

**Realización de proyectos docentes para
asignaturas de Ingeniería Informática bajo
las Directrices del Espacio Europeo de
Educación Superior (US14/04)**

Memoria de Resultados

Coordinador: Dr. Francisco José García Peñalvo
Departamento de Informática y Automática.
Facultad de Ciencias de la Universidad de Salamanca.
Plaza de los Caídos S/N, 37008

Octubre de 2006

Tabla de contenidos

1. Introducción	1
2. Objetivos y resultados del proyecto	3
3. Materiales elaborados	4
4. Conclusiones	6
5. Referencias	7
Apéndice I: Guías ECTS	9
Apéndice II: Artículos publicados	284

1. Introducción

La declaración conjunta de los ministros europeos de educación de junio de 1999 en Bolonia [1] establecía las bases para la definición de Espacio Europeo de Educación Superior (EEES) con anterioridad a 2010. El camino hacia este EEES supone un reto de enormes dimensiones para la Universidad Europea en general y para la Universidad Española en particular.

En lo tocante a las titulaciones de Ingeniería Informática en España han sido bastante las iniciativas, grupos de trabajo y propuestas que se han iniciado para encarar este futuro que se avecina. Sin lugar a dudas, la iniciativa de mayor entidad en el terreno de la Informática ha venido de la mano del proyecto EICE (Los Estudios de Informática y Convergencia Europea) que, financiado por la ANECA (Agencia Nacional de Evaluación de la Calidad y Acreditación), ha tenido como objetivos la elaboración de un Estudio para el diseño de los planes de estudio y títulos oficiales de Grado adaptados al Espacio Europeo de Educación Superior de las titulaciones de informática, y la elaboración del Libro Blanco de dichas titulaciones [2].

Al amparo de la Orden EDU/1025/2004 (B.O.C.y.L N° 124 de 30 de junio de 2004) y del Anuncio de la Agencia de la Calidad del Sistema Universitario de Castilla y León (B.O.C.y.L N° 141 de 23 de julio de 2004), se solicitó el proyecto “Plan de Estudios de Ingeniería Informática en el Contexto de las Universidad Públicas de Castilla y León: Una Experiencia Piloto bajo las Directrices del Espacio Europeo de Educación Superior”, que fue concedido con referencia US14/04 y financiado conjuntamente por la Consejería de Educación de la Junta de Castilla y León (B.O.C y L. n° 196, de fecha 8 de octubre de 2004, se publica la Orden EDU/ 1539/2004, de 6 de octubre) y por la Agencia para la Calidad del Sistema Universitario de Castilla y León (B.O.C y L. n° 196, de fecha 8 de octubre 2004, se publica la RESOLUCIÓN de 1 de octubre de 2004). Este proyecto ha tenido su continuación en el curso académico 2005-2006 al amparo del Acuerdo de 4 de mayo de 2005, de la Agencia para la Calidad del Sistema Universitario de Castilla y León (B.O.C.y.L N° 93 de 17 de mayo de 2005), mediante el proyecto “Realización de proyectos docentes para asignaturas de Ingeniería Informática bajo las Directrices del Espacio Europeo de Educación Superior”, que fue concedido con referencia US14/04 y financiado por la Agencia para la Calidad del Sistema Universitario de Castilla y León (B.O.C y L. n° 175, de fecha 9 de septiembre de 2005, se publica la RESOLUCIÓN de 2 de septiembre de 2005).

El objetivo general de este proyecto ha consistido en trabajar con las piezas clave de un Plan de Estudios, esto es, las asignaturas. Para lo que se va a comenzar estudiando asignaturas que actualmente se imparten en los Planes de Estudio de Ingeniería Informática, para redefinirlas respecto a las directrices del EEES en base a objetivos y competencias, destacando los puntos clave y descartando los tópicos superfluos, relacionando estas asignaturas con aquéllas con las que guardan estrechos nexos y con el resto de las asignaturas y, por último, comparando los resultados de las asignaturas afines en diferentes Centros. Las asignaturas objeto de estudio han sido, preferentemente, aquéllas que tenían que ver con los Contenidos Formativos Comunes (CFC) definidos en el Libro Blanco de la Ingeniería Informática. Con todo este trabajo se cuenta con elementos de incalculable valor para afrontar la construcción de los futuros los Planes de Estudio del Grado en Ingeniería Informática, según el marco de referencia que acabe definiendo el Ministerio, con enfoque ascendente al contar con las piezas clave definidas en función de los objetivos y competencias que se le quieran transmitir a los futuros graduados y postgraduados en Ingeniería Informática.

A mayores, se ha mantenido el objetivo de integrar los resultados obtenidos con los resultados de otros proyectos y grupos de trabajo, para reforzar la red interuniversitaria permanente para la coordinación de los estudios de Ingeniería Informática, tanto a nivel de Grado como de Postgrado, en Castilla y León.

2. Objetivos y resultados del proyecto

Al definir este proyecto se marcaron unos objetivos. A la hora de hacer balance del proyecto se recuerdan dichos objetivos, pero a la vez se comparan con los resultados obtenidos, para ver cuáles han sido satisfactoriamente conseguidos, y cuáles, como se comentaba en la introducción han quedado inconclusos.

Objetivos	Resultados
Consolidar y ampliar el grupo de trabajo formado por profesores de diferentes universidades para el desarrollo del EEES en relación con la Ingeniería Informática.	Este objetivo se ha cumplido con creces, pues la colaboración entre los integrantes del grupo ha sido alta, teniendo en cuenta que se contaba con representantes de las diferentes universidades de Castilla y León
Redefinir, de forma individualizada, diferentes asignaturas de los planes actuales de la Ingeniería Informática, con vistas a adecuar el temario y los contenidos de las mismas a las premisas y directrices del proceso de convergencia al EEES, con una posible adecuación metodológica basada en casos, adecuación de los contenidos a los objetivos y competencias, eliminando todos los contenidos superfluos y una aproximación ETCS. El conjunto objetivo de asignaturas debe estar formado, principalmente, por los núcleos temáticos que conforman los Contenidos Formativos Comunes definidos en el Libro Blanco para la Ingeniería Informática	Se han desarrollado 10 guías ECTS de otras tantas asignaturas ofertadas en las diversas titulaciones de Ingeniería Informática que se imparten en las universidades públicas de Castilla y León
Con aquellas asignaturas que mantienen una estrecha relación entre sí, hacer un estudio de sus objetivos y competencias de forma conjunta para así revisar los objetivos y competencias de las asignaturas individuales, y eliminar redundancias	Esto se ha llevado a cabo con la disciplina de la Ingeniería del Software. Se ha realimentado la definición de las guías
Proceder de una forma similar al objetivo anterior, pero con asignaturas que, no estando tan estrechamente relacionadas, existe una clara relación de dependencia.	Esto se ha llevado a cabo con el resto de asignaturas, en especial aprovechando la relación existente entre la Ingeniería del Software y las Bases de Datos. Se ha realimentado la definición de las guías
Seguir al tanto de los acontecimientos que rodeen la definición del EEES en general y en lo relacionado con la Ingeniería Informática en particular	Miembros del equipo de trabajo participan activamente en la Conferencia de Decanos y Directos de Informática. También se han organizado eventos que han supuesto la puesta en común de los avances hacia la convergencia europea en otras universidades

Tabla 1. Objetivos/Resultados del proyecto

3. Materiales elaborados

Los materiales elaborados en este proyecto han sido los siguientes:

- **Guías ECTS (APÉNDICE I).**- Se han desarrollado 10 guías ECTS, a saber:
 - Guía Docente de Diseño de Bases de Datos. Facultad de Ciencias. Universidad de Salamanca
 - Guía Docente de Ingeniería del Software. Facultad de Ciencias. Universidad de Salamanca
 - Guía Docente de Procesadores de Lenguajes. Facultad de Ciencias. Universidad de Salamanca
 - Guía docente de Hipermedia: Diseño y Evaluación. Escuela Politécnica Superior de Zamora. Universidad de Salamanca
 - Guía Docente de las asignaturas de Ingeniería del Software en la Ingeniería Técnica de Gestión. Escuela Politécnica Superior de Zamora. Universidad de Salamanca
 - Guía Docente de Redes de Ordenadores. Escuela Politécnica Superior de Zamora. Universidad de Salamanca
 - Guía Docente de Sistemas Informáticos. Escuela Politécnica Superior de Zamora. Universidad de Salamanca
 - Guía Docente de Sistemas Operativos. Escuela Politécnica Superior de Zamora. Universidad de Salamanca
 - Guía Docente de Ingeniería del Software II. Escuela Politécnica Superior de Zamora. Universidad de Salamanca
 - Guía Docente de Ingeniería del Software. Escuela Técnica Superior de Ingeniería Informática. Universidad de Valladolid
- **Artículos publicados (APÉNDICE II).**- Se ha presentado el trabajo realizado en varios foros.
 - **García Peñalvo, F. J.** “*Diseño Curricular de una Asignatura en el Contexto del Espacio Europeo de Educación Superior*”. En las *Actas de las I Jornadas de Innovación Educativa de la Escuela Politécnica Superior de Zamora. Las Enseñanzas Técnicas ante el Reto del Espacio Europeo de Educación Superior* (Zamora, 20-22 de junio de 2006). J. L. Pérez Iglesias, M^a L. Pérez Delgado, M. P. Rubio Cavero, J. C. Matos Franco, J. Calvo Gallego (Eds.). Versión CD-ROM. ISBN 84-689-9304-2. Junio, 2006.
 - **Calvo Gallego, J., García Peñalvo, F. J.** “*Guía Docente de la Asignatura Ingeniería del Software II dentro del Marco del EEES*”. En las *Actas de las I Jornadas de Innovación Educativa de la Escuela Politécnica Superior de Zamora. Las Enseñanzas Técnicas ante el Reto del Espacio Europeo de Educación Superior* (Zamora, 20-22 de junio de 2006). J. L. Pérez Iglesias, M^a L. Pérez Delgado, M. P. Rubio Cavero, J. C. Matos Franco, J. Calvo Gallego (Eds.). Versión CD-ROM. ISBN 84-689-9304-2. Junio, 2006.

- **Pérez, J. L., Matos, J. C., García, F. J.** “*La Asignatura de Hipermedia en el Marco del E.E.E.S.*”. En las *Actas de las I Jornadas de Innovación Educativa de la Escuela Politécnica Superior de Zamora. Las Enseñanzas Técnicas ante el Reto del Espacio Europeo de Educación Superior* (Zamora, 20-22 de junio de 2006). J. L. Pérez Iglesias, M^a L. Pérez Delgado, M. P. Rubio Cavero, J. C. Matos Franco, J. Calvo Gallego (Eds.). Versión CD-ROM. ISBN 84-689-9304-2. Junio, 2006.
- **González, A. B., Rodríguez, M. J., Olmos S.** “*Aprendizaje activo en Ingeniería Técnica Informática, esp. Gestión. Sistemas Informáticos*”. En las *Actas de las I Jornadas de Innovación Educativa de la Escuela Politécnica Superior de Zamora. Las Enseñanzas Técnicas ante el Reto del Espacio Europeo de Educación Superior* (Zamora, 20-22 de junio de 2006). J. L. Pérez Iglesias, M^a L. Pérez Delgado, M. P. Rubio Cavero, J. C. Matos Franco, J. Calvo Gallego (Eds.). Versión CD-ROM. ISBN 84-689-9304-2. Junio, 2006.
- **Martín, D., San Juan, Y., Vizán, R., González, A. B.** “*Aprender Aprobar*”. En las *Actas de las I Jornadas de Innovación Educativa de la Escuela Politécnica Superior de Zamora. Las Enseñanzas Técnicas ante el Reto del Espacio Europeo de Educación Superior* (Zamora, 20-22 de junio de 2006). J. L. Pérez Iglesias, M^a L. Pérez Delgado, M. P. Rubio Cavero, J. C. Matos Franco, J. Calvo Gallego (Eds.). Versión CD-ROM. ISBN 84-689-9304-2. Junio, 2006.
- **García Peñalvo, F. J. (Ed.)** “*Los Estudios de Ingeniería Informática en el Espacio Europeo de Educación Superior. Contexto y Realidad en la Comunidad Autónoma de Castilla y León*”. Ediciones Universidad de Salamanca. Colección Aquilafuente, N^o 101. ISBN 84-7800-436-X. 2006.
- **García Peñalvo, F. J., Alonso Romero, L., Cardeñoso Payo, V., Pardo Aguilar, C., Fernández Díaz, R. Á.** “*Perspectiva Histórica de los Estudios de Ingenierías Informáticas en las Universidades Públicas de Castilla y León*”. En F. J. García Peñalvo (Ed.) *Los Estudios de Ingeniería Informática en el Espacio Europeo de Educación Superior. Contexto y Realidad en la Comunidad Autónoma de Castilla y León*. Páginas 1-7. Ediciones Universidad de Salamanca. Colección Aquilafuente, N^o 101. ISBN 84-7800-436-X. Agosto, 2006.
- **García Peñalvo, F. J.** “*Pensando en ECTS. Un Caso Práctico para la Asignatura de Ingeniería del Software*”. En F. J. García Peñalvo (Ed.) *Los Estudios de Ingeniería Informática en el Espacio Europeo de Educación Superior. Contexto y Realidad en la Comunidad Autónoma de Castilla y León*. Páginas 139-156. Ediciones Universidad de Salamanca. Colección Aquilafuente, N^o 101. ISBN 84-7800-436-X. Agosto, 2006.
- **Edición del libro** “*Los Estudios de Ingeniería Informática en el Espacio Europeo de Educación Superior*”.- Libro que recoge los mejores trabajos extendidos de los presentados en las I y II Jornadas sobre la Ingeniería Informática en el Espacio Europeo de Educación Superior (II-EEES’05 y II-EEES’06), celebradas en Salamanca los meses de octubre de 2005 y 2006 respectivamente.

4. Conclusiones

Las conclusiones de este trabajo presentan una parte muy positiva relacionada con el grupo de trabajo, la colaboración existente, la implicación en los debates de la comunidad universitaria, pero también de otras comunidades como la comunidad profesional mediante el Ilustre Colegio de Ingenieros Informáticos de Castilla y León, diversos colectivos de profesores de informática de Portugal y, por supuesto, la CODDI. Esta colaboración se hace extensible a otros grupos de la Comunidad Autónoma que también contaban con sus proyectos de innovación docente en el plano del EEES, creándose una auténtica red de discusión, debate y trabajo.

De nuevo los aspectos más negativos se deben al marca de incertidumbre que ha rodeado, una vez más, el desarrollo del mismo, sintiendo como muchas veces las noticias relacionadas con la evolución del proceso de convergencia en el ámbito político afectaban negativamente al avance en el contexto académico.

A pesar de todo, se han desarrollado 10 guías ECTS de otras tantas asignaturas, lo cual además de ofrecer como resultado un material de referencia de gran valor, a obligado a un buen número de profesores a imbuirse de dos prácticas esenciales en el EEES, por un lado el trabajo con los ECTS, planificando las asignaturas desde un punto de vista de esfuerzo, y, por otro lado, la coordinación necesaria para revisar el trabajo de los demás y sacar una realimentación positiva que aplicar en sus propias guías ECTS.

5. Referencias

1. **European Ministers of Education.** *The European Higher Education Area - Bologna Declaration*, Bologna on the 19th of June 1999.
2. **Grupo EICE.** *Libro Blanco sobre las Titulaciones Universitarias de Informática en el Nuevo Espacio Europeo de Educación Superior*. ANECA – Los Estudios de Informática y la Convergencia Europea. Ingeniería Informática. Versión Final. Marzo 2005.
3. **García Peñalvo, F. J.** “*Diseño Curricular de una Asignatura en el Contexto del Espacio Europeo de Educación Superior*”. En las *Actas de las I Jornadas de Innovación Educativa de la Escuela Politécnica Superior de Zamora. Las Enseñanzas Técnicas ante el Reto del Espacio Europeo de Educación Superior* (Zamora, 20-22 de junio de 2006). J. L. Pérez Iglesias, M^a L. Pérez Delgado, M. P. Rubio Cavero, J. C. Matos Franco, J. Calvo Gallego (Eds.). Versión CD-ROM. ISBN 84-689-9304-2. Junio, 2006.
4. **Calvo Gallego, J., García Peñalvo, F. J.** “*Guía Docente de la Asignatura Ingeniería del Software II dentro del Marco del EEES*”. En las *Actas de las I Jornadas de Innovación Educativa de la Escuela Politécnica Superior de Zamora. Las Enseñanzas Técnicas ante el Reto del Espacio Europeo de Educación Superior* (Zamora, 20-22 de junio de 2006). J. L. Pérez Iglesias, M^a L. Pérez Delgado, M. P. Rubio Cavero, J. C. Matos Franco, J. Calvo Gallego (Eds.). Versión CD-ROM. ISBN 84-689-9304-2. Junio, 2006.
5. **Pérez, J. L., Matos, J. C., García, F. J.** “*La Asignatura de Hipermedia en el Marco del E.E.E.S.*”. En las *Actas de las I Jornadas de Innovación Educativa de la Escuela Politécnica Superior de Zamora. Las Enseñanzas Técnicas ante el Reto del Espacio Europeo de Educación Superior* (Zamora, 20-22 de junio de 2006). J. L. Pérez Iglesias, M^a L. Pérez Delgado, M. P. Rubio Cavero, J. C. Matos Franco, J. Calvo Gallego (Eds.). Versión CD-ROM. ISBN 84-689-9304-2. Junio, 2006.
6. **González, A. B., Rodríguez, M. J., Olmos S.** “*Aprendizaje activo en Ingeniería Técnica Informática, esp. Gestión. Sistemas Informáticos*”. En las *Actas de las I Jornadas de Innovación Educativa de la Escuela Politécnica Superior de Zamora. Las Enseñanzas Técnicas ante el Reto del Espacio Europeo de Educación Superior* (Zamora, 20-22 de junio de 2006). J. L. Pérez Iglesias, M^a L. Pérez Delgado, M. P. Rubio Cavero, J. C. Matos Franco, J. Calvo Gallego (Eds.). Versión CD-ROM. ISBN 84-689-9304-2. Junio, 2006.
7. **Martín, D., San Juan, Y., Vizán, R., González, A. B.** “*Aprender Aprobar*”. En las *Actas de las I Jornadas de Innovación Educativa de la Escuela Politécnica Superior de Zamora. Las Enseñanzas Técnicas ante el Reto del Espacio Europeo de Educación Superior* (Zamora, 20-22 de junio de 2006). J. L. Pérez Iglesias, M^a L. Pérez Delgado, M. P. Rubio Cavero, J. C. Matos Franco, J. Calvo Gallego (Eds.). Versión CD-ROM. ISBN 84-689-9304-2. Junio, 2006.
8. **García Peñalvo, F. J., Alonso Romero, L., Cardeñoso Payo, V., Pardo Aguilar, C., Fernández Díaz, R. Á.** “*Perspectiva Histórica de los Estudios de Ingenierías Informáticas en las Universidades Públicas de Castilla y León*”. En F. J. García Peñalvo (Ed.) *Los Estudios de Ingeniería Informática en el Espacio Europeo de Educación Superior. Contexto y Realidad en la Comunidad*

Autónoma de Castilla y León. Páginas 1-7. Ediciones Universidad de Salamanca. Colección Aquilafuente, N° 101. ISBN 84-7800-436-X. Agosto, 2006.

9. **García Peñalvo, F. J.** “*Pensando en ECTS. Un Caso Práctico para la Asignatura de Ingeniería del Software*”. En F. J. García Peñalvo (Ed.) *Los Estudios de Ingeniería Informática en el Espacio Europeo de Educación Superior. Contexto y Realidad en la Comunidad Autónoma de Castilla y León.* Páginas 139-156. Ediciones Universidad de Salamanca. Colección Aquilafuente, N° 101. ISBN 84-7800-436-X. Agosto, 2006.

Apéndice I: Guías ECTS

Guía Docente de Diseño de Bases de Datos

Versión 2 – 14-09-2006

Ana Belén Gil González

Departamento de Informática y Automática
Facultad de Ciencias – Universidad de Salamanca
Plaza de los Caídos S/N, 37008, Salamanca, España
Tfno. +34 923 294400. Ext. 1302
Fax. +34 923 294514

abg@usal.es

1. Contexto

1.1. Perfil de los créditos y su adecuación al perfil profesional y académico de la titulación

Las bases de datos constituyen un área clásica y bien establecida. La UNESCO, en su clasificación de la ciencia, la engloba en el área 1203 (Ciencia de los ordenadores), con una entrada (1203/12) para las bases de datos (bajo la terminología “bancos de datos”) y otra entrada (1203/18) para los sistemas de información, su diseño y componentes, si bien estamos hablando de una clasificación de los años 1985-86. Al ser un área bastante central de la informática, las bases de datos se interrelacionan con prácticamente todas los demás conocimientos en la titulación de un ingeniero informático, tales como estructuras de datos, los sistemas operativos, etc.

La formación de un ingeniero en informática debe contar por tanto y de manera imprescindible con una visión global de los conceptos y las tecnologías relacionadas con los sistemas de información, así como su especificación y diseño desarrollando aspectos del modelado conceptual, lógico y físico de los modelos de datos, sus técnicas de diseño y finalmente la implementación de las bases de datos.

Debido a que currículos tales como el *Computing Curricula* están muy orientados a la empresa y en general a formar informáticos que se incorporen con rapidez y con una idea muy instrumental, el área de las Bases de Datos se encuentra muy presente, particularmente en el apartado de Gestión de la Información. Este planteamiento y concretamente los tópicos que en esta asignatura se desarrollan son fundamentales para la formación de cualquier ingeniero informático, sea cual sea su perfil académico y su futuro perfil profesional.

Así la Gestión de la Información (Bases de Datos) constituye una de las 14 áreas propuestas en el CC2001. Está relacionada con la organización de la información, y con algoritmos eficientes para el acceso y la actualización de información almacenada. En ella, se incluyen la modelización de relaciones entre datos, la seguridad y la protección de la información en entornos compartidos, y las características de los dispositivos de almacenamiento externo.

Del mismo modo el IS2002 realiza una revisión de contenidos respecto de su antecesor (IS'97) en el currículo que atiende fundamentalmente a tres factores: el advenimiento de Internet, los cambios de en cuanto alfabetización informática de los estudiantes, y el movimiento de acreditación de la información. El IS 2002 recoge dentro de las capacidades tecnológicas que se espera que un titulado en sistemas de información

adquiera la *administración y el diseño de bases de datos* como una de las cuatro capacitaciones tecnológicas que ha de alcanzar.

De modo y manera que constituye un área básica y fundamental en la capacitación de cualquier titulado universitario en Sistemas de Información, y por tanto en Ingeniería Informática en cualquiera de sus niveles técnico o superior.

1.2. Ubicación y relaciones en el plan de estudios

Una cuestión previa a la definición de los objetivos de las asignaturas de Bases de Datos, debería ser la determinación de cuáles son dichas asignaturas dentro del actual plan de estudios de la Ingeniería Técnica en Informática de Sistemas de la Universidad de Salamanca y cuál es su relación con el resto de asignaturas del plan de estudios.

Las asignaturas específicas de Bases de Datos son:

1. **Diseño de Bases de Datos** que se imparte en 2º Curso en el primer cuatrimestre. Se trata de una asignatura troncal de 4.5 teóricos.
2. **Sistemas de Bases de Datos** que se imparte igualmente en 2º Curso durante el segundo cuatrimestre. Es una asignatura troncal de 4.5 créditos (1.5 Teóricos y 3 prácticos).

Realizaremos un estudio de los conocimientos previos con los que se aborda el estudio de estas asignaturas y por otro aquellos a los que estas asignaturas sirvan de base o prerequisites. Las asignaturas de primer curso Sistemas Informáticos, Álgebra, Programación, Laboratorio de Programación y Algoritmia deben ser una buena base para la comprensión de las asignaturas que nos ocupan.

