.** “*The Java Tutorial. A Practical Guide for Programmers*”. <http://java.sun.com/docs/books/tutorial/index.html>. [Última vez visitado, 12-4-2002]. March 2002.

A.3.8 Proyecto I.I.

El proyecto de la Ingeniería Informática se convierte de nuevo en otra excelente ocasión para la consecución del objetivo **P9**, pero esta vez con la experiencia de haber realizado previamente otro proyecto y haber cursado dos años más estudios universitarios de segundo ciclo.

Los objetivos de este proyecto no deben ser un calco de los del proyecto de I.T.I.S, sino que debe convertirse en un trabajo donde el alumno demuestre su madurez, además de sus conocimientos y dominio de la técnica. Así, el proyecto debe ir acompañado de las métricas oportunas, planes de implantación y mantenimiento, justificaciones teóricas basadas en el estado del arte del tema abordado...

En este sentido la guía recogida en [García et al., 2000] sigue siendo una referencia de consulta válida, pero ha de tenerse en cuenta que dicha guía fue desarrollada para los proyectos de I.T.I.S, no para un proyecto de un segundo ciclo.

A.4 Referencias

- [Buxton et al., 1976] **Buxton, J. M., Naur, P., Randell, B. (Editors).** “*Software Engineering Concepts and Techniques*”. In Proceedings of 1968 NATO Conference on Software Engineering, Van Nostrand Reinhold, 1976.
- [García et al., 1999a] **García Peñalvo, F. J., Moreno García, M^a N., González Talaván, G., Moreno Montero, Á. M^a.** “*Plan de Calidad para Asignaturas en Ingenierías Técnicas en Informática*”. Actas del Congreso Nacional de Informática Educativa CONIED’99. Editores M. Ortega y J. Bravo. (Puertollano (Ciudad Real), 17-19 de Noviembre de 1999). Resumen en página 46 y ponencia en versión digital (CD-ROM). 1999.

- [García et al., 1999b] **García Peñalvo, F. J., Moreno García, M^a N., Montero García, E., Arranz Val, P.** “*Evaluación del Profesorado: Un Protocolo de Evaluación por Pares*”. Actas del Congreso Nacional de Informática Educativa CONIED’99. Editores M. Ortega y J. Bravo. (Puertollano (Ciudad Real), 17-19 de Noviembre de 1999). Resumen en página 47 y ponencia en versión digital (CD-ROM). 1999.
- [García et al., 2000] **García Peñalvo, F. J., Maudes Raedo, J. M., Piattini Velthuis, M. G., García-Bermejo Giner, J. R., Moreno García, M^a N.** “*Proyecto de Final de Carrera en la Ingeniería Técnica en Informática: Guía de Realización y Documentación*”. Departamento de Informática y Automática de la Universidad de Salamanca. Versión 1.5.2. <http://tejo.usal.es/~fgarcia/doc/pfc.pdf>. Marzo, 2000.
- [Leveson, 1997] **Leveson, N G.** “*Software Engineering: Stretching the Limits of Complexity*”. Communications of the ACM 40(2): 129-131. February 1997.
- [Parnas, 1997] **Parnas, D. L.** “*Software Engineering: An Unconsummated Marriage*”. Communications of the ACM 40(9): 128. September 1997.
- [Shaw y Tomayko, 1991] **Shaw, M., Tomayko, J. E.** “*Models for Undergraduate Project Courses in Software Engineering*”. Technical Report CMU/SEI-91-TR-10 (ESD-91-TR-10). Software Engineering Institute – Carnegie Mellon University. Pittsburgh, Pennsylvania 15213 (USA). August 1991.