- La asignatura *Sistemas informáticos* (Obligatoria de 6 créditos, 3 teóricos y 3 prácticos), dará una base sobre los componentes físicos de los diferentes sistemas informáticos como memorias, procesadores, sistemas de entrada/salida y de almacenamiento externo.
- La asignatura *Álgebra* (Troncal 6 créditos, 3 teóricos y 3 prácticos) aportará la base matemática necesaria para trabajar con modelos de datos y organizaciones desde el punto de vista abstracto.
- En la asignatura *Algoritmia* (Troncal de 7.5, 4.5 teóricos y 3 prácticos) el alumno habrá estudiado las estructuras elementales de datos y sus algoritmos, así como una primera visión de los ficheros tradicionales.
- En las asignaturas *Programación* (Troncal de 6 créditos teóricos) y *Laboratorio de Programación* (Obligatoria de 4.5 prácticos) el alumno habrá adquirido los conocimientos teóricos y prácticos de programación que le permitirán la realización de las prácticas de la asignatura *Sistemas de Bases de Datos*.

Existe también una relación con otras asignaturas que se imparten en el mismo curso como son:

- *Estructuras de Datos* (Troncal de 6 créditos, 3 teóricos y 3 prácticos) donde se estudian los conceptos generales de organización y diseño de ficheros, así como los fundamentos de las estructuras de datos en memoria. Teniendo en cuenta la importancia de todos estos conceptos en el diseño físico de las Bases de Datos, hay que cuidar, de forma especial, la coordinación temática y temporal entre ambas. Así, se garantizará la necesaria consistencia interdisciplinaria y se evitarán redundancias innecesarias.

- Habrá que tener en cuenta también el contenido de la asignatura de *Sistemas Operativos* (Troncal de 6 créditos teóricos). en la que se estudiarán las capacidades de un sistema moderno de computación, abordando en especial temas referentes a la gestión de recursos y la concurrencia de procesos y aspectos relativos a la seguridad.

Así mismo los contenidos en materias de Bases de Datos servirán como base a otras asignaturas de la titulación como son:

- *Ingeniería del Software* (Obligatoria de 6 créditos, 4.5 teóricos y 1.5 prácticos). La capacidad de hacer diseños de Bases de Datos sirve de apoyo a los contenidos de esta asignatura, en la que entre otros puntos se abordan los métodos de análisis y diseño estructurado. Dichos métodos necesitan que el alumno se haya enfrentado a problemas de modelado conceptual y lógico orientado a sistemas relacionales en la asignatura de *Diseño de Bases de Datos*. Por otro lado las prácticas de Ingeniería del Software tanto se enfoquen desde el desarrollo rápido de prototipos con herramientas RAD (*Rapid Application Development*) basadas en entornos de programación visual, como desde tecnología CASE (*Computer Aided/ Assisted Software Engineering*), necesitan respectivamente de conocimientos previos de programación en entornos de Bases de Datos y modelado de datos. Por todo ello hay que cuidar con particular esmero la coordinación entre la asignatura de Diseño de Bases de Datos y la de Ingeniería del Software.

Las dependencias e interrelaciones entre estas asignaturas se muestran en la Figura 1. En el establecimiento de estas dependencias se ha tenido en cuenta el factor tiempo, que claramente establece el orden lógico en el que se van a cursar las asignaturas.

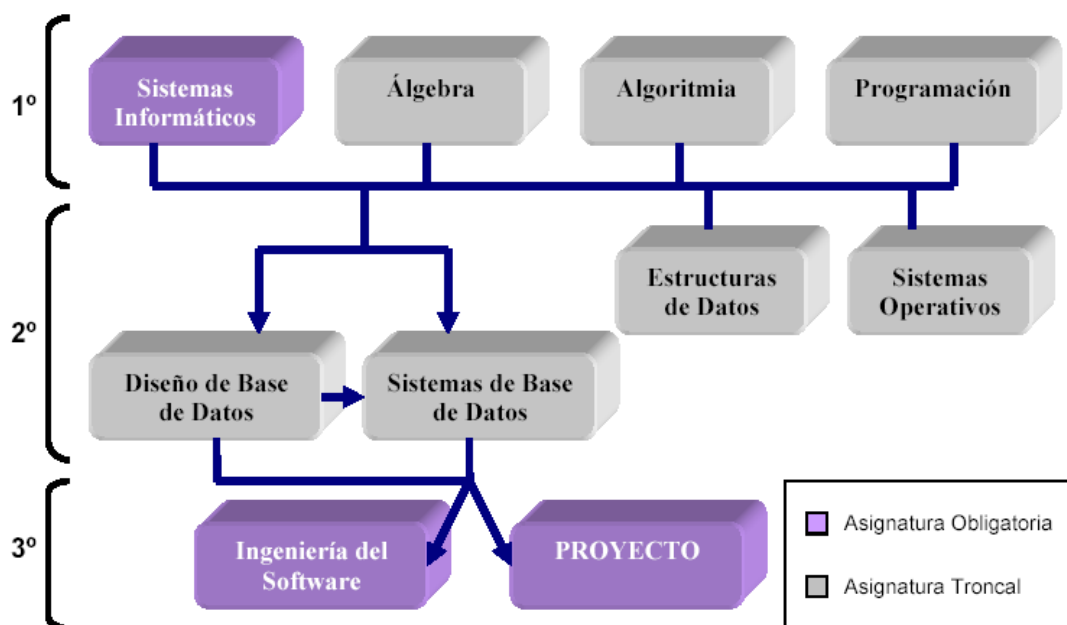


Figura 1. Relaciones de las asignaturas de Bases de Datos en el plan de estudios de ITIS

2. Objetivos

La elaboración de los programas de las asignaturas va a estar condicionada por el planteamiento global que se tenga del campo concreto de la informática objeto de estudio, por la visión personal de la mejor articulación docente de los núcleos principales en torno a los cuales este campo de conocimiento se estructura, y por las características del contexto en donde se va a desarrollar la actividad educativa.

Este contexto a su vez está delimitado fundamentalmente por las características particulares tanto de los estudios en donde esta materia se inserta, como de los alumnos destinatarios y su horizonte profesional al terminarlos, así como por los objetivos docentes, ya que éstos describen los resultados previstos de la enseñanza.

Estos parámetros delimitan los contenidos a transmitir y su articulación, justificando la inclusión de unos y el rechazo de otros, el tratamiento más profundo o intenso de ciertos temas, frente al más rápido o menos exhaustivo de otros.

Entendemos que la materia ha de enfocarse hacia la consecución por el alumno de dos clases de objetivos: *objetivos informativos*, coincidentes con el intento de dotar al alumno de un amplio conjunto de conocimientos; y unos *objetivos formativos*, que aluden a la adquisición de determinados hábitos de conducta mental, destrezas, habilidades, capacidades o aptitudes.

Los primeros serán más ampliamente tratados en el punto destinado al programa de la asignatura, aunque de forma general, se puede resumir en que el alumno debe:

1. Familiarizarse con la terminología, técnicas y razonamientos propios de la materia de bases de datos.
2. Entender, aprender y asimilar los conceptos básicos, ideas fundamentales y datos específicos, permitiendo crear nuevo material en un proceso de síntesis.
3. Desarrollar la capacidad de dividir un todo en sus partes fundamentales, analizando sus interacciones y su modo de organización.
4. Conocer las posibilidades que la aplicación de ideas y técnicas aprendidas le ofrecerán en su vida profesional.
5. Disponer de suficiente información y bibliografía que le permita ampliar sus conocimientos en una determinada materia, cuando sus necesidades así lo requieran.

En cuanto al segundo tipo de objetivos, los *formativos*, destacamos los siguientes:

1. El desarrollo de actitudes que sostengan la comprensión de los principios fundamentales, más valioso que la acumulación o la adquisición de habilidades y técnicas concretas.
2. Aprender a pensar y razonar sobre los distintos aspectos de la asignatura, desarrollando hábitos críticos que permitan al alumno formar un juicio ponderado. El objetivo será lograr la adquisición de la capacidad de valorar un objeto, una idea o un método de conducta.
3. El desarrollo de la habilidad suficiente para enfrentarse, aplicando los conocimientos adquiridos, con la resolución de casos prácticos similares a los que se le plantearán en su vida profesional.
4. La capacidad no sólo de comprender la extensión y significado de lo que ya se conoce en el campo donde se encuadra la disciplina, sino de ser receptivo ante lo nuevo, de afrontarlo y de trabajar con confianza de forma individual.

Es evidente la importancia que hoy se concede en los procesos formativos de cualquier índole y, mucho más en el contexto universitario, a la consecución de objetivos no informativos- “destrezas profesionales“- del alumno. En efecto, se asume que el aprendizaje se compone además de la adquisición de conocimientos, de la modificación de comportamientos. La práctica profesional debe incluirse además como un componente esencial de un currículum universitario. Estas prácticas deben estar orientadas no sólo a la aplicación de los conocimientos adquiridos, sino que deberían incluir la formación en gestión, ética y valores, comunicación oral y escrita, y trabajo en equipo.

En las asignaturas de Bases de Datos se pretende potenciar en el alumno la capacidad de abstracción y reflexión, la actitud crítica y analítica, la capacidad de selección y elección de problemas a plantear, generando hábitos de investigación y de trabajo individual y en grupo con el conocimiento tanto teórico como práctico de las técnicas y la tecnología existente en el área de las bases de datos. Es obvio que en un mundo donde es fundamental la innovación y se vive en un continuo avance tecnológico, los profesionales deberán estar capacitados no sólo para adaptarse al entorno sino también para realizar nuevas aportaciones. Por tanto, nuestras enseñanzas deben permitir la formación del alumno como futuro profesional sin que esto signifique que se vea menos capacitado para otras actividades sino, muy al contrario, que le permita incorporarse con éxito a la empresa del presente y del futuro. Se entiende que un objetivo básico ha de ser el fomentar la capacidad para relacionar los planteamientos teóricos y generales con las situaciones reales y más concretas de su entorno de trabajo.

La correcta y clara formulación de estos objetivos, tanto cognitivos como de actitudes, forma parte del propio programa de las asignaturas, y han de trasladarse a los alumnos no sólo como un elemento importante que orientará la docencia, sino como un criterio básico del proceso de evaluación del aprendizaje.

La Tabla 1 muestra los objetivos particulares que han de ser llevados a cabo en la impartición de la asignaturas de Diseño de Base de Datos. Aparece en la columna izquierda de dicha tabla su justificación más directa en base a currículos analizados en relación a dicha asignatura.

Objetivo	Reflejado en propuesta curricular
Conocer las características y arquitectura de un sistema de base de datos y comprender su modo general de funcionamiento	<ul style="list-style-type: none"> ▪ Objetivos de Bases de Datos I según [Blesa et al. 1999]¹ ▪ IM2 del CC2001² ▪ IS 20002.1 ³
Conocer la evolución que han sufrido los sistemas gestores de Bases de Datos a lo largo del tiempo, desde el inicio con los sistemas jerárquicos y en red, hasta la actualidad	<ul style="list-style-type: none"> ▪ Objetivos de Bases de Datos II[Blesa et al. 1999] ▪ IM2 del CC2001 ▪ IS 20002.1
Dar a conocer el papel y las funciones de la administración de Bases de Datos en el entorno de los sistemas de información en	<ul style="list-style-type: none"> ▪ Objetivos de Administración de Bases de Datos según [Blesa et al. 1999] ▪ IM2 del CC2001

¹ [Blesa et al. 1999] Blesa, P., Brisaboa, N., Canivell, V., Garbajosa, J. , Maudes, J.y Piattini, M."Propuesta de contenidos en Bases de Datos de los places de estudio de Informática". Revista NOVATICA nº 137. pp. 60 a 63. ene./feb. 1999

² [ACM/IEEE-CS, 2001].ACM/IEEE-CS Joint Curriculum Task Force. Computing Curricula 2001. Computer Science. IEEE Computer Society Press. Diciembre 2001

³ [IS 2002] Information Systems 2002(IS2002). ACM/AIS/AITP.

general y, más específicamente, en el corporativo	<ul style="list-style-type: none"> ▪ IS 20002.1 ▪ BCP-02 de ICF2000⁴
Realizar correctamente el diseño conceptual de una base de datos utilizando el modelo entidad-relación	<ul style="list-style-type: none"> ▪ Objetivos de Bases de Datos I y Bases de Datos II según [Blesa et al. 1999] ▪ IM3 del CC2001
Introducir la gestión de una base de datos relacional utilizando el lenguaje SQL	<ul style="list-style-type: none"> ▪ Objetivos de Bases de Datos I según [Blesa et al. 1999] ▪ IM5 del CC2001
Utilizar varios lenguajes formales y comerciales, para definir y manipular la información contenida en una base de datos	<ul style="list-style-type: none"> ▪ IM5 del CC2001
Realizar correctamente el diseño lógico de una base de datos relacional a través de la teoría de la normalización	<ul style="list-style-type: none"> ▪ Objetivos de Bases de Datos I según [Blesa et al. 1999] ▪ IM4 del CC2001 ▪ IM6 del CC2001
Tener una visión general sobre las líneas de evolución futuras de la tecnología de Bases de Datos	<ul style="list-style-type: none"> ▪ IM1 del CC2001 ▪ IM2 del CC2001 ▪ Objetivos de Bases de Datos II según [Blesa et al. 1999]

Tabla 2: Objetivos de la asignatura Diseño de Bases de Datos y justificación curricular

3. Competencias

3.1. Competencias instrumentales (CI)

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

3.1.1. Habilidades cognitivas

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

Generales

- CIC1: Conocer y entender las relaciones que mantiene la asignatura con el resto de asignaturas de la titulación a partir de la introducción explícita de ejemplos y comentarios, aprovechando puntos relevantes del temario.
- CIC2: Conocer, identificar y analizar los criterios y especificaciones apropiadas para problemas concretos de la asignatura, y planificar las correspondientes estrategias para darles solución
- CIC3: Ser capaz de realizar la especificación y diseño completo y normalizado de una base de datos.

Unidad Didáctica I: Conceptos básicos

- CIC4: Introducir al alumno en la disciplina de las bases de datos
- CIC5: Presentar los conceptos básicos sobre Bases de Datos
- CIC6: Proporcionar una visión global de la naturaleza y los objetivos de los sistemas Bases de Datos, presentando las ventajas que ofrecen

⁴ [ICF 2000]: Informatics Curriculum Framework 2000 for Higher Education

- CIC7: Presentar los diferentes niveles de abstracción que permiten ocultar la complejidad de las estructuras que representan la información y los lenguajes de Bases de Datos
- CIC8: Presentar las funciones y los componentes de un sistema de Bases de Datos
- CIC9: Distinguir los diferentes tipos de usuarios en función de la forma de interactuar con el sistema
- CIC10: Destacar las funciones del administrador de la base de datos
- CIC11: Proporcionar al alumno la estructura global de un sistema de Bases de Datos

Unidad Didáctica II: Modelos de Datos

- CIC12: Realizar una introducción al modelado de la Bases de Datos y como éste encaja en el ciclo de vida de los sistemas de información
- CIC13: Presentar el concepto de modelo de datos
- CIC14: Realizar una categorización de los modelos de datos
- CIC15: Identificar los modelos convencionales más ampliamente aceptados, que permiten la representación de los datos en términos de las características técnicas
- CIC16: Conocer los aspectos del modelado conceptual
- CIC17: Realizar correctamente el diseño conceptual de una base de datos utilizando el modelo entidad-relación
- CIC18: Presentar las características más básicas del modelo de datos relacional
- CIC19: Adquirir las técnicas para permitir la reducción de un esquema ER a tablas

Unidad Didáctica III: Bases de Datos Relacionales

- CIC20: Presentar los conceptos sobre Bases de Datos relacionales
- CIC21: Introducir al alumno en la semántica del modelo relacional
- CIC22: Resaltar la importancia de la independencia de datos
- CIC23: Caracterizar los lenguajes relacionales
- CIC24: Presentar la potencia expresiva del álgebra relacional
- CIC25: Familiarizar al alumno con los lenguajes predicativos
- CIC26: Analizar la equivalencia entre los tres lenguajes relacionales
- CIC27: Presentar las características principales del lenguaje SQL
- CIC28: Distinguir los sublenguajes de definición y manipulación de datos
- CIC29: Destacar que SQL es un lenguaje relacionalmente completo

- CIC30: Analizar las estructuras necesarias para utilizar SQL inmerso en un lenguaje anfitrión
- CIC31: Analizar los problemas que puede presentar una base de datos mal diseñada y presentar de forma intuitiva la teoría de la normalización
- CIC32: Introducir los conceptos de dependencias funcionales y multivaluadas
- CIC33: Estudiar la teoría de las dependencias funcionales
- CIC34: Presentar el procedimiento de descomposición por para obtener diseños equivalentes
- CIC35: Analizar algunos algoritmos que permiten obtener de manera sistemática diseños de base de datos con las mejores propiedades
- CIC36: Realizar correctamente el diseño lógico de una base de datos relacional a través de la teoría de la normalización
- CIC37: Analizar la necesidad de mejorar el tiempo de respuesta de una consulta
- CIC38: Presentar las etapas en que se divide el procesamiento de una consulta
- CIC39: Analizar las mejoras que se pueden incluir en cada una de las etapas
- CIC40: Aplicar diferentes estrategias de acceso a una consulta dada

3.1.2. Capacidades metodológicas (CM)

- CIM1: Ser capaz de tomar decisiones de manera razonada.
- CIM2: Tener capacidad de análisis y síntesis.
- CIM3: Ser capaz de manejar bibliografía relacionada con el diseño de bases de datos.
- CIM4: Ser capaz de realizar el diseño lógico normalizado de una base de datos dada en los distintos modelos de datos así como establecer su relación.
- CIM5: Ser capaz de analizar, mejorar y aplicar estrategias de acceso a una consulta dada.

3.1.3. Destrezas tecnológicas

- CIT1: Habilidades básicas de navegación por la Web y uso del resto de servicios de red para la obtención y manejo de la información relacionada con la asignatura.
- CIT2: Manejo avanzado de un sistema de procesamiento de textos para la realización de los informes de las prácticas y la documentación de la práctica voluntarias, así como de programas para la presentación de su defensa pública.

3.1.4. Destrezas lingüísticas (CIL)

- CIL1: Adquirir y utilizar con fluidez un buen lenguaje científico, tanto oral como escrito, siendo riguroso en las explicaciones de cualquier proceso dentro de la asignatura.
- CIL2: Conocer y utilizar la terminología usual de las Bases de datos, tanto en español como en inglés.

3.2. Competencias interpersonales

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

3.2.1. Competencias para tareas colaborativas (CIPTC)

- CIPTC1: Ser capaz de trabajar en equipo para resolver los problemas de modelado enunciados en los talleres prácticos de la asignatura.
- CIPTC2: Ser capaz de presentar en público una solución a un problema planteado y mantener un debate con el resto de la clase sobre la solución planteada, para así buscar colaborativamente la mejor solución al problema.
- CIPTC3: Ser capaz de realizar y defender la práctica voluntaria de la asignatura en equipo.

3.2.2. Compromiso con el trabajo (CIPTR)

- CIPTR1: Se ha de definir un plan de trabajo en el que el volumen de trabajo de todos los miembros del equipo sea similar.
- CIPTR2: Una vez finalizado el trabajo, todos los miembros del grupo deben conocer en profundidad todo el desarrollo realizado.
- CIPTR3: Se debe cumplir el plazo de entrega de los trabajos.
- CIPTR4: Se debe adquirir un compromiso ético entre todos los componentes del grupo.
- CIPTR5: Capacidad de autoformación y autodisciplina, incluyendo la gestión y organización eficaz del tiempo.

3.3. Competencias sistémicas (CS)

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

- CS1: Capacidad de aplicar los conocimientos, métodos y herramientas vistos en la asignatura de Bases de Datos a situaciones y problemas concretos del área de la Ingeniería Informática y de otras disciplinas relacionadas.
- CS2: Capacidad de aprender y aplicar, de forma autónoma e interdisciplinar, nuevos conceptos y métodos relacionados con cada asignatura.
- CS3: Motivación por la calidad y por la creatividad.
- CS4: Capacidad de adoptar el proceso marcado por el método científico y de ingeniería en el planteamiento y realización de trabajos diversos, tanto a nivel académico como profesional.
- CS5: Capacidad de asimilación y adaptación a la evolución del estado del arte en el ámbito de las bases de datos y de la Ingeniería Informática como profesión.
- CS6: Capacidad de comunicar problemas técnicos y sus soluciones a diversidad de audiencias.
- CS7: Capacidad para trabajar de manera eficaz dentro de un equipo de desarrollo.

- CS8: Habilidad para entender y explicar las dimensiones cuantitativas de un problema.
- CS9: Actualización de conocimientos en la disciplina para el desarrollo profesional.

4. Prerrequisitos

4.1. Competencias y contenidos mínimos

- Conocer los elementos básicos de un lenguaje de programación de 3ª generación.
- Conocer las principales formas de representar los algoritmos.
- Conocer los principios fundamentales del diseño de estructuras de datos.
- Conocer los principios fundamentales de almacenamiento de la información en el computador.

4.2. Plan de trabajo para la consecución de los prerrequisitos

Debido a que esta asignatura constituye el primer contacto con las bases de datos de los alumnos, se parte de conocimientos mínimos, los obtenidos durante su primer curso, para comenzar a trabajar en ella. Los prerrequisitos para el estudio y entendimiento de la asignatura Diseño de Bases de Datos se cubren fundamentalmente en las asignaturas de Sistemas Informáticos y programación del primer curso, y en las asignaturas de Estructura de Datos de segundo curso.

5. Temario

La asignatura *Diseño de Bases de Datos* es la primera asignatura en materia de Bases de Datos con que se van a encontrar los alumnos. Sus contenidos serán fundamentales para comprender los temas teóricos de la asignatura Sistemas de Bases de Datos, así como la realización de las prácticas. En la elaboración del programa se ha apostado por una estrategia cuyos ejes básicos son mantener, en primer lugar, lo fundamental de lo que clásicamente han sido contenidos de este tipo de asignaturas: modelado de datos, modelo relacional, lenguajes de consulta relacionales y normalización. El otro eje es no dejar que en ningún caso el alumno pierda de vista que el objeto final de un sistema gestor de Bases de Datos es servir de soporte al desarrollo de sistemas de información que trabajen con información persistente.

5.1. Unidades didácticas y desarrollo

Unidad Didáctica 1. Conceptos Básicos	
Tema 1: Introducción a los Sistemas de Bases de Datos	
	Introducción
	De los sistemas de ficheros a los sistemas de bases de datos
	Requisitos de los sistemas de bases de datos
	Concepto de base de datos
	Niveles de abstracción de una base de datos
Tema 2: El sistema de Gestión de Bases de datos (SGBD)	
	Concepto y funciones de un SGBD
	Lenguajes e interfaces de bases de datos
	Los distintos actores en las bases de datos
	La estandarización de la arquitectura de los SGBD e independencia de datos

Estructura general del SGBD
Historia de los sistemas de bases de datos
Unidad Didáctica 2. Modelos de Datos
Tema 3: Introducción al Modelo de Datos
Introducción
Los modelos de datos en el proceso de diseño de una bases de datos
Clasificación de los modelos de datos
Modelos de datos clásicos
Otros modelos de datos
Tema 4: El modelo entidad-relación
Conceptos básicos del modelo Entidad-Relación (ER)
Restricciones
Claves
Diagrama entidad-relación
Extensiones del modelo Entidad -Relación (EER)
La dimensión temporal en el modelo ER
Tema 5: El modelo de datos Relacional
Introducción al modelo de datos relacional
Transformación de un esquema E-R a un esquema relacional
Unidad Didáctica 3. Bases de Datos Relacionales
Tema 6: Las Bases de datos Relacionales
Introducción
La estructura de las bases de datos relacionales
Restricciones del modelo relacional
Los tres niveles de ANSI en el modelo relacional
Tema 7: Lenguajes Formales
Introducción
Álgebra relacional
Cálculo Relacional
Equivalencias
Tema 8: El Lenguaje estándar SQL
Introducción
Sublenguaje de definición de datos
Sublenguaje de manipulación de datos
Sublenguaje de control
Sublenguaje huesped
Tema 9: Diseño de Bases de Datos Relacionales
Anomalías en Inserciones, Borrados y Actualizaciones. Introducción a la Teoría de la Normalización
La primera Forma Normal
Las dependencias funcionales
Manipulación de Dependencias Funcionales: Cierre Transitivo de un Conjunto de Atributos y Recubrimiento Minimal
La Segunda Forma Normal
La Tercera Forma Normal
La Forma Normal de Boyce-Codd
El Proceso de Descomposición de las relaciones. Pérdidas de Producto y Pérdidas de Dependencias
Dependencias Multivaluadas
La Cuarta Forma Normal
La Desnormalización
Tema 10: Optimización de Consultas
Introducción
Etapas en el proceso de optimización

Optimización semántica
Optimización de las expresiones algebraicas
Estimación del coste de un plan de ejecución
Técnicas de descomposición de consultas

6. Metodología y estrategias de aprendizaje

6.1. Metodología docente

La adecuada combinación de las técnicas docentes, ha de contribuir a la creación de actividades críticas, reflexivas y analíticas en los alumnos de tal forma que puedan llegar a obtener una visión equilibrada e integradora de los distintos aspectos que conforman la los conocimientos para la asignatura de bases de datos.

Los planes de estudio que se definan dentro del EEES han de generar dinámicas activas en el aula que reasignen los papeles del alumno y el profesor exigiendo de ambos mucha más actividad. De este modo el alumno será más partícipe de su aprendizaje a la vez que el profesor deberá organizar y coordinar no sólo las actividades dentro del aula sino generar baterías de trabajo que permitan al alumno profundizar y generar su propio conocimiento más allá del trabajo en el aula. Se trata, pues, de incorporar una **metodología activa, globalizadora y participativa** en la medida en que las condiciones en las que el desarrollo de la actividad docente lo permita. Los cauces docentes básicos para una buena docencia se basan en tres pilares: la explicación, la motivación y la orientación. Estas prácticas se basan en fines educativos tales como:

- Que transmitan no sólo conocimientos sino también procedimientos, esquemas de razonamiento, mecanismos de aplicación, generalización y síntesis y, en definitiva, metodología científica
- Que permitan al alumno desempeñar un papel activo (investigar, exponer, observar, participar, etc.)
- Que le estimulen la necesidad de aprender y la iniciativa para la aplicación de sus conocimientos a problemas reales
- Que presenten una cierta flexibilidad para que el alumno pueda tomar decisiones razonables respecto a cómo desarrollarlas
- Que fomenten tanto el trabajo individual como en equipo

Las actividades con las que cuenta el modelo educativo que se va a seguir en la asignatura de Diseño de Bases de Datos mantienen en la clase magistral un elemento importante, así como la denominada clase activa que se complementarán, dentro de lo posible, propiciando una participación activa del alumnado pero con la constante intervención del profesor. El profesor ha de provocar el aprendizaje y las ganas de aprender en el propio alumno, motivándole a seguir trabajando en las actividades propuestas e incitándole a la búsqueda de información que solucione los problemas suscitados en el desarrollo de las actividades fuera del aula. La serie de actividades con que permiten el desarrollo docente serán las siguientes:

1. Clase teórica o lección magistral.
2. Clases de Problemas.
3. Clases Prácticas.
 - 3.1. Prácticas Guiadas.
 - 3.2. Prácticas Libres. Individuales y en grupo

4. Actividades Docentes Complementarias
 - 4.1. Seminarios y Conferencias.
 - 4.2. Visitas y prácticas en instalaciones y centros profesionales.
 - 4.3. Tutorías.
 - 4.4. Internet como vía de comunicación con y entre los alumnos

- *Clases de teoría con apoyo de material audiovisual* que se entregará al alumno con suficiente anterioridad como para que este pueda tenerlas disponibles durante la clase. Este material audiovisual (textos, transparencias, diagramas, etc.) será de apoyo al desarrollo de las clases a modo de guía pero de ningún modo constituirá el total de información a pedir al alumno. Este material audiovisual permite ilustrar la explicación y agilizar el desarrollo, sobre todo en temas con gran contenido de esquemas, diagramas y demás representaciones gráficas, como ocurre en esta asignatura de bases de datos, que de otro modo pueden ocasionar confusiones, pérdidas de tiempo y concentración por parte de los alumnos y del profesor, al ser dibujadas manualmente sobre la pizarra

Al inicio de la clase es conveniente dar una visión general del tema que permita seguir la exposición con facilidad. Al final de la clase, es aconsejable dar una conclusión que resuma lo expuesto. Si todavía no se ha llegado a dicha conclusión se debe de remarcar hacia donde se pretende llegar y mostrar el camino recorrido.

Entre el principio y el final debería de discurrir una exposición teórica correcta que suscite en el alumno el interés por el tema, motivándolo en el aprendizaje de la materia; avanzando con razonamientos claros, mostrando la relación entre los conceptos precedentes y/o consecutivos, y enfocada hacia los temas fundamentales, consiguiendo que el alumno no pierda en ningún momento una visión global de la asignatura.

- *Clases de problemas.* En ellas, el alumno debe aplicar los conocimientos teóricos adquiridos para resolver problemas o supuestos prácticos, constituyendo un eficaz factor de realimentación de los mismos. Su adecuada inclusión dentro del desarrollo de la materia permitirá reforzar y aplicar los conceptos expuestos en teoría, y fomentar en el alumno la capacidad de análisis y síntesis. No es conveniente establecer una división tajante entre clases teóricas y de problemas sino entremezclar ambas, de modo que las exposiciones teóricas se alternarán con ejercicios ilustrativos o aclaratorios. Asimismo, el profesor puede obtener una información muy valiosa de estas clases, ya que le permiten detectar dificultades de comprensión y aplicación de los conceptos teóricos.
- *Clases prácticas.* Parece interesante proponer, para afianzar los conocimientos teóricos, un conjunto de problemas, junto con alguna sugerencia de cara a su solución, que deben ser resueltos por los alumnos mediante su trabajo personal individual o en grupo. De esta forma el alumno completará el proceso de aprendizaje, razonando y enfrentándose a dudas y a conceptos poco claros, que de otro modo no se habrían presentado. Posteriormente, el profesor puede averiguar cuál es el grado de éxito en su solución, y orientar y resolver los más complejos con los alumnos. En caso de dificultades especiales, es aconsejable organizar un seminario para tratar estos problemas. Los enunciados de los

problemas, así como las soluciones aportadas por los alumnos y corregidas por el profesor pueden dejarse disponibles al alumno en fotocopidora o en la página Web de la asignatura.

- *Seminarios o conferencias.* Existen temas que no pueden ser tratados en toda su extensión durante un curso académico, bien por la propia limitación del tiempo asignado a la asignatura, bien porque un adecuado tratamiento de los mismos requiere del concurso de expertos, o bien por la inexistencia de instrumentos o medios especiales. En estos casos parece oportuno la utilización de seminarios o cursos monográficos. Los seminarios, en concreto, se pueden plantear desde varios enfoques. En primer lugar, el profesor o un experto invitado puede exponer un tema específico y, posteriormente, debatirlo con los alumnos en un ambiente más distendido e informal que el de las clases, intentando que los alumnos expongan sus dudas, comentarios o sugerencias. La discusión puede estar precedida de una conferencia o simplemente de algunas observaciones del individuo que representa el papel de director. Éste con antelación habrá preparado un esquema general para dirigir la discusión hacia determinadas metas.

Otra opción consiste en que sean los propios alumnos los que presenten un tema, previamente preparado, para proceder a continuación a su debate y discusión con el resto de los alumnos y el profesor. Este tipo de actividades sirve para fomentar las facultades expositivas de los alumnos y promover la crítica y la creatividad.

- *Visitas y prácticas en instalaciones y centros profesionales.* Por otra parte, también sería conveniente llevar a los estudiantes a los centros donde desempeñarán sus labores profesionales. En general es sumamente importante que los estudiantes vean de cerca los centros donde desarrollan su trabajo los especialistas en la materia, a fin de conocer de cerca la realidad de las enseñanzas explicadas en clase, y para que adquieran una visión directa y global del funcionamiento de esos centros. Otro beneficio a obtener de este tipo de contactos es el que el empresario conozca la formación de los alumnos con vistas a una posible contratación futura.
- *Tutorías.* El alumnado tiene a su disposición en la actualidad, seis horas de tutorías a la semana en las que puede consultar cualquier duda relacionada con los contenidos, organización y planificación de la asignatura. Las tutorías sirven para poner en contacto directo al profesor con los alumnos, fomentando la relación entre ambos. Este contacto mutuo debe ser utilizado por ambas partes: los alumnos para consultar al profesor todas las dudas surgidas, inquietudes, opiniones, perspectivas profesionales, buscar bibliografía específica, etc. El profesor debe buscar en dicho contacto los elementos de autoevaluación que le permitan detectar el grado de entendimiento y las dificultades que encuentran los alumnos en la materia, detectando los conceptos captados de forma deficiente y que, por tanto, necesiten una revisión del planteamiento expuesto en las clases teóricas o prácticas. En definitiva, el profesor puede y debe observar en las tutorías la marcha general y particular de la asignatura.
- *Posibilidades de la web.* Se convierte en el vehículo de comunicación y registro de información de la asignatura. Por una parte el material de carácter docente recopilado y publicado en la página *Web de la asignatura* es un activo de gran valor para la preparación la asignatura. Por otro lado el correo electrónico ofrece un medio de comunicación entre el alumno y el profesor.

6.2. Estrategias de aprendizaje

Se detalla las actuaciones concretas a realizar para la aplicación y aprovechamiento de la metodología propuesta. Se estructura en las siguientes fases:

6.2.1. Recopilación de la documentación de la asignatura

- Los alumnos dispondrán en la página web de la asignatura de toda la información y normativa relacionada con la asignatura: temario, criterios de evaluación, bibliografía, apuntes, enunciados de prácticas, trabajos, enlaces de interés, lecturas complementarias, avisos...
- El profesor mantiene actualizada la información de esta página para que se convierta en un vehículo de comunicación con los alumnos.
- Los alumnos deben conocer y manejar con fluidez toda la información y la documentación que se integra en la página de la asignatura.
- Asiduamente, el alumno debe consultar la página para estar actualizado.

7. Planificación del tiempo y del esfuerzo

La asignatura de Diseño de Bases de Datos tiene en el plan de estudios actual 4,5 créditos LRU asignados, todos ellos estimados como de teoría.

Para hacer su estudio en ECTS se ha tomado tomando la equivalencia de crédito LRU a crédito ECTS, con implicación de cambio de metodología docente.

Al hacer el estudio de carga de trabajo máxima, se tiene que:

- 1 ECTS -> 25 ó 30 horas de trabajo
- 4,5cr LRU -> 45 horas presenciales + (45*1,5) horas de asimilación = 135 horas de trabajo
- 4,5 ECTS estarían acotados dentro de los tiempos de trabajo siguientes: 112,5 ≤ 4,5 ECTS ≤ 180

Dos son los escenarios más probables: el resultante de tomar el crédito ECTS como 30 horas de trabajo y el resultante de tomarlo como 25 horas de trabajo.

Por temas de organización y calendario, se ha optado por partir del escenario que asigna 25 horas de trabajo al ECTS.

Para poder elaborar las guías, cuyo destino final sería la realización de experiencias piloto, dentro de los planes de estudio vigentes, se debe tener en cuenta la definición del crédito del R.D. 1497/1987 y sus posteriores modificaciones. Así:

- La unidad de valoración de las enseñanzas se corresponde a diez horas de enseñanza teórica, práctica o de sus equivalencias entre las que podrán incluirse actividades académicas dirigidas, que habrán de preverse en el correspondiente plan docente junto con los mecanismos y medios objetivos de comprobación de los resultados académicos de las mismas. Todo ello sin perjuicio del cumplimiento del régimen de dedicación del profesorado, de conformidad con el Real Decreto 898/1985, de 30 de abril sobre régimen del profesorado universitario.
- En ningún caso, salvo que se trate de enseñanzas en Universidades a distancia, el porcentaje del crédito correspondiente a las actividades académicas dirigidas será superior al **30%**.

De esta forma, si 4,5 créditos LRU equivalen a 45 horas, se debe asegurar el 70% como mínimo de presencialidad, esto es, 31,5 horas de docencia teórica. Este planteamiento, exige reestructurar el material docente y actualizar los contenidos para que en su desarrollo capaciten al alumno en los conocimientos y destrezas que ha de adquirir.

En la Tabla 1 se presenta la organización del esfuerzo del alumno para cubrir los 4,5 ECTS asociados a la asignatura. Por su parte en la Tabla 2 se distribuyen las horas presenciales entre los diferentes temas y el examen.

			A	B	C	D	E
	Técnica	Actividad	Horas equivalentes de clase	Factor de trabajo del alumno†	Horas de trabajo personal del alumno	Horas totales (A+C)	ECTS (D ÷ 25)
Teoría	Seguimiento de la parte teórica de las unidades didácticas	Asimila contenidos. Se plantea dudas que planteará a los profesores en las tutorías	28	1,5	42	70	2,8
Práctica	Seguimiento de la parte práctica de las unidades didácticas	Experimenta, practica, modela, diseña	15	1,5	22,5	37,5	1,5
Ejercicios de apoyo	Ejercicios de repaso y afianzamiento de lo aprendido. Pueden computar o no para la superación de la asignatura, dependiendo de la tipología de éstos	Realiza, resuelve problemas, tests...	-	-	5	5	0,2
Práctica obligatoria	Práctica en grupo de obligada realización	Realiza un Diagrama ER con paso a relacional completada con elementos de diseño	-	-	5	5	0,2
Examen	Controles de evaluación continua y/o examen final de la asignatura	Exámenes tipo test, supuestos prácticos...	3			3	0,12
Otras actividades	Tutorías personalizadas y grupales	Recibe orientación personalizada	-	-	2	2	0,08
	Búsquedas en la red, participación en foros especializados...	Busca elementos para completar los contenidos	-	-	1	1	0,04
TOTAL			44		77,5	120,5	4,82

Tabla 2. Escenario ECTS = 25 horas de trabajo

† Número de horas dedicadas por el alumno al trabajo personal (organización de apuntes, estudio, documentación...) por cada hora de clase.

Ajustando la presencialidad y el trabajo personal del alumno mediante la tabla 2, se concluye que la planificación de esta asignatura supone entorno a unas 120 horas, ajustándose dentro de los límites establecidos en el planteamiento inicial del ECTS ($112,5 \leq (4,5 \text{ ECTS} = 120) \leq 180$).

NÚMERO DE HORAS PRESENCIALES		
Actividad	Horas Teoría	Horas Prácticas
Unidad Didáctica I: Conceptos básicos		
Tema 1	1	
Tema 2	1	
Unidad Didáctica II: Modelos de Datos		
Tema 3	3	
Tema 4	3	3
Tema 5	3	2
Unidad Didáctica III: Bases de Datos Relacionales		
Tema 6	3	1
Tema 7	3	3
Tema 8	2	
Tema 9	4	4
Tema 10	2	2
Examen	3	
TOTAL: 43	28	15

Tabla 3. Reparto de horas presenciales

Distribución Temporal. Diseño de Bases de Datos

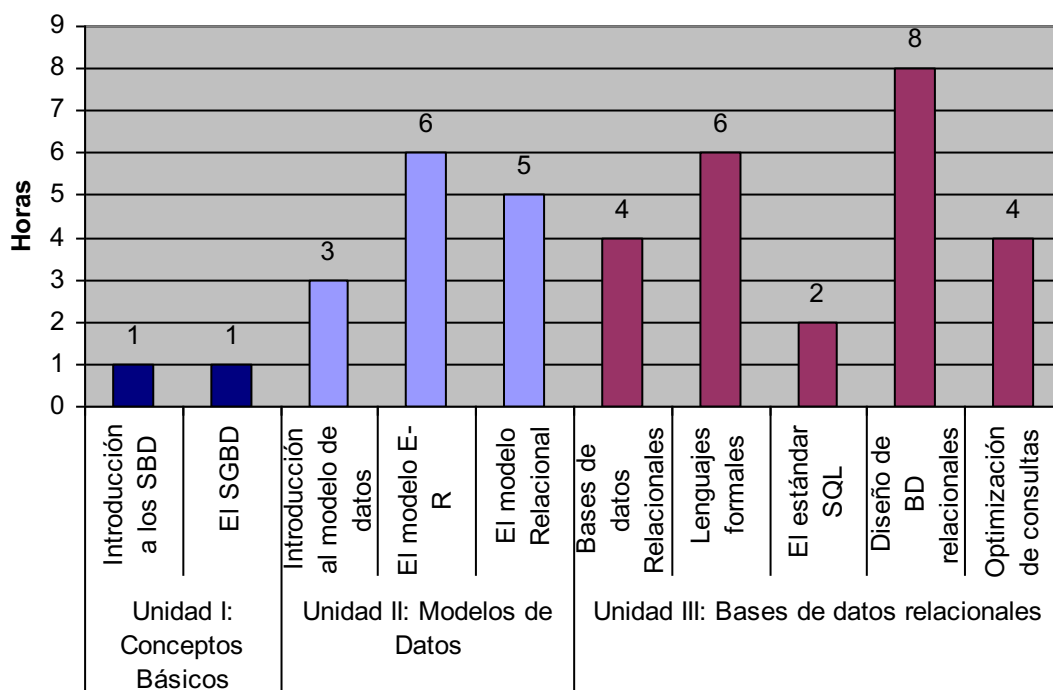


Figura 1 Distribución temporal del programa de la Asignatura Diseño de Bases de Datos

En la figura 1, se representa la carga horaria que se propone para cada uno de los temas. El esquema describe el desarrollo paulatino de cada aspecto del diseño de bases de datos,

primero los conceptos básicos, después el diseño en base a los modelos de datos y finalmente se concreta en capacitar al alumnos para diseño bases de datos relacionales. Se puede observar cómo se le asigna una mayor atención al tema de diseño de bases de datos relacionales. Con este esfuerzo se pretende que el alumno asimile en el mayor grado posible la importancia del diseño de bases de datos normalizadas, y por tanto, sea capaz de entender y realizar correctamente el diseño de una base de datos.

El siguiente tema en interés lo constituye el diseño de Bases de Datos mediante la teoría de normalización. Se considera fundamental para la formación del futuro ingeniero el dominio de la teoría de normalización, es por ello que se le otorga a este tema mayor número de horas.

8. Evaluación de los procesos y resultados de aprendizaje

8.1. Sistema de evaluación

La evaluación de esta asignatura se realiza mediante una prueba escrita, formada por dos partes; una primera que evalúa cuestiones teóricas y una segunda cuestiones prácticas.

- **Parte teoría** (40% de la nota final). Se realizarán un número suficiente de preguntas, en torno a unas 30, con no menos de 4 alternativas pero de respuesta única. Esto nos permite cubrir casi la totalidad del temario. Esta parte es eliminatoria, debiendo ser superada con al menos un 4 por el alumno para proceder a la calificación de la segunda.
- **Parte práctica** (60% de la nota final). En el apartado de cuestiones y/o problemas se plantea la realización de pruebas de las denominadas situación problemática con supuestos prácticos.

Se justifica mayor importancia en la nota final a la parte práctica debido al carácter fuertemente aplicado de la disciplina.

Por otro lado y de forma complementaria se oferta al alumno la posibilidad de realizar un trabajo voluntario e individual para el **Fomento del Trabajo Personal del Alumno**.

La idea es fomentar y valorar el trabajo personal del alumno fuera de clase. Para ello se propone al alumno la realización de trabajos sobre temas relacionados con la asignatura que pueden elegir de una lista proporcionada por el profesor. Estos trabajos serán publicados en Internet, para que sirvan de base a futuros trabajos. El carácter de este ejercicio es individual y voluntario, pero su calificación servirá para mejorar la nota final del alumno en un punto como máximo.

La nota final de la asignatura se calculará conforme a la siguiente fórmula:

Si (Teoría \geq 4,0) y (Práctica \geq 5,0)

Nota Final = (Teoría*0,4) + (Práctica*0,6)

Si no



Fin si

8.2. Criterios de evaluación

La calificación se hará conforme a las siguientes pautas:

Matrícula de honor

- La nota final es superior o igual a 9,7 puntos
- El conocimiento y la comprensión de la materia se extienden más allá del trabajo cubierto por el programa.
- La comprensión conceptual es sobresaliente.
- Los problemas relacionados con la asignatura son resueltos con soltura y se es capaz de razonar alternativas de solución estableciendo comparativas entre ellas.
- La participación en las diferentes actividades, incluyendo las clases, sobresale por su corrección y satisfacción.
- Ha realizado de igual modo la práctica voluntaria

Sobresaliente

- La nota final es superior o igual a 8,6 puntos
- El conocimiento y la comprensión de la materia son muy satisfactorios.
- La comprensión conceptual es sobresaliente.
- Los problemas relacionados con la asignatura son resueltos con soltura y se es capaz de razonar alternativas de solución.
- La participación en las diferentes actividades, incluyendo las clases, ha sido muy correcta y muy satisfactoria.

Notable

- La nota final es superior o igual a 6,75 puntos
- El conocimiento y la comprensión de la materia son satisfactorios.
- La comprensión conceptual es notable.
- Los problemas relacionados con la asignatura son resueltos con soltura.
- La participación en las diferentes actividades, incluyendo las clases, ha sido correcta y bastante satisfactoria.

Aprobado

- La nota final es superior o igual a 5 puntos
- El conocimiento y la comprensión de la materia son básicos.
- La comprensión conceptual es suficiente.
- Los problemas relacionados con la asignatura son resueltos con éxito razonable.
- La participación en las diferentes actividades, incluyendo las clases, ha sido correcta pero no siempre satisfactoria.

Suspenso

- La nota final es inferior a 5 puntos
- El conocimiento y la comprensión de la materia no son aceptables.
- La comprensión conceptual es insuficiente.

- Los problemas relacionados con la asignatura no se resuelven de forma adecuada.
- La participación en las diferentes actividades, incluyendo las clases, ha sido nula o escasa y deficiente.

9. Bibliografía

9.1. Bibliografía básica comentada

Date, C.J. "Introducción a los Sistemas de Bases de Datos". 7ª Ed. Pearson Education, 2001

El libro es una versión en castellano de la obra "*An introduction to database systems. 7th edition*", de C.J. Date publicado en el año 2000. Se trata de la séptima edición de este clásico dentro del mundo de las Bases de Datos. Contiene importantes cambios respecto a la versión anterior, con actualizaciones y modificaciones en cada uno de sus temas, es notable en este aspecto la profundización realizada en aspectos de denormalización y diseño ortogonal, así como enfoques del modelado semántico. Lo más relevante está en la inclusión de nuevos temas relativos a los desarrollos presentes de las bases de datos tales como bases de datos temporales, la aplicación de las bases de datos al apoyo de la toma de decisiones. Cada capítulo incluye ejercicios e información adicional sobre los temas tratados en ellos. Es un buen libro de cabecera para el programa de la asignatura.

C. J. Date. "An Introduction to Database Systems". 8th Ed. Addison Wesley Longman, 2003

Se trata de una reedición de la anterior obra (7ª edición), revisada en su traducción en español, que presenta como novedades la inclusión de más temas y más ampliados los que ya estaban, entre ellos incluye un tema sobre Bases de Datos Relacionales/Objetuales y otro tema sobre WWW y XML relacionado con las Bases de Datos. Por ello puede ser, de mayor utilidad si cabe, para cualquiera de las dos asignaturas que centran la atención de la presente memoria.

Sigue, por tanto, siendo un formidable y exhaustivo libro de introducción a las Bases de Datos, siempre en constante actualización. La desventaja, siempre debido a la reticencia del alumnado medio de no haber sido todavía traducido al castellano a fecha de la realización de la presente memoria.

R. Elmasri, S. B. Navathe. "Fundamentos de Sistemas de Bases de Datos". 3ª Ed. Addison Wesley. 2002

El libro es una versión en castellano de la obra "*Fundamentals of Database Systems*". 3thEd. Benjamin/Cummings, Ed. Publicada en el 2000. Aunque hay que hacer notar que acaba de salir en el 2004 la 4ª edición de la obra original en inglés. Se trata de un buen libro de introducción a las Bases de Datos que en general es bastante completo, cubriendo aspectos que van desde los lenguajes, al diseño y a las técnicas de implementación.

De cara a la asignatura de *Diseño de Bases de Datos* es quizás más notorio el tratamiento que, con una profundidad adecuada, hace de las partes de diseño conceptual y lógico, lenguajes de interrogación de Bases de Datos relacionales (álgebra y cálculo) y sistemas heredados.

Este libro se hace también necesario en muchos aspectos de la asignatura de *Sistemas de Bases de Datos*, pues dedica también un amplio espacio a relatar la implementación física de los sistemas gestores de Bases de Datos.

Como novedades respecto a la segunda edición aporta la inclusión de apartados dedicados a aspectos avanzados, como las los sistemas de bases de datos orientadas objetos y los sistemas objeto-relacionales, así como un tratamiento más completo y actualizado del modelo relacional. Incluye además un tratamiento actualizado de todos estos conceptos en distintos SGBD, tales como Oracle, Microsoft Access, INFORMIX, etc.

A. Silberschatz, H.F. Korth y S. Sudarshan. "*Fundamentos de Bases de Datos*". 4ª Edición. Ed. McGraw-Hill, 2002

Se trata de la cuarta edición de este libro, que puede considerarse un clásico en el mundo de las Bases de Datos. Se trata de la traducción al castellano de la obra "*Database System Concepts*" 4^{ed} Ed. Mc Graw Hill de 2001. Incluye temas relacionados con las nuevas áreas de investigación como el XML en Bases de Datos y tres capítulos nuevos de estudio de sistemas de bases de datos comerciales líderes: Oracle, IBM, DB2 y Microsoft SQL Server. Se han añadido además al final de cada capítulo una sección con información sobre herramientas software referido, además de nuevos ejercicios y referencias actualizadas.

Adicionalmente existe una dirección *Web* del libro⁵ donde se pueden encontrar transparencias asociadas a los temas, así como recomendaciones de distribución de los contenidos del libro en el programa de una asignatura de Bases de Datos. Estos materiales están actualmente incluso revisados y llevados al castellano en la Web de la editorial McGraw Hill⁶.

Se recomienda la adquisición de esta obra a los alumnos, pues además de ser completa y actualizada, vale tanto para la asignatura de Diseño de Bases de Datos como Sistemas de Bases de Datos. Aunque es en Sistemas de Bases de Datos donde se utiliza como libro de referencia fundamental.

A. de Miguel y M. Piattini. "*Concepción y diseño de Bases de Datos. Del modelo E-R al modelo relacional*". Ed. Ra-ma, 1993

Este libro proporciona unos conocimientos básicos en la materia, sin olvidar, por ello, el adecuado rigor teórico. Presenta, de forma clara y precisa, los conceptos fundamentales de las Bases de Datos, ofreciendo un tratamiento sistemático de los tres modelos de datos convencionales más difundidos en el mercado. Explica, también, el modelo E-R extendido, enfatizando la importancia de un modelo conceptual y proporcionando unos principios metodológicos que ayudan a realizar un buen diseño conceptual. Suministra, también una buena base teórica para el diseño lógico de las Bases de Datos. Fundamentalmente realiza una detallada revisión sobre el proceso de creación de una base de datos y en especial los aspectos relativos al diseño en el modelo relacional. Trata, de una forma bastante amplia, algunos de los temas propuestos y que, por tanto, puede servir al alumno tanto para introducirse en la materia como para profundizar en diversos temas.

⁵ <http://www.cs.yale.edu/homes/avi/db-book/index.html>

⁶ <http://www.mcgraw-hill.es/ocl/silberschatz>

A. de Miguel, M. Piattini. “Fundamentos y Modelos de Bases de Datos”. 2ª Edición, RA-MA, 1999

Se trata de un libro que surge ya en su primera edición con la intención de dirigirse a una audiencia amplia que incluye a alumnos en su primera asignatura dedicada a las Bases de Datos, a alumnos de cuarto ciclo de Formación Profesional y profesionales y usuarios informáticos. Por ello, su atractivo es evidente de cara a afrontar las asignaturas de Bases de Datos que trata esta memoria. Esta obra destaca la distribución racional de sus temas básicos que completa con la inclusión de los sistemas orientados a objeto y la revisión de los contenidos dedicados a SQL-3 y a confidencialidad en Bases de Datos.

A. de Miguel, M. Piattini, E. Marcos. “Diseño de Bases de Datos Relacionales”. RA-MA, 1999

Este libro es el complemento ideal de [Miguel & Piattini 1999], ya que profundiza en todos los aspectos de diseño de bases de datos relacionales, presentando una metodología completa que abarca desde el modelado conceptual al diseño físico. También analiza a fondo y de manera rigurosa la teoría de la normalización.

A. de Miguel, P. Martínez, E. Castro, J.M. Cavero, D. Cuadar, A.M. Iglesias, C. Nieto. “Diseño de Bases de Datos Relacionales. Problemas Resueltos”. RA-MA, 2001

Este libro constituye una herramienta fundamental en la docencia de las asignaturas de bases de datos objeto de esta memoria. Resulta una tarea complicada encontrar libros de problemas que se adapten de manera tan sencilla a los cursos de bases de datos descritos. A esto hay que añadir que este libro constituye un material bien estructurado y completo con un método eficaz en el que todos los problemas se acompañan de la solución comentada. Reseñar que dicho libro completa una serie, convirtiéndose en un material muy valioso, llevado por el hilo conductor de Adoración de Miguel, referente de solidez profesional en las todas actividades relacionadas con las bases de datos en España, tanto de investigación como de docencia.

L. Grau, I. López. “Problemas de Bases de Datos”. Ed. Sanz y Torres, 2ª Edición, 2001

Este libro constituye un apoyo a asignaturas de base de datos para los alumnos de informática de la UNED en el segundo curso de ITIG e ITIS. Aunque pretende dar cabida a todos los aspectos básicos que se han de tratar en un curso introductorio, destaca sobre todo por su amplia recopilación de problemas referentes a álgebra relacional y cálculo relacional tanto de tuplas como de dominios. Anotar si cabe, el tema de problemas de normalización, donde aparece una serie amigable de 10 problemas muy básicos e intuitivos

que pueden constituir un acercamiento en el desarrollo de clases teóricas a la normalización.

G. W. Hansen, J. V. Hansen. "Diseño y Administración de Bases de Datos". 2ª Edición Prentice-Hall. 2000

Es la traducción al Castellano, en su 3ª reimpresión, del libro "*Database Management and Design*". 2nd Ed. Prentice-Hall, publicado en 1997. Se trata de un libro de carácter general que cubre buena parte de los aspectos de las asignaturas tratadas en este proyecto docente, convirtiéndolo en un libro adecuado de consulta para el alumno. Personalmente pienso que la tipografía y la edición del libro no ayudan en absoluto a que el libro sea amigable, aunque evidentemente los contenidos, que debería de ser lo que pesara son relevantes.

Entre los contenidos más relevantes del libro cabe destacar: el modelo cliente/servidor (Tema 9), la introducción a la administración de Bases de Datos incluyendo aspectos de control y recuperación (Tema 11), la implantación y ciclo de vida de una base de datos (Temas 3 y 13) y los sistemas heredados (Temas 15 y 16), donde resalta, entre otras, la descripción de los métodos de acceso en *IMS*.

S.M. Deen. "Fundamentos de los Sistemas de Bases de Datos". Ed. Gustavo Gili, 1987

Este libro trata fundamentalmente de los principios generales de las Bases de Datos, utilizando el modelo relacional y de red como vehículo fundamental para la realización de esos principios. Son especialmente interesantes los capítulos cuarto y quinto que tratan cuestiones de arquitectura y facilidades operativas de las Bases de Datos, respectivamente. En general, es un libro que se adecua bien al programa propuesto.

G. Gardarin. "Bases de Datos". Ed. Paraninfo, 1987

Se trata de un libro muy equilibrado entre el tratamiento formal y riguroso de los problemas y la presentación de las soluciones técnicas que se han encontrado. Recorre toda la problemática tradicional de las Bases de Datos, estableciendo claramente las definiciones de los conceptos y facilitando la comunicación con el lector.

M. L. Gillenson. "Introducción a las Bases de Datos". Ed. McGraw-Hill, 1988

El libro aborda, con un lenguaje sencillo, los fundamentos técnicos de las Bases de Datos, describiendo los diferentes métodos que se utilizan para organizar, almacenar y tener acceso a los datos. Examina temas como administración de datos, seguridad, recuperación, concurrencia, Bases de Datos distribuidas y máquinas para Bases de Datos, a un nivel muy asequible.

I.T. Hawryskiewicz. "*Database analysis and desing*". Ed. Macmillan, 1991

El libro presenta el ciclo completo para el diseño de una base de datos, esbozando los campos de estudio, la teoría que se ha desarrollado sobre ellos y las técnicas construidas. Es un libro que recoge resultados e investigaciones efectuados sobre el área del diseño de las Bases de Datos. En particular, se presentan la teoría relacional, el modelado semántico, el software de gestión, el diseño físico de ficheros, las Bases de Datos orientadas al objeto y las deductivas. La mayor parte del texto cubre aspectos técnicos, y sólo en algunas partes se combinan con planteamientos teóricos. Los primeros capítulos se ocupan del desarrollo del modelo de la empresa y de las especificaciones de las Bases de Datos; los últimos capítulos cubren aspectos físicos. Cabe destacar sobre todos los temas recogidos el estudio que realiza sobre los modelos semánticos y la colección de problemas propuestos que contiene.

A. Lucas Gómez. "*Diseño y gestión de sistemas de Bases de Datos*". Ed. Paraninfo, 1993

Este libro es una introducción a las Bases de Datos en general, que analiza todos los componentes que integran un sistema de Bases de Datos. La primera parte aborda su diseño y los principales modelos, aunque el libro está enfocado al modelo relacional. La segunda parte trata los lenguajes que se utilizan para acceder a los sistemas, con el propósito de manejar la información. Profundiza en el lenguaje SQL y presenta, de un modo más general, el DL/I de IMS y el lenguaje Natural de Adabas.

E. Rivero Cornelio. "*Bases de Datos relacionales*". Ed. Paraninfo, 1992

Este libro presenta los conceptos básicos del modelo relacional de datos, la teoría de la normalización y un método de diseño basado en gráficos de entidades, relaciones y atributos. Los temas se presentan con un buen estilo didáctico, de manera que los nuevos conceptos y definiciones se introducen justificándolos en función del problema que intenta resolver. El libro está orientado a conseguir diseñar adecuadamente estructuras relacionales de datos que sean coherentes con el significado de estos. No contempla otros aspectos importantes como concurrencia de accesos, protección de la confidencialidad, optimización de consultas, etc. Incluye, sin embargo, un gran número de ejemplos y ejercicios con sus soluciones.

J. D. Ullman. "*Principles of Database Systems*". 2ª Edición, Ed. Computer Scienice Press, 1982

Es otro de los textos clásicos de Bases de Datos. Es un libro que plantea con mucho rigor todos los temas que trata, demostrando muchos de los resultados que presenta. Introduce el modelo relacional haciendo uso de la teoría matemática asociada. Gran parte del libro se encarga de describir las relaciones, el álgebra y el cálculo, y los lenguajes de consulta que han sido diseñados utilizando los conceptos anteriores. Incluye, también, una explicación de cómo la teoría de las Bases de Datos relacionales se puede utilizar para diseñar buenos sistemas y una presentación de los primeros trabajos realizados sobre la optimización de consultas.

Luque I., Gómez-Nieto M.A. Diseño y uso de bases de datos relacionales. Rama 1997

Este libro desarrolla los aspectos más elementales sobre el lenguaje de consulta SQL. Dispone de nueve capítulos en los que utilizando una serie de ejemplos con enunciados cercanos a la vida cotidiana, realiza el modelo conceptual, el esquema relacional y un conjunto de preguntas en SQL que ejemplificaran la manipulación de la información almacenada en la base de datos. Este texto destaca por ser un manual de iniciación en los conocimientos sobre las bases de datos y sobre el lenguaje SQL. Posee una visión muy pedagógica de la enseñanza del diseño de bases de datos de manera básica.

9.2. Bibliografía complementaria

- Battini, C., Ceri, S. y Navathe, S.B. "Diseño Conceptual de Bases de Datos: Un Enfoque de Entidades-Interrelaciones". Addison-Wesley/Díaz de Santos. 1994
- Benavides, J., Olaizola, J.M. y Rivero Cornelio, E. "SQL Para usuarios y Programadores". Ed. Paraninfo, 1991
- Cairo, O. y Guardati, S. "Estructuras de datos". Ed. McGraw-Hill, 1993
- Cárdenas, A.F. "Sistemas de administración de bancos de datos". Ed. Limusa, 1983
- Celma, M., Casamayor, J.C., Mota, L. "Bases de Datos Relacionales". Prentice Hall, 2003
- Date, C.J. y Darwen, H... "A guide to the SQL standard: A User's Guide to the standard relational language SQL (3rd. Edition)". Addison-Wesley, 1993
- Date, C. J. "Bases de Datos. Una guía práctica". Ed. Addison-Wesley, 1987
- Fernández Baizán, M. C. "El Modelo Relacional de Datos: de los Fundamentos a los Modelos Deductivos". Díaz de Santos. 1987
- Folk, M.J. y Zoellick, B. "Estructuras de archivos". Ed. Addison-Wesley, 1992
- Frost, R. "Bases de Datos y sistemas expertos". Ed. Diaz de Santos, 1989
- Hughes, J.G. "Object-Oriented Databases". Ed. Prentice Hall, 1991
- Jackson, G.A. "Introducción al diseño de Bases de Datos Relacionales". Ed. Anaya Multimedia, 1990
- Kim, W. y Lochovsky, F.H. "Object-Oriented Concepts, Databases, and Applications". Ed. ACM Press, 1989

- Kroenke, D. "Procesamiento de Bases de Datos. Fundamentos, Diseño e Implementación". 8ª edición. Prentice Hall, 2003
- Marín, F., Quiros, A. y Torres, A. "Dbase IV/SQL: Introducción a las Bases de Datos relacionales". Ed. Ra-ma, 1989
- Özsu, M. T., Valduriez, P. "Principles of Distributed Database Systems". Prentice Hall, 1999
- Rob, P., Coronel, C. "Sistemas de Bases de Datos. Diseño, Implementación y Administración". Thomson Learning, 2004
- Tsai, A. "Sistemas de Bases de Datos: Administración y uso". Ed. Prentice Hall, 1990
- Ullman, J.D.; Widom, J. "Introducción a las Bases de Datos" Prentice-Hall International, 1999.
- Widom, J. y Ceri, S. "Active Database Systems: Triggers and Rules for Advanced Database Processing". Morgan Kaufmann. 1996

9.3. Otros recursos y/o Herramientas para la docencia de la asignatura

- **Reno 3.0**
 - Software incluido con el libro "Diseño de Bases de Datos Relacionales" (De Miguel, Piattini y Marcos, Ra-Ma, 1999). Sirve para aplicar los algoritmos de manejo de dependencias funcionales y de normalización (hasta FNBC) a esquemas relacionales.
- **WinRDBI**
 - <http://www.eas.asu.edu/~cse412/winrdbi.html>
 - Herramienta educativa, desarrollada por la *Arizona State University*, para aprender los lenguajes relacionales formales practicándolos. Para ello, funciona con un interfaz de usuario amigable en entorno Windows que posibilita la creación de bases de datos relacionales, la inserción de contenidos en dichas bases de datos, y la formulación de consultas en los siguientes lenguajes relacionales:
 - Álgebra Relacional (AR)
 - Cálculo Relacional de Tuplas (CRT)
 - Cálculo Relacional de Dominios (CRD)
 - SQL (versión simplificada de SQL-92)
- **WinSQL**
 - <http://www.synametrics.com/SynametricsWebApp/WinSQL.jsp>
 - Software cliente que permite el acceso a múltiples servidores de bases de datos relacionales (Oracle, DB2, Sybase, SQL Server, MySQL, etc.) desde Windows mediante el protocolo ODBC (*Open Database Connectivity*).

9.4. Enlaces de interés

- **ACM/SIGMOD- Special Interest Group on the Management Of Data.**
 - <http://www.acm.org/sigmod/>
 - El grupo de interés especial de la ACM dedicado a la gestión de datos se preocupa de los principios, técnicas y aplicaciones de los sistemas de gestión de bases de datos y de la tecnología de gestión de datos. Sus miembros incluyen desarrolladores de software, investigadores de los mundos académico e industrial, practicantes, usuarios y estudiantes. SIGMOD patrocina la conferencia anual SIGMOD/PODS. Publica revistas, colecciones de literatura y publicaciones, así como otros materiales en papel o digital.
- **IEEE TCDE- *Technical Committee on Data Engineering.***
 - <http://www.ccs.neu.edu/groups/IEEE/tcde/index.html>
 - El comité técnico de ingeniería de datos de la *IEEE Computer Society* se preocupa del papel de los datos en el diseño, desarrollo, gestión y utilización de sistemas de información. Los aspectos de mayor interés incluyen el diseño de bases de datos, el conocimiento de los datos y su procesamiento, los lenguajes para describir los datos, definir el acceso y la manipulación de bases de datos, las estrategias y mecanismos para el acceso a los datos, la seguridad y control de integridad, y los sistemas distribuidos. El TCDE patrocina la *International Conference on Data Engineering* (ICDE) y publica trimestralmente el *Data Engineering Bulletin*.
- **VLDB Endowment- *Very Large Data Base Endowment Inc.***
 - <http://www.vldb.org/>
 - Es una fundación estadounidense sin ánimo de lucro con el propósito de promover e intercambiar trabajos eruditos en bases de datos y áreas relacionadas en todo el mundo. Sus actividades principales son la organización de las conferencias VLDB y la publicación de la VLDB Journal, en colaboración con Springer-Verlag.
- **EDBT- *Extending Database Technology.***
 - <http://www.edbt.org>
 - La fundación EDBT es una organización apolítica y sin ánimo de lucro con el objetivo de promover y apoyar el progreso en los campos de las bases de datos y la tecnología y aplicaciones de los sistemas de información. Su mayor actividad es la promoción de la *International Conference on Extending Database Technology* (EDBT), que se celebra bienalmente desde 1988. La fundación también promueve escuelas de verano internacionales, desde 1991.
- **ODMG - *Object Database Management Group.***
 - www.odmg.org

- Se trata de un grupo de fabricantes y usuarios de bases de datos que desarrolla estándares para los sistemas de gestión de bases de datos orientadas a objetos.
- **ACM/SIGKDD** - *Special Interest Group on Knowledge Discovery in Data and Data Mining*.
 - <http://www.acm.org/sigkdd>
 - La tarea principal del SIGKDD es proporcionar un foro para el avance y la adopción de la “ciencia” del descubrimiento de conocimiento y la minería de datos. Para ellos, el SIGKDD fomenta la investigación básica en KDD (a través de conferencias de investigación anuales, un boletín y otras actividades relacionadas), la adopción de “estándares” en el mercado sobre terminología, evaluación y metodología, así como la educación interdisciplinar entre investigadores, practicantes y usuarios del KDD. Las actividades concretas del SIGKDD incluyen la conferencia anual de Knowledge Discovery and Data Mining y el boletín SIGKDD Explorations.

En menor medida, existen otras organizaciones, como la ACM/SIGMIS⁷ - *Special Interest Group on Management Information Systems* o el *Transaction Processing Performance Council*⁸, la AIS⁹ (Association for Information Systems), la AITP¹⁰ (*Association of Information Technology Professionals*, antes DPMA), la SIM¹¹ (*Society for Information Management*), la IACIS¹² (*International Association for Computer Information Systems*), y la IAIM¹³ (*International Academy for Information Management*).

- **A First Course in Database Systems; de Ullman y Widom.**
 - <http://www-db.stanford.edu/~ullman/fcdb.html>
 - libro original en inglés de "Introducción a los Sistemas de Bases de Datos". Prentice Hall, 1999; uno de los libros incluidos como básicos en nuestra asignatura.
 - Incluye propuestas de proyectos de diseño e implementación de bases de datos, ejercicios resueltos, transparencias, e información de ORACLE para prácticas de laboratorio.
- **Mylopoulos**

⁷ <http://www.acm.org/sigmis/>

⁸ <http://www.tpc.org/>

⁹ <http://www.aisnet.org/>

¹⁰ <http://www.aitp.org/>

¹¹ <http://www.simnet.org/>

¹² <http://www.iacis.org/>

¹³ <http://www.iaim.org/>

- <http://www.cs.toronto.edu/~jm/340S/>
- <http://www.cs.toronto.edu/~jm/2507S/>
- Curso sobre “Information Systems Analysis and Design” y “Conceptual Modeling” de la Universidad de Toronto (Canadá). Mylopoulos es uno de los más conocidos especialistas mundiales en modelado conceptual.

Guía Docente de Ingeniería del Software

Versión 2.0 – 26-9-2006

Francisco José García Peñalvo

Departamento de Informática y Automática

Facultad de Ciencias – Universidad de Salamanca

Plaza de los Caídos S/N, 37008, Salamanca, España

Tfno. +34 923 294400. Ext. 1302

Fax. +34 923 294514

fgarcia@usal.es

1. Contexto

1.1. Perfil de los créditos y su adecuación al perfil profesional y académico de la titulación

Para un ingeniero en informática es imprescindible tener unos conocimientos robustos de cómo desarrollar un sistema software, teniendo en cuenta todas las fases de su ciclo de vida, desde que surge la necesidad de realizar dicho sistema hasta su implementación.

La *Ingeniería del Software*, que tradicionalmente ha aparecido dentro de la Ciencia de la Computación, en los últimos años está siendo considerada como una entidad curricular independiente, aunque con profundas raíces en la Ciencia de la Computación y las Matemáticas. De hecho, se ha definido un cuerpo de conocimiento para la Ingeniería del Software (*Software Engineering Body of Knowledge – SWEBOK*), y más recientemente se ha publicado el currículo público (2004) del *Computing Curriculum – Software Engineering* por parte de la acción conjunta entre IEEE-CS y ACM, que se mantiene como uno de los cinco perfiles profesionales en el *Computing Curricula 2005*, junto a los perfiles de *Computer Engineering*, *Computer Science*, *Information Systems* e *Information Technology*.

La Ingeniería del Software como asignatura dentro del Plan de Estudios actual de la Ingeniería Técnica en Informática de Sistemas, consta de 6 créditos LRU (4,5 teóricos y 1,5 prácticos). Su cometido es presentar las actividades técnicas e ingenieriles que se llevan a cabo en el ciclo de vida de un producto software. Además, describe los problemas, principios, métodos y tecnologías asociadas con la Ingeniería del Software. No se puede obviar la importancia que tienen los requisitos en el éxito de un proyecto software, por ello la ingeniería de requisitos debe ser objeto de un estudio detallado, introduciendo las técnicas básicas de obtención, documentación, especificación y prototipado de los requisitos de un sistema software. Por otra parte, se debe introducir al alumno en la sistematización de la creación del software mediante el uso de métodos de ingeniería del software, tanto desde el punto de vista estructurado, típicamente tomando el método de Yourdon como referencia principal, como desde un punto de vista orientado a objetos, tomando UML (*Unified Modeling Language*) y el Proceso Unificado como bases fundamentales. En esta asignatura es demasiado ambicioso afrontar en profundidad los métodos de análisis y diseño, por ello se enfoca mayormente desde el análisis de sistemas y de requisitos, dejando los aspectos más

relacionados con el diseño para otras asignaturas. Para la parte práctica de esta asignatura se debe hacer uso de herramientas CASE (*Computer Aided/Assisted Software Engineering*).

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

1.2. Ubicación y relaciones en el plan de estudios

La asignatura de Ingeniería del Software se imparte en el primer cuatrimestre del tercer curso de la Ingeniería Técnica en Informática de Sistemas, cuyo Plan de Estudios actual data de 1997. Esta asignatura se relaciona directamente con otras de este mismo Plan de Estudios, pero además con otras de la titulación de segundo ciclo Ingeniería en Informática, cuyo Plan de Estudio data de 1998.

Las dependencias e interrelaciones entre estas asignaturas se muestran en la Figura 1. En el establecimiento de estas dependencias se ha tenido en cuenta el factor tiempo, que claramente establece el orden lógico en el que se van a cursar las asignaturas.

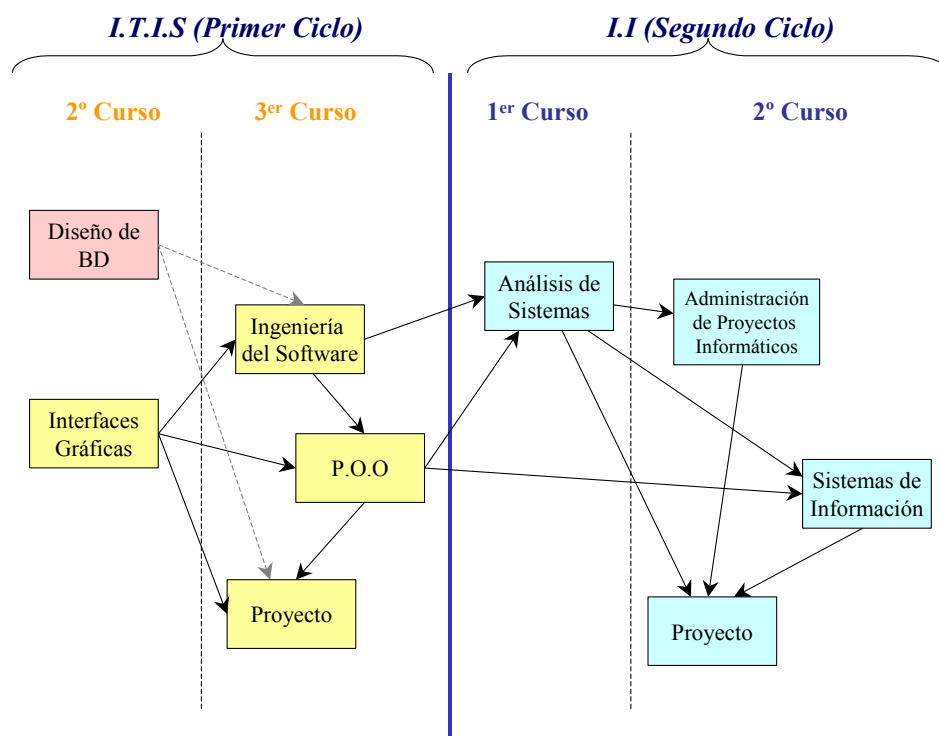


Figura 1. Relaciones de la Ingeniería del Software con otras asignaturas

2. Objetivos

2.1. Objetivos instrumentales generales

- OI1: Aplicar los conocimientos adquiridos mediante la resolución de problemas y una práctica obligatoria.
- OI2: Utilizar con fluidez herramientas CASE.
- OI3: Adquirir y emplear un buen lenguaje formal, tanto oral como escrito, siendo riguroso en las explicaciones de cualquier proceso.
- OI4: Conocer y utilizar la terminología utilizada en Ingeniería del Software.

- OI5: Adquirir un buen manejo de la bibliografía recomendada en la asignatura, de forma que se potencia la autosuficiencia a la hora de completar la formación.
- OI6: Comprender el ámbito de la Ingeniería del Software dentro de la Ingeniería Informática y dentro de los perfiles profesionales.
- OI7: Tomar conciencia de las implicaciones del trabajo de ingeniero de software, tanto individualmente como formando parte de un equipo
- OI8: Adquirir una visión inicial del campo de la Ingeniería del Software.
- OI9: Tener una perspectiva global del proceso asociado al ciclo de vida de un producto software.
- OI10: Conocer los problemas, principios, métodos y herramientas propios de la Ingeniería del Software.
- OI11: Conocer los elementos básicos de un modelo objeto, soportado en el lenguaje unificado de modelado (UML).
- OI12: Conocer la importancia y la influencia de los requisitos en el éxito de un proceso software.
- OI13: Aplicar las técnicas básicas de ingeniería de requisitos.
- OI14: Conocer y aplicar los principios de un método de análisis y diseño orientado a objetos.
- OI15: Conocer y aplicar los principios de un método de análisis y diseño estructurado.
- OI16: Adquirir la capacidad de crear documentaciones técnicas completas, correctas y legibles.

2.2. *Objetivos interpersonales generales*

- OIP1: Destrezas para la participación responsable: capacidad de coordinación, asistencia, contribuciones al grupo...
- OIP2: Capacidad de trabajar en equipo adquiriendo y mejorando las habilidades sociales y la inteligencia emocional.
- OIP3: Comprometerse de forma ética con el trabajo, con el resto de los integrantes del grupo y consigo mismo.

2.3. *Objetivos sistémicos generales*

- OS1: Capacidad de integrar los conocimientos y destrezas prácticas de las diferentes asignaturas del plan de estudio para resolver situaciones reales relacionadas con la Ingeniería Informática, así como con otras disciplinas relacionadas.
- OS2: Reforzar el hábito de plantearse interrogantes. Ante un problema preguntarse por el número de soluciones, la relación entre ellas, cómo afectaría a las condiciones iniciales alguna modificación...
- OS3: Capacidad de aplicar y relacionar, de forma autónoma, los contenidos de Ingeniería del Software de forma interdisciplinar.
- OS4: Adquirir una comprensión del método científico, a través de las diversas actividades realizadas en la asignatura de Ingeniería del Software, y

asimilar su importancia como manera de pensar y actuar en la labor de científico e ingeniero, fomentando la capacidad de abstracción y el espíritu crítico.

OS5: Desarrollar la madurez necesaria en el proceso de abstracción para abordar problemas reales y plantear modelos y soluciones de forma razonada y correcta.

OS6: Reforzar el hábito de desarrollar diferentes alternativas, cuestionando las características, riesgos y viabilidad de cada una, para cada problema planteado.

3. Competencias

3.1. Competencias instrumentales

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

3.1.1. Habilidades cognitivas

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

Generales

CIC1: Conocer y entender las relaciones que mantiene la asignatura con el resto de asignaturas de la titulación a partir de la introducción explícita de ejemplos y comentarios, aprovechando puntos relevantes del temario.

CIC2: Conocer y comprender algunas de las capacidades, aptitudes y conocimientos que la asignatura aporta para el desarrollo de los diversos perfiles profesionales.

CIC3: Ser capaz de crear documentaciones técnicas completas, correctas y legibles.

Unidad Didáctica I: Conceptos básicos

CIC4: Conocer el alcance de un producto software. Razonar acerca de la calidad del software.

CIC5: Entender el método de resolución de problemas que sustenta la Ingeniería del Software.

CIC6: Diferenciar las perspectivas propias del dominio de problema y del dominio de la solución.

CIC7: Entender el concepto de proceso software y conocer los principales modelos de proceso o paradigmas de ciclo de vida del software. Razonar acerca de su necesidad en la producción de software de calidad.

CIC8: Entender la noción de metodología y conocer los diferentes tipos existentes.

CIC9: Conocer la estructura de un proceso iterativo e incremental, que esté dirigido por casos de uso.

CIC10: Conocer el concepto y alcance de las herramientas CASE.

CIC11: Comprender los elementos que conforman un modelo objeto.

- CIC12: Conocer y comprender el lenguaje de modelado unificado (UML), con especial atención a las vistas estática y de interacción.
- CIC13: Ser capaz de aplicar las vistas estáticas y de interacción en modelos concretos y de complejidad baja y media, correspondientes a diversos artefactos software.

Unidad Didáctica II: Requisitos

- CIC14: Entender la importancia y el alcance de los requisitos en el ciclo de vida del software y en el éxito o fracaso de un proyecto software.
- CIC15: Comprender la implicación humana en la captura de requisitos. Razonar acerca de cómo influye esta implicación en la complejidad de los procesos relacionados con la obtención de requisitos.
- CIC16: Diferenciar los diferentes tipos de requisitos y las implicaciones de cada uno de ellos en el proceso software. Razonar acerca de la importancia de los requisitos no funcionales en la calidad del producto software.
- CIC17: Conocer las características que debe cumplir una especificación de requisitos.
- CIC18: Conocer y aplicar el proceso de obtención y documentación de requisitos.
- CIC19: Conocer y comprender la vista de casos de uso de UML.
- CIC20: Ser capaz de aplicar la vista de casos de uso en proyectos software de complejidad baja y media.

Unidad Didáctica III: Análisis y diseño orientado a objetos

- CIC21: Comprender la esencia de la etapa de análisis dentro de un proceso orientado a objetos.
- CIC22: Entender el alcance un modelo de dominio.
- CIC23: Conocer cómo identificar las clases conceptuales y las relaciones estructurales entre ellas propias de un modelo de dominio.
- CIC24: Ser capaz de aplicar todo el proceso de análisis orientado a objetos dentro de un proceso iterativo e incremental para realizar el modelo de análisis.
- CIC25: Ser capaz de diferenciar entre clases entidad, control e interfaz en un modelo de análisis.
- CIC26: Ser capaz de realizar una primera aproximación a la arquitectura del software con las clases fundamentales descubiertas en el análisis.
- CIC27: Conocer y comprender los principales principios y conceptos propios del diseño del software, con una mención especial a los principios de abstracción, refinamiento sucesivo, ocultación de la información y modularidad.
- CIC28: Conocer los principios fundamentales del diseño de la parte de interactiva de un sistema software.
- CIC29: Comprender la esencia de la etapa de diseño dentro de un proceso orientado a objetos.

- CIC30: Entender cómo se produce la transición desde la fase de análisis a la fase de diseño en un proceso iterativo, incremental y orientado a objetos.
- CIC31: Ser capaz de aplicar todo el proceso de diseño orientado a objetos dentro de un proceso iterativo e incremental para realizar el modelo de diseño.
- CIC32: Entender el concepto de diseño arquitectónico, tanto a un nivel lógico como de despliegue.
- CIC33: Comprender y aplicar un patrón de capas para dividir y organizar los subsistemas de diseño. Entender la necesidad de establecer interfaces claras, concisas y adecuadas para establecer las dependencias entre los subsistemas y facilitar la mantenibilidad del sistema completo.
- CIC34: Conocer y comprender algunos patrones de diseño orientado a objetos.

Unidad Didáctica IV: Análisis y diseño estructurado

- CIC35: Comprender la esencia de la etapa de análisis dentro de un proceso estructurado.
- CIC36: Comprender y aplicar los principios que permiten realizar un modelo funcional organizado en niveles. Conocer y comprender los componentes que conforman un diagrama de flujo de datos (DFD).
- CIC37: Conocer las extensiones del modelado funcional para los sistemas de tiempo real.
- CIC38: Entender la importancia de un diccionario de datos como parte del modelo funcional. Ser capaz de construir diccionarios de datos.
- CIC39: Entender la importancia de las técnicas de especificación de procesos como parte del modelo funcional. Ser capaz de construir especificaciones de procesos utilizando diferentes técnicas.
- CIC40: Profundizar en el entendimiento y realización de modelos conceptuales de información mediante el uso del diagrama entidad/relación extendido.
- CIC41: Conocer los principios del modelado del comportamiento a través de los diagramas de transición de estados.
- CIC42: Entender cómo mantener la consistencia entre los diferentes modelos realizados en el análisis estructurado.
- CIC43: Conocer y comprender los principios de un método de análisis estructurado moderno.
- CIC44: Comprender la esencia de la etapa de diseño dentro de un proceso estructurado.

3.1.2. Capacidades metodológicas

- CIM1: Ser capaz de tomar decisiones de manera razonada.
- CIM2: Tener capacidad de análisis y síntesis.
- CIM3: Ser capaz de manejar bibliografía relacionada con la Ingeniería del Software.
- CIM4: Ser capaz de documentar la solución a un problema dado en las diferentes fases del ciclo de vida con los diferentes niveles de abstracción y aproximación que esto requiere.

CIM5: Ser capaz de aplicar correctamente y manejar con comodidad los diferentes elementos de los que dispone un lenguaje de modelado concreto, ya sea orientado a objetos u orientado a procesos.

CIM6: Ser capaz de modelar un sistema de información desde las diferentes perspectivas que exige un determinado proceso.

3.1.3. Destrezas tecnológicas

CIT1: Habilidades básicas de navegación por la Web y uso del resto de servicios de red para la obtención y manejo de la información relacionada con la asignatura.

CIT2: Manejar las operaciones básicas y de administración de diferentes sistemas operativos para la instalación de las diferentes herramientas CASE a utilizar en la asignatura.

CIT3: Manejar con fluidez diferentes herramientas CASE.

CIT4: Manejo avanzado de un sistema de procesamiento de textos para la realización de los informes de las prácticas y la documentación de la práctica obligatoria.

3.1.4. Destrezas lingüísticas

CIL1: Adquirir y utilizar con fluidez un buen lenguaje científico, tanto oral como escrito, siendo riguroso en las explicaciones de cualquier proceso dentro de la asignatura.

CIL2: Conocer y utilizar la terminología usual de la Ingeniería del Software, tanto en español como en inglés.

3.2. Competencias interpersonales

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

3.2.1. Competencias para tareas colaborativas

CIPTC1: Ser capaz de realizar de trabajar en equipo para resolver los problemas de modelado enunciados en los talleres prácticos de la asignatura.

CIPTC2: Ser capaz de presentar en público una solución a un problema planteado y mantener un debate con el resto de la clase sobre la solución planteada, para así buscar colaborativamente la mejor solución al problema.

CIPTC3: Ser capaz de realizar y defender la práctica obligatoria de la asignatura en equipo.

3.2.2. Compromiso con el trabajo

CIPTR1: Se ha de definir un plan de trabajo en el que el volumen de trabajo de todos los miembros del equipo sea similar.

CIPTR2: Una vez finalizado el trabajo, todos los miembros del grupo deben conocer en profundidad todo el desarrollo realizado.

CIPTR3: Se debe cumplir el plazo de entrega de los trabajos.

CIPTR4: Se debe adquirir un compromiso ético entre todos los componentes del grupo.

3.3. Competencias sistémicas

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

- CS1: Capacidad de aplicar los conocimientos, métodos y herramientas vistos en la asignatura de Ingeniería del Software a situaciones y problemas concretos del área de la Ingeniería Informática y de otras disciplinas relacionadas.
- CS2: Capacidad de aprender y aplicar, de forma autónoma e interdisciplinar, nuevos conceptos y métodos relacionados con cada asignatura.
- CS3: Motivación por la calidad y por la creatividad.
- CS4: Capacidad de adoptar el proceso marcado por el método científico y de ingeniería en el planteamiento y realización de trabajos diversos, tanto a nivel académico como profesional.
- CS5: Capacidad de asimilación y adaptación a la evolución del estado del arte en el ámbito de la Ingeniería del Software y de la Ingeniería Informática como profesión.

4. Prerrequisitos

4.1. Competencias y contenidos mínimos

- Conocer los elementos básicos de un lenguaje de programación de 3ª generación.
- Conocer las principales formas de representar los algoritmos.
- Conocer el concepto de tipo abstracto de datos.
- Conocer los principios fundamentales del diseño de estructuras de datos.
- Conocer los principios fundamentales del modelado conceptual de datos mediante el diagrama entidad/relación.
- Conocer los mecanismos de transformación de un modelo conceptual de datos a un modelo lógico de datos.

4.2. Plan de trabajo para la consecución de los prerrequisitos

Los prerrequisitos para el estudio y entendimiento de la asignatura Ingeniería del Software se cubren fundamentalmente en las asignaturas de programación del primer curso, y en las asignaturas de Estructura de Datos y Bases de Datos del segundo curso.

5. Temario

5.1. Unidades didácticas

Unidad Didáctica I: Conceptos básicos

Tema 0: Sumario de la asignatura

Tema 1. Introducción a la Ingeniería del Software

Tema 2. Modelo Objeto. Una Descripción de UML

Unidad Didáctica II: Requisitos

Tema 3. Introducción a la Ingeniería de Requisitos

Unidad Didáctica III: Análisis y diseño orientado a objetos

Tema 4. Análisis Orientado a Objetos

Tema 5. Principios del Diseño del Software

Tema 6. Diseño Orientado a Objetos

Unidad Didáctica IV: Análisis y diseño estructurado

Tema 7. Análisis Estructurado

Tema 8. Diseño Estructurado

5.2. Desarrollo de las unidades didácticas

Unidad Didáctica I: Conceptos básicos

Tema 0: Sumario de la asignatura

0.1. Motivación

0.2. Enfoque de ingeniería en el desarrollo del software

0.3. Docencia de la Ingeniería del Software

0.4. Asignatura Ingeniería del Software en ITIS

0.4.1. Objetivos

0.4.2. Temario

0.4.3. Evaluación

0.4.4. Bibliografía básica recomendada

0.4.5. Otras fuentes

0.4.6. Enlaces

0.4.7. Herramientas CASE

0.4.8. Tutorías

Tema 1. Introducción a la Ingeniería del Software

1.1. Software

1.2. Conceptos básicos de la Ingeniería del Software

1.3. Proceso software

1.4. Modelos de proceso software

1.5. Metodologías

1.6. Proceso Unificado

1.7. CASE

Tema 2. Modelo Objeto. Una Descripción de UML

2.1. Introducción a la orientación a objetos

2.2. Modelo objeto

2.3. ¿Qué es UML?

2.4. Historia de UML

- 2.5. Visión global de UML
- 2.6. Vista estática
- 2.7. Vista de interacción
- 2.8. Vista de casos de uso
- 2.9. Vista de máquina de estados
- 2.10. Vista de actividad
- 2.11. Vistas físicas

Unidad Didáctica II: Requisitos

Tema 3. Introducción a la Ingeniería de Requisitos

- 3.1. Introducción
- 3.2. Ingeniería de requisitos
- 3.3. Requisitos
- 3.4. Especificación de requisitos del software
- 3.5. MDB: Una metodología de elicitación de requisitos
- 3.6. Vista de casos de uso en UML
- 3.7. Requisitos en el Proceso Unificado
- 3.8. Caso de estudio

Unidad Didáctica III: Análisis y diseño orientado a objetos

Tema 4. Análisis Orientado a Objetos

- 4.1. Introducción
- 4.2. Análisis orientado a objetos
- 4.3. Modelo del dominio
- 4.4. Análisis en el Proceso Unificado

Tema 5. Principios del Diseño del Software

- 5.1. Introducción
- 5.2. Proceso de diseño del software
- 5.3. Principios y conceptos del diseño del software

Tema 6. Diseño Orientado a Objetos

- 6.1. Introducción
- 6.2. Diseño en el Proceso Unificado
- 6.3. Diseño de la arquitectura
- 6.4. Patrones de diseño orientado a objetos

Unidad Didáctica IV: Análisis y diseño estructurado

Tema 7. Análisis Estructurado

- 7.1. Introducción

- 7.2. Modelado funcional
- 7.3. Modelado de información
- 7.4. Modelado de comportamiento
- 7.5. Balanceo de modelos
- 7.6. Método de análisis de Yourdon

Tema 8. Diseño Estructurado

- 8.1. Introducción
- 8.2. Diagrama de estructuras
- 8.3. Diseño arquitectónico
- 8.4. Diseño de datos

6. Metodología y estrategias de aprendizaje

6.1. Metodología docente

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

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

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

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

Para caminar en este sentido, el modelo educativo que se va a seguir en la asignatura de Ingeniería del Software tiene en la clase magistral un elemento importante, pero ya no exclusivo, en la transmisión de conocimiento. Este tipo de enseñanza se va a complementar con otros procesos entre los que cabe destacar las prácticas basadas en

enseñanza colaborativa y organizadas en talleres. Concretamente las actividades que se proponen son las siguientes:

- *Clases de teoría con apoyo de material audiovisual.* En estas clases se presentarán los contenidos básicos de un cierto tema. Las clases comenzarán con una breve introducción de los contenidos que se pretenden transmitir en la clase, así como con un breve comentario a los conceptos vistos en clases anteriores y que sirven de enlace a los que se pretenden desarrollar. El desarrollo de la clase se llevará a cabo con medios audiovisuales, textos, transparencias... que permitan un adecuado nivel de motivación e interés en los alumnos. Se debe intentar motivar a los alumnos a intervenir en cualquier momento en las clases para hacer éstas más dinámicas y facilitar el aprendizaje. Es importante intentar terminar la exposición con las conclusiones más relevantes del tema tratado.

Las transparencias que se utilizarán en clase son un subconjunto de las que se facilitan a los alumnos en la página web y en la fotocopidora. Estas transparencias son una guía para el estudio, pero no son sustitutas de la bibliografía recomendada. Todos los temas comparten una misma estructura que se compone de los siguientes ítems:

- *Portada:* Con el título del tema y fecha de última modificación.

Ingeniería del Software

Tema 1: Introducción a la Ingeniería del Software

Dr. Francisco José García Peñalvo
(fgarcia@usal.es)

3º I.T.I.S.
Fecha de última modificación: 5-10-2005

Universidad de Salamanca – Departamento de Informática y Automática

- *Resumen:* Con el resumen del tema, unos descriptores en forma de palabras clave y los capítulos de la bibliografía recomendada que deben consultar para ampliar/preparar el tema.

Ingeniería del Software
Introducción a la Ingeniería del Software

Resumen

Resumen	Se presentan los conceptos clásicos relacionados con el software y la Ingeniería del Software. El objetivo de este tema es tomar conciencia de la importancia de abordar la construcción del software desde una perspectiva de ingeniería. Se exponen los elementos constituyentes de un paradigma de desarrollo del software. Se ofrece una visión general del concepto de proceso, así como de los diferentes modelos de proceso software. Se introduce el concepto de metodología de desarrollo como contraposición al desarrollo anárquico y artesanal de aplicaciones, tan relacionado con la tan nombrada crisis del software. Se presenta, como ejemplo de proceso, el Proceso Unificado. Y se termina el tema hablando de herramientas CASE
Descriptores	Software; Aplicaciones del software; Crisis del software; Proceso software; Modelos de proceso; Ciclo de vida; Metodología; Método; Proceso Unificado; Herramienta CASE
Bibliografía	[Larman, 2003] Capítulo 2 [Plattini et al., 2004] Capítulos 3 y 4 [Pfleeger, 2002] Capítulos 1 y 2 [Pressman, 2002] Capítulos 1, 2 y 31 [Sommerville, 2002] Capítulos 1, 2 y 3

Universidad de Salamanca – Departamento de Informática y Automática © Dr. Francisco J. García Peñalvo 2

- *Esquema*: Con el índice del tema.

Ingeniería del Software
Introducción a la Ingeniería del Software

Esquema	
■ Software	
■ Conceptos básicos de la Ingeniería del Software	
■ Proceso software	
■ Modelos de proceso software	
■ Metodologías	
■ Proceso Unificado	
■ CASE	
■ Aportaciones principales del tema	
■ Ejercicios	
■ Lecturas complementarias	
■ Referencias	

Universidad de Salamanca – Departamento de Informática y Automática © Dr. Francisco J. García Peñalvo 3

- *Desarrollo del tema*: Con los apartados en los que se divide el tema.
- *Aportaciones principales del tema*: A forma de resumen del tema, incluye las aportaciones y conclusiones más importantes del tema.
- *Ejercicios*: Conjunto de enunciados con cuestiones y ejercicios sobre el tema, se incluyen también ejercicios resueltos.
- *Lecturas complementarias*: Lecturas opcionales para profundizar en el tema presentado.
- *Referencias*: Lista de todas las referencias que se citan en el desarrollo del tema.
- *Talleres de prácticas*. Las clases prácticas presenciales estarán dedicadas a la resolución colaborativa de problemas de modelado. El taller se organiza de la siguiente forma: para cada sesión de prácticas los alumnos organizados en grupos de trabajo de tres integrantes tendrán que resolver un problema de modelado cuyo enunciado encontrarán previamente en la web de la asignatura. Los talleres se organizan en sesiones de trabajo de dos horas, de las cuales los primeros quince minutos se reservan para que los integrantes del grupo cierren los últimos detalles. Un grupo voluntario expondrá la solución y posteriormente se llevará a cabo un debate, moderado por el profesor, con el resto de los grupos para discutir la solución, errores, alternativas... Finalmente, el grupo que defiende el trabajo realizará un informe final que se pondrá a disposición de todos en la web de la asignatura. Todos los grupos deben entregar su versión de la solución como mero control del trabajo de prácticas.
- *Práctica obligatoria*. Los grupos de prácticas deberán realizar una especificación de requisitos del software (completada con algunas partes de diseño) sobre un tema libre que ellos han debido de cerrar. Esa práctica tendrá que defenderse en grupo y representará una importante parte de la nota final de la asignatura.
- *Entrega de ejercicios*. Un alumno individualmente puede entregar ejercicios resueltos por él, cuyos enunciados debe encontrar en la bibliografía (sin que estén resueltos). Éstos serán discutidos con el profesor en horas de tutoría.
- *Tutorías*. El alumnado tiene a su disposición seis horas de tutorías a la semana en las que puede consultar cualquier duda relacionada con los

contenidos, organización y planificación de la asignatura. Las tutorías pueden ser individualizadas, pero se admite tutorías grupales para resolver problemas relacionados con las actividades a realizar en grupo.

- *Página web.* Se convierte en el vehículo de comunicación y registro de información de la asignatura.

6.2. Estrategias de aprendizaje

Se detalla las actuaciones concretas a realizar para la aplicación y aprovechamiento de la metodología propuesta. Se estructura en las siguientes fases:

6.2.1. Recopilación de la documentación de la asignatura

- Los alumnos dispondrán en la página web de la asignatura de toda la información y normativa relacionada con la asignatura: temario, criterios de evaluación, bibliografía, apuntes, enunciados de prácticas, trabajos, enlaces de interés, lecturas complementarias, avisos...
- El profesor mantiene actualizada la información de esta página para que se convierta en un vehículo de comunicación con los alumnos.
- Los alumnos deben conocer y manejar con fluidez toda la información y la documentación que se integra en la página de la asignatura.
- Asiduamente, el alumno debe consultar la página para estar actualizado.

6.2.2. Planificación de las clases teóricas

- Las clases teóricas se dedicarán a la presentación de contenidos y a la discusión sobre las dudas que surjan durante las exposiciones.
- Las clases se desarrollarán según el siguiente esquema:
 1. El profesor presenta los objetivos a conseguir y el contexto en el que se va a desarrollar la clase.
 2. Se presentan los conceptos.
 3. Se aclaran las dudas cuando éstas surjan. El profesor puede incentivar el debate con preguntas para hacer la clase más participativa.
 4. El profesor termina con un resumen de los principales conceptos tratados en la clase y puede introducir la siguiente clase.
- El alumno debe haber realizado una lectura previa de los contenidos que se van a tratar en la clase, máxime cuando no todos los contenidos se desarrollan en las clases magistrales.
- Una vez terminada la clase magistral, se debe estudiar de forma autónoma su contenido y en caso de no entender algo intentar aclararlo utilizando la bibliografía recomendada o cualquiera de los materiales adicionales recomendados en la asignatura.

6.2.3. Planificación de las clases prácticas

- El profesor publicará en la página de la asignatura los enunciados, con indicación de plazos de entrega, con suficiente antelación.
- En el caso de los talleres el grupo de trabajo debe reunirse con antelación para leer y comprender bien el enunciado y realizar su propuesta de solución.

- En el caso de la práctica obligatoria el grupo de trabajo debe leer y entender bien todos los puntos antes de ponerse a llevarla a cabo.
- Durante los talleres el profesor moderará el debate que se origine sobre el problema y la solución presentada, dando argumentos a cada una de las decisiones o alternativas que se planteen.
- Los alumnos deben cumplir el calendario de entrega de prácticas.
- Los informes de los talleres quedarán publicados en la web de la asignatura.
- Las prácticas obligatorias se defenderán en grupo.

6.2.4. Evaluación

- Se plantea una forma de evaluación continua.
- Para aquellos alumnos que no superen la evaluación continua o no se quieran o no puedan acogerse a ella existirá una prueba final.

7. Planificación del tiempo y del esfuerzo

La asignatura de Ingeniería del Software tiene en el plan de estudios actual 6 créditos LRU asignados, 4,5 créditos de teoría y 1,5 créditos de práctica. Para hacer su estudio en ECTS se ha tomado tomando la equivalencia de crédito LRU a crédito ECTS, con implicación de cambio de metodología docente.

Al hacer el estudio de carga de trabajo máxima, se tiene que:

- 6 ECTS -> 150-180 horas de trabajo
- 4,5cr LRU -> 45 horas presenciales + (45*1,5) horas de asimilación = 112,5 horas
- 1,5cr LRU -> 15 horas presenciales + (15*2) horas de asimilación = 45 horas
- 112,5 + 45 = 157,5 horas de trabajo
- $150 < 157,5 < 180$

Dos son los escenarios más probables: el resultante de tomar el crédito ECTS como 30 horas de trabajo y el resultante de tomarlo como 25 horas de trabajo.

Por temas de organización y calendario, se ha optado por partir del escenario que asigna 25 horas de trabajo al ECTS.

Para poder elaborar las guías, cuyo destino final sería la realización de experiencias piloto, dentro de los planes de estudio vigentes, se debe tener en cuenta la definición del crédito del R.D. 1497/1987 y sus posteriores modificaciones. Así:

- La unidad de valoración de las enseñanzas se corresponde a diez horas de enseñanza teórica, práctica o de sus equivalencias entre las que podrán incluirse actividades académicas dirigidas, que habrán de preverse en el correspondiente plan docente junto con los mecanismos y medios objetivos de comprobación de los resultados académicos de las mismas. Todo ello sin perjuicio del cumplimiento del régimen de dedicación del profesorado, de conformidad con el Real Decreto 898/1985, de 30 de abril sobre régimen del profesorado universitario.

- En ningún caso, salvo que se trate de enseñanzas en Universidades a distancia, el porcentaje del crédito correspondiente a las actividades académicas dirigidas será superior al **30%**.

De esta forma, si 6 créditos LRU equivalen a 60 horas, se debe asegurar el 70% como mínimo de presencialidad, esto es, 42 horas entre horas de teoría y de práctica.

En la Tabla 1 se presenta la organización del esfuerzo del alumno para cubrir los 6 ECTS asociados a la asignatura. Por su parte en la Tabla 2 se distribuyen las horas presenciales entre los diferentes temas y el examen.

			A	B	C	D	E
	Técnica	Actividad	Horas equivalentes de clase	Factor de trabajo del alumno [†]	Horas de trabajo personal del alumno	Horas totales (A+C)	ECTS (D ÷ 25)
Teoría	Seguimiento de la parte teórica de las unidades didácticas	Asimila contenidos. Se plantea dudas que planteará a los profesores en las tutorías	34	1,5	51	85	3,4
Práctica	Seguimiento de la parte práctica de las unidades didácticas	Experimenta, practica, modela, simula	8	2	16	24	0,96
Ejercicios de apoyo	Ejercicios de repaso y afianzamiento de lo aprendido. Pueden computar o no para la superación de la asignatura, dependiendo de la tipología de éstos	Realiza, resuelve problemas, tests...	-	-	5	5	0,2
Práctica obligatoria	Práctica en grupo de obligada realización	Realiza una ERS completada con elementos de diseño	-	-	30	30	1,2
Examen	Controles de evaluación continua y/o examen final de la asignatura	Exámenes tipo test, supuestos prácticos...	3			3	0,12
Otras actividades	Tutorías personalizadas y grupales	Recibe orientación personalizada	-	-	2	2	0,08
	Búsquedas en la red, participación en foros especializados...	Busca elementos para completar los contenidos	-	-	1	1	0,04
TOTAL			45		105	150	6

Tabla 1. Escenario ECTS = 25 horas de trabajo

[†] Número de horas dedicadas por el alumno al trabajo personal (organización de apuntes, estudio, documentación...) por cada hora de clase.

NÚMERO DE HORAS PRESENCIALES		
Actividad	Horas Teoría	Horas Prácticas
Unidad Didáctica I: Conceptos básicos		
Tema 0	1	
Tema 1	6	
Tema 2	7	
Unidad Didáctica II: Requisitos		
Tema 3	5	2
Unidad Didáctica III: Análisis y diseño orientado a objetos		
Tema 4	3	4
Tema 5	2	
Tema 6	6	
Unidad Didáctica IV: Análisis y diseño estructurado		
Tema 7	3,5	2
Tema 8	0,5	
Examen	3	
TOTAL: 45	37	8

Tabla 2. Reparto de horas presenciales

8. Bibliografía

8.1. Bibliografía básica

- Booch, G., Rumbaugh, J., Jacobson, I. (1999) *El Lenguaje Unificado de Modelado*. Addison Wesley.
- Jacobson, I., Booch, G., Rumbaugh, J. (2000) *El Proceso Unificado de Desarrollo de Software*. Addison-Wesley.
- Larman, C. (2003) *UML y Patrones*. 2ª Edición. Prentice-Hall.
- Pfleeger, S. L. (2002) *Ingeniería del Software*. Teoría y Práctica. Prentice Hall.
- Piattini, M. G., Calvo-Manzano, J. A., Cervera, J., Fernández, L. (2004) *Análisis y Diseño de Aplicaciones Informáticas de Gestión. Una perspectiva de Ingeniería del Software*. Ra-ma.
- Pressman, R. S. (2005) *Ingeniería del Software: Un Enfoque Práctico*. 6ª Edición. McGraw-Hill.
- Rumbaugh, J., Jacobson, I., Booch, G. (2000) *El Lenguaje Unificado de Modelado. Manual de Referencia*. Addison-Wesley.
- Sommerville, I. (2005) *Ingeniería del Software*. 7ª Edición, Pearson Addison-Wesley.
- Yourdon, E. (1993) *Análisis Estructurado Moderno*. Prentice-Hall Hispanoamericana.

8.2. Bibliografía complementaria

- Gamma, E., Helm, R., Johnson, R., Vlissides, J. (2003) *Patrones de Diseño*. Addison-Wesley.

Larman, C. (2005) *Applying UML and Patterns. An Introduction to Object-Oriented Analysis and Design and Iterative Development*. 3rd Edition. Prentice Hall.

Meyer, B. (1999) *Construcción de Software Orientado a Objetos*. 2^a Edición. Prentice Hall.

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

8.3. Otros recursos

- Durán, A., Bernárdez, B. (2002) Metodología para la Elicitación de Requisitos de Sistemas Software (versión 2.3). Informe Técnico LSI-2000-10, Universidad de Sevilla. <http://www.lsi.us.es/~amador>. [Última vez visitado, 22-9-2005].
- Durán, A., Bernárdez, B. (2001) Metodología para el Análisis de Requisitos de Sistemas Software (versión 2.2). Universidad de Sevilla. <http://www.lsi.us.es/~amador>. [Última vez visitado, 22-9-2005].
- OMG (2003) OMG Unified Modeling Language Specification. Version 1.5. Object Management Group Inc. Document formal/03-03-01. March 2003. <http://www.omg.org/docs/formal/03-03-01.pdf>. [Última vez visitado, 22-9-2005]
- OMG (2005) Unified Modeling Language: Superstructure. Version 2.0. Object Management Group Inc. Document formal/05-07-04. August 2005. <http://www.omg.org/cgi-bin/doc?formal/05-07-04>. [Última vez visitado, 22-9-2005].
- Pohl, K. (1997) Requirements Engineering: An Overview. En M. Dekker (Ed.), *Encyclopedia of Computer Science and Technology*, 36. Disponible en <ftp://sunsite.informatik.rwth-aachen.de/pub/CREWS/CREWS-96-02.pdf>. [Última vez visitado, 22-9-2005].

8.4. Enlaces de interés

- **Association for Computing Machinery (ACM)**
 - <http://www.acm.org>.
 - Fundada en 1947 fue la primera sociedad científica y de educación del mundo. El portal de información que presenta es impresionante, tanto en cuanto a enlaces de interés, grupos de trabajo, documentos electrónicos, conferencias como por su biblioteca digital conteniendo revistas y actas de congresos (<http://portal.acm.org>).
- **Cetus Links - Object-Oriented**
 - <http://www.cetus-links.org>.
 - La colección de enlaces más completa sobre Orientación a Objetos. A fecha de 17 de marzo de 2004 se tenían contabilizados 18193 enlaces.
- **Consejo Superior de Informática**
 - <http://www.csi.map.es/>.
 - Web del Consejo Superior de Informática donde, entre otras cosas, se puede obtener la documentación y herramientas sobre Métrica 3 (<http://www.csi.map.es/csi/metrica3/index.html>).
- **DACS (Data & Analysis Center for Software)**
 - <http://www.dacs.dtic.mil>.
 - Sitio perteneciente al DoD de EEUU con numerosa información y enlaces a lugares relacionados con la Ingeniería del Software.
- **European Software Institute (ESI)**
 - <http://www.esi.es>.

- El Instituto del Software Europeo tiene su sede en Bilbao (España). Dispone tanto de documentación privada para los miembros del instituto como documentación pública con los análisis de proyectos, necesidades de empresas y software europeo.
- **Guide to the Software Engineering Body of Knowledge (SWEBOK)**
 - <http://www.swebok.org>.
 - Proyecto para establecer un cuerpo de conocimiento común para la Ingeniería del Software.
- **Institute of Electrical and Electronics Engineers (IEEE)**
 - <http://www.ieee.org>.
 - Otra prestigiosa organización compuesta por diversas sociedades, donde la que más relación tiene con los temas abordados en el presente curso es la IEEE Computer Society (<http://computer.org>).
 - A semejanza de ACM, ofrece información sobre conferencias, estándares, educación y mantiene otra biblioteca digital con revistas y actas de congresos.
- **Object Management Group (OMG)**
 - <http://www.omg.org>.
 - Es un consorcio internacional de industrias con el fin de promover el uso de la Orientación a Objetos en la Ingeniería del Software. A diferencia de organizaciones como ISO o IEEE, OMG desarrolla estándares de “*facto*” como consenso entre las empresas que la forman.
 - Dicho servidor ofrece publicaciones electrónicas y enlaces a estándares y herramientas del sector relacionado con la tecnología de objetos.
 - En <http://www.omg.org/uml> se encuentran las especificaciones de la versión actual (así como de las anteriores) de UML.
- **IBM Rational Software**
 - <http://www-306.ibm.com/software/rational/>.
 - En esta dirección se tiene valiosa información sobre UML y RUP (documentos, informes, artículos, presentaciones, bibliografía recomendada...).
 - Además, se pueden obtener versiones de demostración de diferentes herramientas que comercializan, siendo Rational Rose la más difundida.
- **R. S. Pressman & Associates, Inc.**
 - <http://www.rspa.com>.
 - Bajo la dirección de Roger S. Pressman y la difusión internacional de su afamado libro sobre Ingeniería del Software, aparece una empresa de consultoría en Ingeniería del Software.
 - Lo más interesante que ofrece esta dirección es un portal que da entrada a otras fuentes de información relacionadas con cada uno de los capítulos tratados en su libro.

- **Software Engineering Institute (SEI)**
 - <http://www.sei.cmu.edu>.
 - El Instituto de Ingeniería del Software en la Universidad Carnegie Mellon, es uno de los lugares más activos en pro de la Ingeniería del Software.
 - Se pueden encontrar documentos asociados a módulos curriculares en Ingeniería del Software, informes técnicos sobre diferentes áreas de la Ingeniería del Software, una revista en línea...
 - Incluye además enlaces a otras organizaciones relacionadas con la Ingeniería del Software.
- **The World Wide Web Consortium**
 - <http://www.w3.org/>.
 - Sitio central donde se recogen las especificaciones de los diversos lenguajes relacionados con la Web (HTML, XML, RDF, SOAP...).
- **The World Wide Web Virtual Library Formal Methods**
 - <http://www.afm.sbu.ac.uk>.
 - Sitio especializado en información sobre métodos formales.
- **UML Resource Center**
 - <http://www-306.ibm.com/software/rational/uml/>.
 - Sitio donde se centraliza gran parte de la información sobre UML.
- **Unified Modeling Language (UML) Dictionary**
 - <http://usecasedriven.com/UML.htm>.
 - Diccionario sobre términos de UML.
- **Webliography Software Engineering**
 - http://polaris.umuc.edu/~skerby/help/wbib_swe.htm.
 - Colección de enlaces sobre Ingeniería del Software.
- **WWW Virtual Library - Software Engineering**
 - <http://mingo.info-science.uiowa.edu/soft-eng>.
 - Colección de enlaces sobre Ingeniería del Software.

8.5. Herramientas CASE

- **ArgoUML**
 - <http://argouml.tigris.org>.
- **Enterprise Architect**
 - <http://www.deiser.com>.
- **Left CASE**
 - <http://zarza.fis.usal.es>.

- **Microsoft Visio**
 - <http://www.microsoft.com/office/visio>.
- **PowerDesigner**
 - <http://www.sybase.com/products/developmentintegration/powerdesigner>.
- **Rational Rose**
 - <http://www-306.ibm.com/software/rational/>.
- **REM**
 - http://www.lsi.us.es/descargas/descarga_programas.php?id=3.
- **Studio CASE**
 - <http://zarza.fis.usal.es/~fgarcia>.
- **Visual Paradigm**
 - <http://www.visual-paradigm.com/>.
- **Visual UML**
 - <http://www.visualobjectmodelers.com/products.htm>.

9. Evaluación de los procesos y resultados de aprendizaje

9.1. Sistema de evaluación

La evaluación de esta asignatura se divide en dos partes diferenciadas:

- **Parte de Teoría** (50% de la nota final).
 - Un examen final que consta de dos partes que hay que aprobar independientemente:
 - Un test.
 - Un conjunto de supuestos teórico/prácticos.
 - La parte de teoría se guardará durante el curso académico actual.
- **Parte Práctica** (50% de la nota final).
 - Práctica realizada en grupos de trabajo.
 - Se realizará una defensa de dicho trabajo.
 - Todo grupo tiene derecho a dos defensas, siempre que la primera se haya hecho antes del mes de mayo.
 - La parte práctica se guardará durante el curso académico actual.

De forma complementaria a este sistema de evaluación tradicional, el alumno se puede acoger opcionalmente a una evaluación continua.

- Se tendrá en cuenta la asistencia y la participación activa en clase.
- Todos los alumnos que se acojan a esta modalidad deberán haber entregado su ficha en el primer mes de clase.
- En relación con la parte de test del examen de teoría:
 - Se realizarán 3 pruebas de test durante las clases de teoría:

- Prueba 1: Unidad I
- Prueba 2: Unidades II y III
- Prueba 3: Unidad IV
- Se eliminará la parte de test del examen de teoría si se obtiene una calificación superior o igual a 5 en el conjunto de estas pruebas, obteniendo en todas ellas al menos una nota de 3:
 - $\text{Nota test} = (\text{Prueba 1} * 0,3) + (\text{Prueba 2} * 0,5) + (\text{Prueba 3} * 0,2)$
- En relación con la parte del conjunto de supuestos teórico/prácticos del examen de teoría:
 - En los talleres
 - Todos los grupos obligatoriamente entregan el modelo realizado al final de la sesión (impreso y realizado con alguna herramienta de modelado).
 - Si se detecta fraude se resta 1 punto.
 - Grupos voluntarios defienden los trabajos y realizan informes con entrega en 15 días.
 - Por la defensa se obtiene entre 0 y 0,5 puntos, por entregar el informe 0,5 puntos.
 - Si no hay grupos voluntarios, se elegirá un grupo que defienda el trabajo.
 - Por la defensa se obtiene entre -1 y 0,5 puntos, la entrega del informe es obligatoria y no aporta nota (pero puede restar si no se entrega en condiciones).
 - La participación activa, acertada y continuada puede aportar hasta 0,5 puntos.
 - Entrega de ejercicios
 - Entregar 3 ejercicios de modelado, con enunciados sacados de libros (y no resueltos en ellos), tras su revisión en horario de tutorías aporta 0,75 puntos

La nota final de la asignatura se calculará conforme a la siguiente fórmula:

Si (Teoría \geq 5,0) y (Práctica \geq 5,0)

Nota Final = (Teoría*0,5) + (Práctica*0,5)

Si no



Fin si

9.2. Criterios de evaluación

La calificación se hará conforme a las siguientes pautas:

Matrícula de honor

- La nota final es superior o igual a 9,75 puntos

- El conocimiento y la comprensión de la materia se extienden más allá del trabajo cubierto por el programa.
- La comprensión conceptual es sobresaliente.
- Los problemas relacionados con la asignatura son resueltos con soltura y se es capaz de razonar alternativas de solución estableciendo comparativas entre ellas.
- La participación en las diferentes actividades, incluyendo las clases, sobresale por su corrección y satisfacción.

Sobresaliente

- La nota final es superior o igual a 8,5 puntos
- El conocimiento y la comprensión de la materia son muy satisfactorios.
- La comprensión conceptual es sobresaliente.
- Los problemas relacionados con la asignatura son resueltos con soltura y se es capaz de razonar alternativas de solución.
- La participación en las diferentes actividades, incluyendo las clases, ha sido muy correcta y muy satisfactoria.

Notable

- La nota final es superior o igual a 6,75 puntos
- El conocimiento y la comprensión de la materia son satisfactorios.
- La comprensión conceptual es notable.
- Los problemas relacionados con la asignatura son resueltos con soltura.
- La participación en las diferentes actividades, incluyendo las clases, ha sido correcta y bastante satisfactoria.

Aprobado

- La nota final es superior o igual a 5 puntos
- El conocimiento y la comprensión de la materia son básicos.
- La comprensión conceptual es suficiente.
- Los problemas relacionados con la asignatura son resueltos con éxito razonable.
- La participación en las diferentes actividades, incluyendo las clases, ha sido correcta y pero no siempre satisfactoria.

Suspenso

- La nota final es inferior a 5 puntos
- El conocimiento y la comprensión de la materia no son aceptables.
- La comprensión conceptual es insuficiente.
- Los problemas relacionados con la asignatura no se resuelven de forma adecuada.

- La participación en las diferentes actividades, incluyendo las clases, ha sido escasa y deficiente.

10. Análisis de coherencia de la guía docente

Este análisis permite condensar las diferentes relaciones existentes entre los objetivos y competencias a desarrollar con el plan de trabajo del alumnado y el sistema de evaluación empleado. El análisis se ha estructurado en tres tablas separadas en función del tipo de objetivo y competencia analizados en cada caso: instrumental, interpersonal y sistémico.

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS		BLOQUES DE CONTENIDOS				PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Instrumentales		Unidad I	Unidad II	Unidad III	Unidad IV		
O11 O12 O16	CIC3 CIC5		2	3	4,6	7	Enseñanza presencial (Talleres de prácticas) Enseñanza no presencial (Trabajo en grupos/Realización de la práctica obligatoria) Tutorías	Procedimientos Talleres prácticos Práctica obligatoria Criterios Grado de comprensión y aplicación de los conceptos de la Ingeniería del Software
O13 O14 O15	CIC1 CIC2		0,1,2	3	4,5,6	7,8	Enseñanza presencial (Clases magistrales/Talleres de prácticas/Defensa de práctica/Exposición de ejercicios) Enseñanza no presencial (Trabajo en grupos/Realización de la práctica obligatoria/Revisión bibliográfica/Consulta de recursos en Internet) Tutorías	Procedimientos Defensas de prácticas en grupo Criterios Grado de comprensión y aplicación de los conceptos de la Ingeniería del Software Evaluación del trabajo en equipo
O16 O17 O18	CIC4 CIC5 CIC6		0,1				Enseñanza presencial (Clases magistrales/Talleres de prácticas) Enseñanza no presencial (Trabajo en grupos/Realización de la práctica obligatoria/Revisión bibliográfica/Consulta de recursos en Internet) Tutorías	Procedimientos Tutorías Conferencias invitadas Criterios Detectar el conocimiento y visión que tienen del mercado laboral
O19	CIC7 CIC8		1				Enseñanza presencial (Clases magistrales/Talleres de prácticas/Defensa de práctica) Enseñanza no presencial	Procedimientos Examen Práctica obligatoria

						(Trabajo en grupos/Realización de la práctica obligatoria/Revisión bibliográfica/Consulta de recursos en Internet) Tutorías	Criterios Grado de comprensión y aplicación del concepto de proceso software
OI10	CIC8 CIC10	1,2	3	4,6	7,8	Enseñanza presencial (Clases magistrales/Talleres de prácticas/Defensa de práctica/Exposición de ejercicios) Enseñanza no presencial (Trabajo en grupos/Realización de la práctica obligatoria/ Realización de ejercicios/Revisión bibliográfica/Consulta de recursos en Internet) Tutorías	Procedimientos Ejercicios Examen Talleres Práctica obligatoria Criterios Grado de comprensión y aplicación de las técnicas y herramientas de la Ingeniería del Software
OI11	CIC11-CIC13	2				Enseñanza presencial (Clases magistrales/Talleres de prácticas/Defensa de práctica/Exposición de ejercicios) Enseñanza no presencial (Trabajo en grupos/Realización de la práctica obligatoria/Realización de ejercicios/Revisión bibliográfica/Consulta de recursos en Internet) Tutorías	Procedimientos Ejercicios Examen Talleres Práctica obligatoria Criterios Grado de comprensión y aplicación del modelo objeto
OI12 OI13	CIC14-CIC20		3			Enseñanza presencial (Clases magistrales/Talleres de prácticas/Defensa de práctica/Exposición de ejercicios) Enseñanza no presencial (Trabajo en grupos/Realización de la práctica obligatoria/Realización de ejercicios/Revisión bibliográfica/Consulta de recursos en Internet) Tutorías	Procedimientos Ejercicios Examen Talleres Práctica obligatoria Criterios Grado de comprensión y aplicación del concepto de requisito
OI14	CIC21-CIC34			4,6		Enseñanza presencial (Clases magistrales/Talleres de prácticas/Defensa de práctica/Exposición de ejercicios) Enseñanza no presencial (Trabajo en grupos/Realización de la práctica obligatoria/Realización de ejercicios/Revisión bibliográfica/Consulta de recursos en Internet) Tutorías	Procedimientos Ejercicios Examen Talleres Práctica obligatoria Criterios Grado de comprensión y aplicación de un método de ADOO

OI15	CIC35-CIC44				7,8	<p>Enseñanza presencial (Clases magistrales/Talleres de prácticas/Exposición de ejercicios)</p> <p>Enseñanza no presencial (Trabajo en grupos/Realización de ejercicios/Revisión bibliográfica/Consulta de recursos en Internet)</p> <p>Tutorías</p>	<p>Procedimientos</p> <p>Ejercicios Examen Talleres Práctica obligatoria</p> <p>Criterios</p> <p>Grado de comprensión y aplicación de un método estructurado</p>
Elementos transversales	CIM1-CIM6 CIT1-CIT4 CIL1-CIL2	0,1,2	3	4,5,6	7,8	<p>Enseñanza presencial (Clases magistrales/Talleres de prácticas/Defensa de práctica/Exposición de ejercicios)</p> <p>Enseñanza no presencial (Trabajo en grupos/Realización de la práctica obligatoria/Realización de ejercicios/Revisión bibliográfica/Consulta de recursos en Internet)</p> <p>Tutorías</p>	<p>Procedimientos</p> <p>Ejercicios Examen Talleres Práctica obligatoria</p> <p>Criterios</p> <p>Grado de destreza en la competencias transversales</p>

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS		BLOQUES DE CONTENIDOS				PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Interpersonales		Unidad I	Unidad II	Unidad III	Unidad IV		
OIP1	CIPTC1		2	3	4,6	7	Enseñanza presencial (Talleres de prácticas/Exposición de ejercicios)	Procedimientos Actividades en grupo Criterios Grado de destreza en trabajos en grupo
OIP2	CIPTC2 CIPTC3		2	3	4,6	7	Enseñanza presencial (Talleres de prácticas/Defensa de práctica/Exposición de ejercicios)	Procedimientos Actividades en grupo Criterios Grado de destreza en trabajos en grupo
OIP3	CIPTR1-CIPTR4		2	3	4,6	7	Enseñanza presencial (Talleres de prácticas/Defensa de práctica/Exposición de ejercicios)	Procedimientos Actividades en grupo Criterios Grado de destreza en trabajos en grupo

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS		BLOQUES DE CONTENIDOS				PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Sistémicas		Unidad I	Unidad II	Unidad III	Unidad IV		
OS1	CS1		0,1,2	3	4,5,6	7,8	Enseñanza presencial (Clases magistrales/Talleres de prácticas/Defensa de práctica/Exposición de ejercicios) Enseñanza no presencial (Trabajo en grupos/Realización de la práctica obligatoria/Realización de ejercicios/Revisión bibliográfica/Consulta de recursos en Internet) Tutorías	Procedimientos Ejercicios Examen Talleres Práctica obligatoria Trabajos y prácticas de otras asignaturas Proyecto fin de carrera Criterios Nivel de actuación en destrezas transferibles
OS2 OS3 OS5	CS2 CS5		0,1,2	3	4,5,6	7,8	Enseñanza presencial (Talleres de prácticas/Defensa de práctica/Exposición de ejercicios) Enseñanza no presencial (Realización de ejercicios/Revisión)	Procedimientos Prácticas Trabajos en grupo Ejercicios

						bibliográfica/Consulta de recursos en Internet) Tutorías	Criterios Nivel de precisión y eficiencia relacionados con la resolución de problemas nuevos
OS4 OS6	CS3 CS4	1,2	3	4,6	7,8	Enseñanza presencial (Talleres de prácticas/Defensa de práctica/Exposición de ejercicios) Enseñanza no presencial (Realización de ejercicios/Revisión bibliográfica/Consulta de recursos en Internet) Tutorías	Procedimientos Prácticas Trabajos en grupo Ejercicios Criterios Grado de análisis y evaluación de los procedimientos relacionados con la resolución de problemas

Guía Docente de Procesadores de Lenguajes

Luis Alonso Romero

Vivian López Batista

Departamento de Informática y Automática

Facultad de Ciencias – Universidad de Salamanca

Plaza de los Caídos S/N, 37008, Salamanca, España

Tfno. +34 923 294400. Ext. 1513

Fax. +34 923 294514

lalonso@usal.es , vivian@usal.es

1. Contexto

1.1. Perfil de los créditos y su adecuación al perfil profesional y académico de la titulación

El objetivo principal de esta asignatura, troncal, es estudiar los principios de diseño e implementación de los procesadores de lenguajes: programas que traducen ficheros escritos en un lenguaje de alto nivel a ficheros que contienen código máquina casi-ejecutable. Cualquier profesional informático va a hacer un uso continuo de compiladores e intérpretes durante su vida. Por lo tanto, un conocimiento acerca del funcionamiento interno de estas herramientas cotidianas parece muy conveniente. Pero, además, de su estudio se van a derivar una serie de efectos secundarios, algunos de enorme utilidad y que encuentran aplicación fuera del campo de los procesadores de lenguajes. Entre ellos, podemos citar los siguientes:

- Desarrollo de interfaces textuales, para interpretar comandos o desarrollar técnicas de interacción hombre-máquina.
- Tratamiento de ficheros de texto con información estructurada . Lenguajes y herramientas como PERL, TCL, sed, fgrep, grep, etc incorporan expresiones regulares para detectar y actuar sobre textos.
- Procesadores de texto que incorporan expresiones regulares (vi, emacs,..)
- Diseño e implementación de lenguajes “orientados a página” : HTML, postscript, tex, .
- Procesamiento de lenguaje natural, en el que las primeras fases del tratamiento son la de análisis léxico y sintáctico.
- Reconocimiento sintáctico de patrones, en el cual los diferentes patrones pueden caracterizarse por relaciones entre sus atributos, que pueden ser descritas por una gramática. Este tipo de tratamiento es muy empleado en visión artificial.

Puede parecer una tontería estudiar cómo se construye un producto de software que es muy probable que nunca tengamos que construir. Podemos mencionar un cierto número de razones para refutar esta afirmación:

1. La construcción de compiladores es una rama de la ciencia de la computación que ha tenido mucho éxito y una de las primeras en ganarse el título de “ciencia”.

2. Dado que hay un gran manejo de ficheros dentro de un compilador, algunas de las herramientas que se han desarrollado (lex, yacc) tienen aplicaciones muy interesantes fuera del campo de los procesadores de lenguajes: generadores de páginas HTML, Postscript, etc.
3. Hay algunos algoritmos muy útiles para su uso en otros campos: “hash”, recolección de basura, grafos, ...

En el caso 1, las razones del éxito se derivan de tres razones:

Una adecuada estructuración del problema. Los compiladores analizan la entrada, construyen una representación semántica y sintetizan la salida. Este paradigma análisis-síntesis es muy potente, y muy aplicable a otras ramas. Sin esta separación entre las fases de análisis y síntesis, los compiladores (y los lenguajes) no estarían donde están ahora. Pensemos que el primer compilador real, Fortran II, supuso un trabajo equivalente a 18 años-hombre. Y eso que el Fortran II era un lenguaje mínimo en comparación a los actuales. Sin esta metodología, cada nuevo lenguaje requeriría un compilador diferente para cada máquina. Si tenemos L lenguajes y M máquinas diferentes eso nos llevaría a $L \cdot M$ compiladores diferentes. Con una metodología de análisis-síntesis, la fase de análisis hasta encontrar la representación semántica intermedia sería siempre la misma, y solamente habría que construir la fase de síntesis. Esto nos conduce a $L+M$ compiladores distintos (Grune, “Modern Compiler Design”):

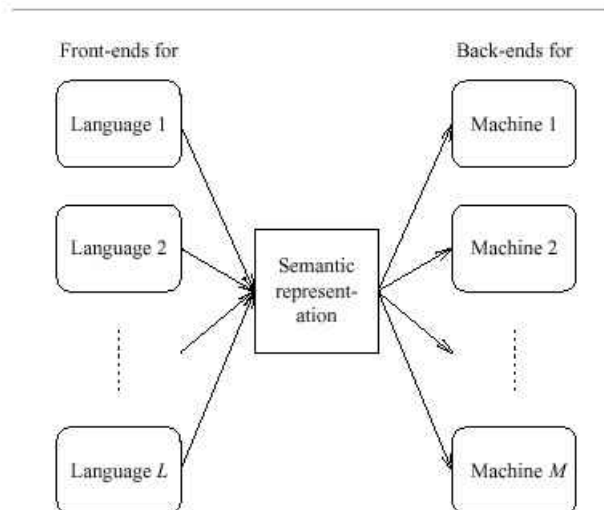


Figure 1.4 Creating compilers for L languages and M machines.

Un empleo juicioso de los formalismos. Algunas partes del compilador tienen excelentes formalismos ya desarrollados. Tal es el caso de las expresiones regulares y la gramáticas libres de contexto, cuyos formalismos fueron desarrollados desde 1960 hasta 1990, aproximadamente. Por otra parte, existen formalismos que están todavía en fase de desarrollo y su uso todavía no se ha generalizado. Tal es el caso de las gramáticas atribuidas para el análisis semántico, la correspondencia de patrones y la programación dinámica para la generación y optimización de código, etc.

Uso de herramientas de generación de programas. Una vez que se tiene un formalismo adecuado en el que se pueda describir las tareas de un programa, se puede escribir un programa para generar el programa. Como ejemplos podemos citar los analizadores de léxico generados a partir de expresiones regulares (lex, jlex), los analizadores sintácticos a partir de especificaciones gramaticales (yacc, bison, cup), etc. Esta

herramientas facilitan enormemente la construcción de los compiladores y permiten dedicar el esfuerzo a tareas de mayor nivel de abstracción.

1.2. Ubicación y relaciones en el plan de estudios

La asignatura de Procesadores de Lenguajes se imparte en el segundo cuatrimestre del primer curso del Segundo Ciclo de la Ingeniería Informática, cuyo Plan de Estudios actual data de 1998. Como tal, no existe una relación muy directa con las demás asignaturas del Plan de Estudios, salvo las de contenido muy genérico relacionadas con la Ingeniería del Software (aparte, naturalmente, del Proyecto Fin de Carrera).

En esta asignatura se ponen en juego, desde un punto de vista aplicado, los conocimientos adquiridos a lo largo de la carrera. Proporciona un buen ejercicio de síntesis sobre la teoría, los métodos y las técnicas aprendidas en las asignaturas de programación, estructuras de datos y algoritmos y, sobre todo, teoría de autómatas y lenguajes formales. Es una asignatura imprescindible para completar una formación básica en informática

1.3.- Pre-requisitos.-

Aparte de un buen conocimiento de algún lenguaje de programación (C, Java), al alumno se le suponen los conocimientos básicos sobre teoría de lenguajes recibidos en la asignatura “Teoría de Autómatas y Lenguajes Formales”, de la Ingeniería Técnica o del Primer Ciclo. No obstante, en el capítulo 2 de la asignatura se hace un repaso de los principales conceptos descritos.

2. Objetivos

2.1. Objetivos instrumentales generales

- OI1: Aplicar los conocimientos adquiridos mediante una práctica obligatoria realizada en grupo de dos alumnos.
- OI2: Utilizar con fluidez herramientas de construcción de analizadores léxicos y sintácticos..
- OI3: Adquirir y emplear un buen lenguaje formal, tanto oral como escrito, siendo riguroso en las explicaciones de cualquier proceso.
- OI4: Conocer y utilizar la terminología utilizada en Compiladores.
- OI5: Adquirir un buen manejo de la bibliografía recomendada en la asignatura, de forma que se potencia la autosuficiencia a la hora de completar la formación.
- OI6: Comprender el ámbito de los procesadores de lenguajes dentro de la Ingeniería Informática y dentro de los perfiles profesionales.
- OI7: Adquirir la capacidad de crear documentaciones técnicas completas, correctas y legibles.

2.2. Objetivos interpersonales generales

OIP1: Aplicar los objetivos interpersonales comunes: destrezas para la participación responsable: capacidad de coordinación, asistencia, contribuciones al grupo...

2.3. *Objetivos sistémicos generales*

- OS1: Aplicar los objetivos sistémicos generales comunes en el ámbito de esta materia.
- OS2: Desarrollar la madurez necesaria en el proceso de abstracción para abordar problemas reales y plantear modelos y soluciones que se basen, al menos parcialmente, en la Teoría de Lenguajes Formales.
- OS3: Reforzar el hábito de desarrollar diferentes alternativas de solución, tanto de análisis como de síntesis, cuestionando las características, riesgos y viabilidad de cada una, para cada problema planteado.

2.4.- *Objetivos específicos*

- OE1. Comprender cada parte del proceso de compilación y las diferentes formas de abordarlo, según las características del problema y el tipo de solución requerida.
- OE2. Estudiar la importancia del análisis sintáctico en el proceso de compilación y reconocer y saber aplicar cada tipo de gramática con el analizador sintáctico más adecuado.
- OE3. Relacionar el análisis sintáctico con el semántico, y para ello, estudiar la extensión de las representaciones sintácticas para incorporar atributos que permiten la incorporación de información semántica
- OE4. Relacionar las representaciones intermedias resultantes del análisis sintáctico y semántico con la gestión de memoria y todos los problemas derivados de la generación de código de un programa.
- OE5. Saber utilizar herramientas representativas para la producción de un procesador de lenguaje.
- OE6. Desarrollar una actitud crítica ante los LP y los problemas relacionados con su implementación, así como la mejor manera de aprovechar su funcionalidad.

3. Competencias

3.1. Competencias instrumentales

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

3.1.1. Habilidades cognitivas

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

Generales

- CIC1: Conocer y entender las relaciones que mantiene la asignatura con el resto de asignaturas de la titulación a partir de la introducción explícita de ejemplos y comentarios, aprovechando puntos relevantes del temario.
- CIC2: Conocer y comprender algunas de las capacidades, aptitudes y conocimientos que la asignatura aporta para el desarrollo de los diversos perfiles profesionales.
- CIC3: Ser capaz de crear documentaciones técnicas completas, correctas y legibles.

Competencias específicas.

Unidad Didáctica 1) Introducción. Generalidades. Historia. Compiladores.

- CE1.- Conocer la génesis e historia de los lenguajes de programación.
- CE2.- Aprender qué es un compilador
- CE3.- Conocer los diferentes paradigmas de programación.
- CE4.- Aprender la estructura y fases de un compilador

Bibliografía .-

Programming Language Essentials.- H.Bal, D. Grune. Editorial Addison Wesley, 97.
Modern Compiler Design.- Grune, Bal, Cerial, Lagendoen.- Wiley 2001

Unidad Didáctica 2) Aspectos formales. Definiciones. BNF. Lenguajes Formales. Gramáticas. Análisis sintáctico. Grafos sintácticos.

- CE1.- Recordar los conceptos básicos de teoría de autómatas y lenguajes formales.
- CE2.- Comprender la clasificación jerárquica de Chomsky.
- CE3.- Conocer las principales metodologías de descripción y representación de gramáticas.

Bibliografía .-

Programming Language Essentials.- H.Bal, D. Grune. Editorial Addison Wesley, 97.
Compiladores: Conceptos Fundamentales. Teufel, Schmidt, Teufel. Editorial Addison Wesley, 95
Lenguajes, Gramáticas y Autómatas.- P.Isasi, P.Martínez, D.Borrajo. Editorial Addison Wesley, 97

Unidad Didáctica 3) Análisis de léxico. Tablas de símbolos. Herramientas.

- CE1.- Conocer el funcionamiento de un analizador léxico.
- CE2.- Entender la relación con la tabla de símbolos.
- CE3.- Entender la relación con la gestión de errores.
- CE4.- Aprender a especificar formalmente un compilador.
- CE5.- Conocer alguna herramienta de generación de analizadores léxicos.

Bibliografía .-

Compiladores: Conceptos Fundamentales. Teufel, Schmidt, Teufel. Editorial Addison Wesley, 95

Construcción de Compiladores.- K.C.Louden. Editorial Thomson, 2004.

Compiladores: Principios, Técnicas y Herramientas. Aho, Sethi, Ullman. Editorial Addison Wesley, 90.

Diseño de Compiladores.- Alicia Garrido, J. Manuel Iñesta, Francisco Moreno, J. Antonio Perez.- Universidad de Alicante 2002

Introduction to Compiling Techniques.- J.P.Bennet. Mc Graw Hill, 96

Lex&Yacc.- Levine, Mason, Brown. Editorial O'Reilly, 92

Unidad Didáctica 4) Análisis sintáctico. Construcción de analizadores. Herramientas.

- CE1. Introducir el funcionamiento de un análisis sintáctico y terminología.
- CE2. Presentar la especificación formal de analizadores en forma de gramáticas
- CE3. Identificar los tipos de analizadores sintácticos existentes
- CE4. Entender la técnica de análisis descendente
- CE5. Presentar los analizadores con retroceso
- CE6. Aprender el cálculo de los conjuntos primero y siguiente.
- CE7. Aprender a comprobar las condiciones LL(1) para una gramática
- CE8. Aprender a modificar las gramáticas para que cumplan LL(1)
- CE9. Presentar los analizadores descendentes recursivos
- CE10. Presentar los analizadores descendentes dirigidos por tabla
- CE11. Entender la técnica de análisis ascendente
- CE12. Aprender la especificación de gramáticas para análisis ascendente
- CE13. Aprender el algoritmo de análisis por reducción desplazamiento
- CE14. Aprender el esquema de los analizadores SLR y LALR
- CE15. Distinguir los diferentes tipos de gramáticas
- CE16. Conocer algunas de las herramientas disponibles: yacc, bison, cup, etc.

Bibliografía.-

1. *Construcción de Compiladores*.- K.C.Louden. Editorial Thomson, 2004.
2. *Compiladores: Principios, Técnicas y Herramientas*. Aho, Sethi, Ullman. Editorial Addison Wesley, 90.
3. *Diseño de Compiladores*.- Alicia Garrido, J. Manuel Iñesta, Francisco Moreno, J. Antonio Perez.- Universidad de Alicante 2002
4. *Introduction to Compiling Techniques*.- J.P.Bennet. Mc Graw Hill, 96
5. *Lex&Yacc*.- Levine, Mason, Brown. Editorial O'Reilly, 92
6. *Modern Compiler Design*.- Grune, Bal, Ceriel, Lagendoen.- Wiley 2001

Unidad Didáctica 5) Semántica y análisis de tipos. Traducción dirigida por la sintaxis. Gramáticas Atribuidas.

CE1.- Comprender el concepto de “semántica” de una sentencia.

CE2 .- Examinar las distintas alternativas de tratamiento de la semántica.

CE3.- Aplicar los principios de la traducción dirigida por sintaxis: ETDS y DDS.

CE4.- Dar una introducción formal a las gramáticas atribuidas y su relación con los esquemas de análisis sintáctico.

Bibliografía:

1. *Construcción de Compiladores*.- K.C.Louden. Editorial Thomson, 2004.
2. *Diseño de Compiladores*.- Alicia Garrido, J. Manuel Iñesta, Francisco Moreno, J. Antonio Perez.- Universidad de Alicante 2002
3. *Introduction to Compiling Techniques*.- J.P.Bennet. Mc Graw Hill, 96

Unidad Didáctica 6) Gestión y manejo de errores de léxico, sintácticos y semánticos.

CE1.- Comprender la importancia de una buena gestión de los errores.

CE2.- Analizar los diferentes tipos de errores que se pueden producir durante el proceso de compilación.

CE3.- Examinar las diferentes técnicas de tratamiento y recuperación de los errores.

Bibliografía:

1. *Construcción de Compiladores*.- K.C.Louden. Editorial Thomson, 2004.
2. *Diseño de Compiladores*.- Alicia Garrido, J. Manuel Iñesta, Francisco Moreno, J. Antonio Perez.- Universidad de Alicante 2002
3. *Introduction to Compiling Techniques*.- J.P.Bennet. Mc Graw Hill, 96

Unidad Didáctica 7) Generación y Optimización de código. Código intermedio. Gestión de memoria.

- CE1.- Estudiar la parte final, de síntesis, del compilador.
- CE2.- Comprender la importancia del código intermedio y las diferentes propuestas.
- CE3.- Examinar y comprender las diferentes alternativas en la gestión de memoria.
- CE3.- Tratar el concepto de “optimización” del código y examinar algunas técnicas genéricas.

Bibliografía:

1. *Construcción de Compiladores*.- K.C.Louden. Editorial Thomson, 2004.
2. *Diseño de Compiladores*.- Alicia Garrido, J. Manuel Iñesta, Francisco Moreno, J. Antonio Perez.- Universidad de Alicante 2002
3. *Introduction to Compiling Techniques*.- J.P.Bennet. Mc Graw Hill, 96

Unidad Didáctica 8) Herramientas de generación de compiladores. (Prácticas)

- CE1.- Conocer a nivel de usuario las diferentes herramientas para ayuda a la construcción de compiladores: lex, yacc, cup, yaccie, etc.
- CE2.- Ser capaz de usarlas para construir las primeras fases de un pequeño compilador.

Bibliografía:

1. *Construcción de Compiladores*.- K.C.Louden. Editorial Thomson, 2004.
2. *Lex&Yacc*.- Levine, Mason, Brown. Editorial O'Reilly, 92

3.1. Capacidades metodológicas

CIM1: Tener capacidad de análisis y síntesis.

CIM2: Ser capaz de manejar bibliografía relacionada con los Procesadores de lenguajes.

3.1.3. Destrezas tecnológicas

CIT1: Manejar con fluidez diferentes herramientas de diseño de Procesadores de Lenguajes..

3.1.4. Destrezas lingüísticas

CIL1: Conocer y utilizar la terminología usual de la Ingeniería del Software, tanto en español como en inglés.

3.2. *Competencias interpersonales*

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

3.2.1. Competencias para tareas colaborativas

CIPTC1: Ser capaz de realizar de trabajar en equipo para resolver los problemas de compilación enunciados en las prácticas de la asignatura.

CIPTC2: Ser capaz de presentar en público una solución a un problema planteado y mantener un debate con el resto de la clase sobre la solución planteada, para así buscar colaborativamente la mejor solución al problema.

CIPTC3: Ser capaz de realizar y defender la práctica obligatoria de la asignatura en equipo.

3.2.2. Compromiso con el trabajo

CIPTR1: Se debe cumplir el plazo de entrega de los trabajos.

CIPTR2: Se debe adquirir un compromiso ético entre todos los componentes del grupo.

3.3. Competencias sistémicas

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

CS1: Capacidad de aplicar los conocimientos, métodos y herramientas vistos en la asignatura de Procesadores de Lenguajes a situaciones y problemas concretos del área de la Ingeniería Informática y de otras disciplinas relacionadas.

CS2: Motivación por la calidad y por la creatividad.

CS3: Capacidad de adoptar el proceso marcado por el método científico y de ingeniería en el planteamiento y realización de trabajos diversos, tanto a nivel académico como profesional.

4. Prerrequisitos

4.1. Competencias y contenidos mínimos

- Conocer los elementos básicos de un lenguaje de programación de 3^a generación.
- Conocer los principios fundamentales del diseño de estructuras de datos.
- Conocer la teoría de lenguajes formales.

4.2. Plan de trabajo para la consecución de los prerrequisitos

Los prerrequisitos para el estudio y entendimiento de la asignatura Procesadores de Lenguajes se cubren fundamentalmente en las asignaturas de Teoría de Autómatas y Lenguajes Formales de la Ingeniería Técnica de Informática de Sistemas o del Primer Ciclo de la Ingeniería en Informática.

5. Temario

5.1. Unidades didácticas

Unidad Didáctica 1) Introducción. Generalidades. Historia. Compiladores.

Unidad Didáctica 2) Aspectos formales. Definiciones. BNF. Lenguajes Formales. Gramáticas. Análisis sintáctico. Grafos sintácticos.

Unidad Didáctica 3) Análisis de léxico. Tablas de símbolos. Herramientas.

Unidad Didáctica 4) Análisis sintáctico. Construcción de analizadores. Herramientas.

Unidad Didáctica 5) Semántica y análisis de tipos. Traducción dirigida por la sintaxis. Gramáticas Atribuidas

Unidad Didáctica 6) Gestión y manejo de errores de léxico, sintácticos y semánticos.

Unidad Didáctica 7) Generación y Optimización de código. Código intermedio. Gestión de memoria.

Unidad Didáctica 8) Herramientas de generación de compiladores. (Prácticas)

6. Metodología y estrategias de aprendizaje

6.1. Metodología docente

Siendo congruentes con los principios que presiden lo que se ha dado en llamar Espacio Europeo de Educación Superior (EEES), y con carácter general, el Ingeniero en Informática debe estar capacitado para aprender a conocer, hacer, convivir y ser, en su ámbito personal, profesional y social, de acuerdo con lo recogido en el informe de la UNESCO sobre las perspectivas de la educación en el siglo XXI.

Para caminar en este sentido, el modelo educativo que se va a seguir en la asignatura de Procesadores de Lenguajes, como en casi todas las asignaturas de la Ingeniería en Informática, tiene en la clase magistral un elemento importante, pero ya no exclusivo, en la transmisión de conocimiento. Este tipo de enseñanza se va a complementar con otros procesos entre los que cabe destacar las prácticas basadas en enseñanza colaborativa. Concretamente las actividades que se proponen son las siguientes:

- *Clases de teoría con apoyo de material audiovisual.* En estas clases se presentarán los contenidos básicos de un cierto tema. Las clases comenzarán con una breve introducción de los contenidos que se pretenden transmitir en la clase, así como con un breve comentario a los conceptos vistos en clases anteriores y que sirven de enlace a los que se pretenden desarrollar. El desarrollo de la clase se llevará a cabo con medios audiovisuales, textos, transparencias... que permitan un adecuado nivel de motivación e interés en los alumnos. Es importante intentar terminar la exposición con las conclusiones más relevantes del tema tratado.

Las transparencias que se utilizarán en clase están extraídas de las notas que se facilitan a los alumnos en la página web de la asignatura. Estas notas son una guía para el estudio, pero no son sustitutas de la bibliografía recomendada.

- *Prácticas.* Las clases prácticas presenciales estarán dedicadas a la familiarización del alumno en el uso de alguna de las herramientas disponibles para la ayuda al diseño y construcción de compiladores. Entre ellas, podemos citar:
 - *Lex, flex, jlex* : para construcción de analizadores léxicos basados en expresiones regulares.

- *Yacc, bison, cup* : para la construcción de analizadores sintácticos ascendentes.
- *Jaccie* : herramienta de gran ayuda a la docencia porque permite construir, paso a paso, los componentes de un compilador, ilustrar alguno de los pasos y conceptos intermedios, elegir entre políticas de análisis, etc.
- *Práctica obligatoria*. Los alumnos, en grupos de 2, deberán realizar una práctica obligatoria que consistirá en la implementación de la primera fase (hasta la generación de código intermedio) de un pequeño compilador de un lenguaje cuya especificación se le dará a lo largo del curso. Esta práctica tendrá que entregarse antes de una fecha determinada, defenderse en grupo y representará una importante parte de la nota final de la asignatura.
- *Tutorías*. El alumnado tiene a su disposición seis horas de tutorías a la semana en las que puede consultar cualquier duda relacionada con los contenidos, organización y planificación de la asignatura. Las tutorías pueden ser individualizadas, pero se admite tutorías grupales para resolver problemas relacionados con las actividades a realizar en grupo.
- *Página web*. Se convierte en el vehículo de comunicación y registro de información de la asignatura.

6.2.4. Evaluación

- Existirá una prueba final, en las fechas determinadas en el calendario escolar, que consistirá en dos partes:
 - Una prueba teórica de tipo test.
 - Una prueba práctica, escrita, con algunos problemas relacionados con la teoría de la asignatura.

La nota final se calculará como resultado de la de la prueba final (65%) y de la práctica obligatoria (35%).

7. Planificación del tiempo y del esfuerzo

La asignatura de Procesadores de Lenguajes tiene en el plan de estudios actual 9 créditos LRU asignados, 6 créditos de teoría y 3 créditos de práctica. Para hacer su estudio en ECTS se ha tomado tomando la equivalencia de crédito LRU a crédito ECTS, con implicación de cambio de metodología docente.

Al hacer el estudio de carga de trabajo máxima, se tiene que:

- 9 ECTS -> 225-270 horas de trabajo
- 6 cr teoría LRU -> 60 horas presenciales + (60*1,5) horas de asimilación = 150 horas
- 3 cr práctica LRU -> 30 horas presenciales + (30*2) horas de asimilación = 90 horas

- $150 + 90 = 240$ horas de trabajo

Por temas de organización, calendario, y uniformización con el resto de las asignaturas se ha optado por partir del escenario que asigna 25 horas de trabajo al ECTS.

Para poder elaborar las guías, cuyo destino final sería la realización de experiencias piloto, dentro de los planes de estudio vigentes, se debe tener en cuenta la definición del crédito del R.D. 1497/1987 y sus posteriores modificaciones. Así:

- La unidad de valoración de las enseñanzas se corresponde a diez horas de enseñanza teórica, práctica o de sus equivalencias entre las que podrán incluirse actividades académicas dirigidas, que habrán de preverse en el correspondiente plan docente junto con los mecanismos y medios objetivos de comprobación de los resultados académicos de las mismas. Todo ello sin perjuicio del cumplimiento del régimen de dedicación del profesorado, de conformidad con el Real Decreto 898/1985, de 30 de abril sobre régimen del profesorado universitario.
- En ningún caso, salvo que se trate de enseñanzas en Universidades a distancia, el porcentaje del crédito correspondiente a las actividades académicas dirigidas será superior al **30%**.

De esta forma, si 9 créditos LRU equivalen a 90 horas, se debe asegurar el 70% como mínimo de presencialidad, esto es, 63 horas entre horas de teoría y de práctica.

En la Tabla 1 se presenta la organización del esfuerzo del alumno para cubrir los 9 ECTS asociados a la asignatura. Por su parte en la Tabla 2 se distribuyen las horas presenciales entre los diferentes temas y el examen.

			A	B	C	D	E
	Técnica	Actividad	Horas equivalentes de clase	Factor de trabajo del alumno†	Horas de trabajo personal del alumno	Horas totales (A+C)	ECTS (D ÷ 25)
Teoría	Seguimiento de la parte teórica de las unidades didácticas	Asimila contenidos. Se plantea dudas que planteará a los profesores en las tutorías	40	1,5	60	100	4
Práctica	Seguimiento de la parte práctica de las unidades didácticas 8	Experimenta, practica, modela	25	2	50	75	3
Práctica obligatoria	Práctica en grupo de obligada realización	Realiza las primeras fases de un compilador	-	-	40	40	1,7
Examen	examen final de la asignatura	Exámenes tipo test, supuestos prácticos...	2,5			2,5	0,1

† Número de horas dedicadas por el alumno al trabajo personal (organización de apuntes, estudio, documentación...) por cada hora de clase.

Otras actividades	Tutorías personalizadas y grupales	Recibe orientación personalizada	-	-	2,5	2,5	0,1
	Búsquedas en la red, participación en foros especializados...	Busca elementos para completar los contenidos	-	-	2,5	2,5	0,1
TOTAL			67,5		155	222,5	9

Tabla 1. Escenario ECTS = 25 horas de trabajo

NÚMERO DE HORAS PRESENCIALES		
Actividad	Horas Teoría	Horas Prácticas
Unidad Didáctica 1: Conceptos básicos	4	
Unidad Didáctica 2:	5	
Unidad Didáctica 3:	6	
Unidad Didáctica 4:	8	
Unidad Didáctica 5:	7	
Unidad Didáctica 6:	3	
Unidad Didáctica 7:	7	
Unidad Didáctica 8:	0	25
Examen	2,5	
TOTAL: 67,5	42,50	25

Tabla 2. Reparto de horas presenciales

8. Bibliografía

8.1. Bibliografía básica

1. *Compiladores: Conceptos Fundamentales.* Teufel, Schmidt, Teufel. Editorial Addison Wesley, 95
2. *Construcción de Compiladores.*- K.C.Louden. Editorial Thomson, 2004.
3. *Compiladores: Principios, Técnicas y Herramientas.* Aho, Sethi, Ullman. Editorial Addison Wesley, 90.
4. *Diseño de Compiladores.*- Alicia Garrido, J. Manuel Iñesta, Francisco Moreno, J. Antonio Perez.- Universidad de Alicante 2002
5. *Introduction to Compiling Techniques.*- J.P.Bennet. Mc Graw Hill, 96
6. *Modern Compiler Implementation in C: Basic Techniques.* Appel, Ginsberg. Editorial Cambridge University Press, 97.
7. *Lex&Yacc.*- Levine, Mason, Brown. Editorial O'Reilly, 92
8. *Programming Language Essentials.*- H.Bal, D. Grune. Editorial Addison Wesley, 97.
9. *Lenguajes, Gramáticas y Autómatas.*- P.Isasi, P.Martínez, D.Borrajo. Editorial Addison Wesley, 97
10. *Linkers&Loaders.*- John R. Levine.- Morgan Kaufmann 2000
11. *Modern Compiler Design.*- Grune, Bal, Ceriel, Lagendoen.- Wiley 2001

8.3. Otros recursos

<http://www.lcc.uma.es/docencia/ETSIInf/pl/pl1.html> Apuntes de Procesadores de Lenguaje Equipo docente formado por R. Conejo Muñoz, L. Mora Lõpez, y E. Medina Cano Universidad de Málaga

<http://www.di.uniovi.es/procesadores/apuntes/apuntes.htm> Apuntes de Compiladores J. M. Cueva Lovelle (principalmente) y el resto del equipo docente: M.C. Luengo Díez, J.E. Labra Gayo, A.A. Juan Fuente, R. Izquierdo Castanedo y F. Ortín Soler Universidad de Oviedo

<http://www.cs.princeton.edu/%7Eappel/modern/java/JLex/> Página web de JLex.

<http://jflex.de/> Página web de Jflex. Se usa para especificar analizadores léxicos. Para ello se utilizan reglas que definen expresiones regulares como patrones en que encajar los caracteres que se van leyendo del archivo fuente, obteniendo tokens. JFlex actualiza la herramienta JLex, también utilizada en esta asignatura. Se ofrece también su página web y una guía sobre su uso; el alumno puede utilizar la guía sin problemas, ya que los analizadores descritos en JLex son válidos en JFlex.

<http://www2.cs.tum.edu/projects/cup/> Página web de Cup Esta herramienta permite especificar gramáticas formales facilitando el análisis sintáctico, pero también permite asociar acciones a los símbolos que van apareciendo en las reglas gramaticales, de forma que el análisis semántico y la generación de código intermedio pueden integrarse en el código. Se ofrece una guía de uso de Cup para la práctica.

<http://www2-data.informatik.unibw-muenchen.de/Research/Tools/JACCIE/downloadEN/download.html> Página web de jaccie. Esta herramienta no se utiliza en el desarrollo de la práctica, pero resulta útil como ayuda para la especificación de analizadores léxicos y sintácticos. Puede ayudar también al estudio teórico de la asignatura, ya que permite calcular conjuntos de primeros y siguientes, visualizar árboles de análisis sintáctico y comprobar conflictos gramaticales.

<http://ant.apache.org/> Página web de ant, una herramienta muy útil para automatizar la compilación y ejecución de programas escritos en java. La generación de un compilador utilizando JFlex y Cup se realiza mediante una serie de llamadas a clases java y al compilador de java. Ant permite evitar situaciones habituales en las que, debido a configuraciones particulares del proceso de compilación, la práctica sólo funciona en el ordenador del alumno.

9. Evaluación de los procesos y resultados de aprendizaje

9.1. Sistema de evaluación

La evaluación de esta asignatura se divide en dos partes diferenciadas:

- **Parte de Teoría** (65% de la nota final).
 - Un examen final que consta de dos partes que hay que aprobar independientemente:
 - Un test.
 - Un conjunto de supuestos teórico/prácticos.
 - La parte de teoría se guardará durante el curso académico actual.
- **Parte Práctica** (35% de la nota final).
 - Práctica realizada en grupos de trabajo de dos alumnos .

- Se realizará una defensa de dicho trabajo.
- Todo grupo tiene derecho a dos defensas, siempre que la primera se haya hecho antes del mes de mayo.
- La parte práctica se guardará durante el curso académico actual.

La nota final de la asignatura se calculará conforme a la siguiente fórmula:

Si (Teoría $\geq 5,0$) y (Práctica $\geq 5,0$)

Nota Final = (Teoría*0,65) + (Práctica*0,35)

Si no

suspenso

Fin si

Guía docente de Hipermedia: Diseño y Evaluación

Versión 1.0 – 9-5-2006

JOSÉ LUIS PÉREZ IGLESIAS – jpi@usal.es

Dpto. de Informática y Automática - Escuela Politécnica Superior de Zamora

Avda. Cardenal Cisneros 34, “Campus Viriato”, 49022, Zamora

Tlfno. 980 545000 Ext. 3636 – Fax. 980 545001

1. Contexto

1.1. Perfil de los créditos y su adecuación al perfil profesional y académico de la titulación

La Hipermedia, como asignatura dentro del Plan de Estudios actual, consta de 7,5 créditos LRU (3 teóricos y 4,5 prácticos). El objetivo de la asignatura es proporcionar al alumno los conceptos básicos de la interacción persona-ordenador (Figura 2). Ésta es la disciplina relacionada con el diseño, evaluación e implementación de sistemas de computadores interactivos de uso humano y el estudio de los fenómenos que la rodean. Esta asignatura debería ser un primer paso en una formación completa en tecnologías interactivas que abarcase las siguientes áreas:

- Introducción a la interacción persona-ordenador
- Sistemas multimedia e hipermedia
- Sistemas de realidad virtual y realidad aumentada
- Interacción en lenguaje natural
- Interfaces gráficas de usuario
- Interfaces para Internet

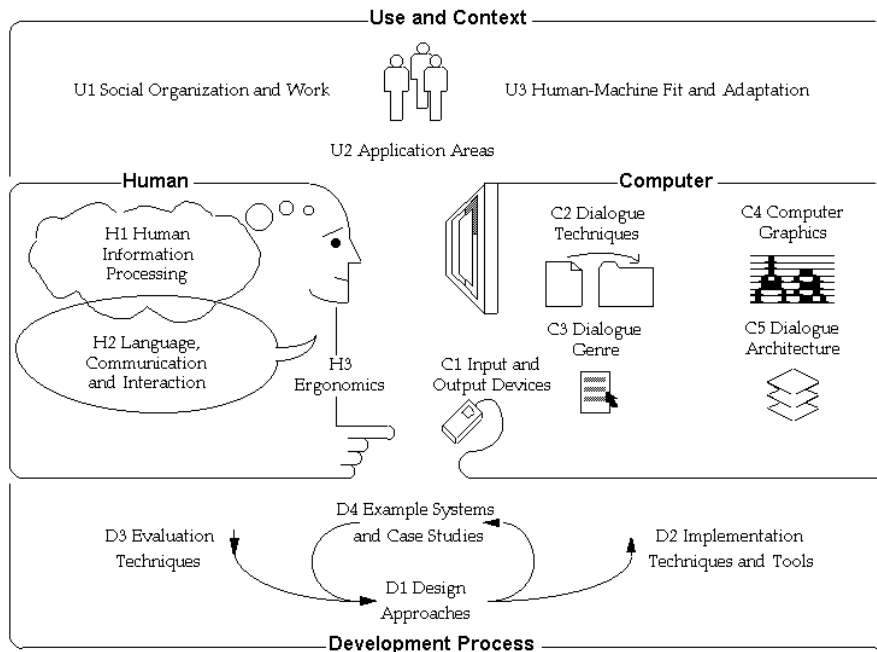


Figura 2. Interacción Persona-Ordenador

Tomada del *ACM SIGCHI Curricula for Human-Computer Interaction, 2004*.

Con ello el alumno adquiere una formación que combina los aspectos teóricos y prácticos de la disciplina y está perfectamente capacitado para desarrollar sistemas interactivos, con especial énfasis en el desarrollo de interfaces de manipulación directa, interfaces multimedia e interfaces Web, así como conocer todos los parámetros necesarios en la evaluación de dichos desarrollos.

Informe de ACM/IEEE-CS “*Joint Curriculum Task Force Computing Curricula 2005*” establece doce áreas temáticas para distribuir la materia de la disciplina de informática (ver en Anexo). Cabe citar que en el *Computing Curricula* del 2005 (CC 2005), esta materia se cita dentro del perfil profesional de *Computer Science*, en la categoría interfaz-hombre-máquina¹⁴, si bien el peso en el resto de los perfiles profesionales es bastante significativo, como se muestra en la Tabla 1 del Anexo.

1.2. Ubicación y relaciones en el plan de estudios

La asignatura de *Hipermedia: Diseño y evaluación* se imparte en el primer cuatrimestre del 3º curso de la Ingeniería Técnica en Informática de Gestión, cuyo plan de estudios actual data del año 2003 (B.O.E. de 13/02/2003).

Las interfaces de usuario son una parte importante del éxito de una aplicación interactiva. Por ello, esta asignatura está estrechamente relacionada con casi cualquier asignatura que requiera la interacción con un computador (incluidas las asignaturas de

¹⁴ El Curricula para la Interacción Hombre-Máquina se desarrolla en el *SIGCHI Curricula For Human-Computer Interaction*.

segundo ciclo). En la Figura 3 se muestra la relación con aquellas a las que afecta más directamente.

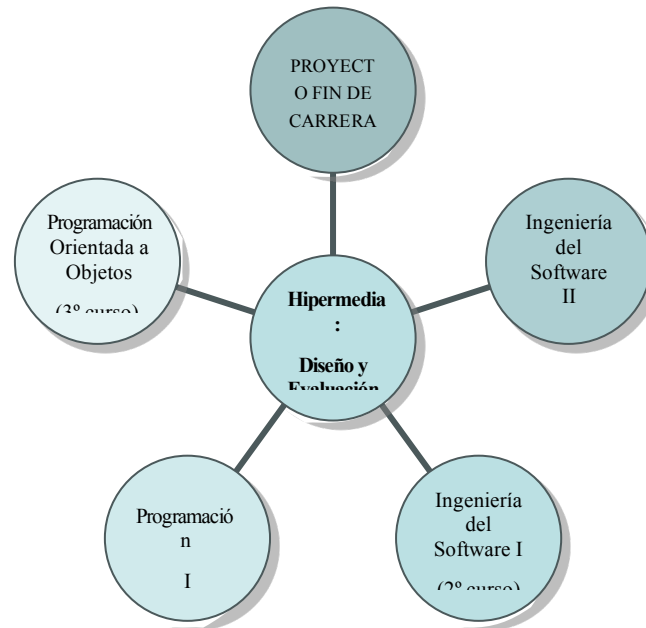


Figura 3. Relación con las asignaturas de la Titulación de Ingeniero Técnico en Informática de Gestión.

2. Objetivos

2.1. Objetivos instrumentales generales

- OI1: Aplicar los conocimientos adquiridos mediante la resolución de problemas y prácticas obligatorias.
- OI2: Adquirir y emplear un buen lenguaje formal, tanto oral como escrito, siendo riguroso en las explicaciones de cualquier proceso.
- OI3: Conocer y utilizar la terminología utilizada en Interacción persona-ordenador.
- OI4: Adquirir un buen manejo de la bibliografía recomendada en la asignatura, de forma que se potencia la autosuficiencia a la hora de completar la formación.
- OI5: Comprender el ámbito de la Hipermedia dentro de la Ingeniería Informática dentro de los perfiles profesionales.
- OI6: Tomar conciencia de las implicaciones del trabajo, tanto individualmente como formando parte de un equipo
- OI7: Adquirir una visión inicial del campo de la interacción persona-ordenador (IPO).

- OI8: Analizar los factores humanos que afectan y condicionan la interacción con una máquina.
- OI9: Analizar los distintos dominios de aplicación de la IPO: entornos, modelos y teorías.
- OI10: Aprender a realizar diseños centrados en el usuario.
- OI11: Desarrollar interfaces efectivos.
- OI12: Utilizar con fluidez herramientas de diseño de interfaces.
- OI13: Tener en cuenta factores de accesibilidad, para facilitar el acercamiento de la máquina a las personas con capacidades limitadas.
- OI14: Realizar software centrado en las personas, analizando su forma de trabajar mediante el análisis de sus tareas y las situaciones en que se producen.
- OI15: Conocer y aplicar los principios de un método de análisis y diseño estructurado.
- OI16: Adquirir la capacidad de desarrollar documentaciones técnicas completas, correctas y legibles, siguiendo una guía de estilo apropiada.

2.2. Objetivos interpersonales generales

- OIP1: Destrezas para la participación responsable: capacidad de coordinación, asistencia, contribuciones al grupo, etc.
- OIP2: Capacidad de trabajar en equipo adquiriendo y mejorando las habilidades sociales y la inteligencia emocional.
- OIP3: Comprometerse de forma ética con el trabajo, con el resto de los integrantes del grupo y consigo mismo.

2.3. Objetivos sistémicos generales

- OS1: Capacidad de integrar los conocimientos y destrezas prácticas de las diferentes asignaturas del plan de estudio para resolver situaciones reales relacionadas con la Ingeniería Informática, así como con otras disciplinas relacionadas.
- OS2: Adquirir el hábito de plantearse interrogantes. Ante un problema desarrollar distintas hipótesis de resolución y su relación entre ellas, así como posibles efectos al plantear variantes del problema.
- OS3: Capacidad de aplicar y relacionar, de forma autónoma, los contenidos de la Hipermedia de forma interdisciplinar.
- OS4: Comprender y utilizar el método científico mediante diversas actividades realizadas en la asignatura, y asimilar su importancia como manera de

pensar y actuar en la labor de científico e ingeniero, fomentando principalmente la capacidad de abstracción y el espíritu crítico.

OS5: Desarrollar la madurez necesaria en el proceso de abstracción para abordar problemas reales y plantear modelos y soluciones de forma razonada y correcta.

OS6: Reforzar el hábito de desarrollar diferentes alternativas, cuestionando las características, riesgos y viabilidad de cada una, para cada problema planteado.

3. Competencias

3.1. Competencias instrumentales

En este apartado es conveniente realizar una distinción entre habilidades cognitivas, capacidades metodológicas, destrezas tecnológicas y destrezas lingüísticas.

3.1.1. Habilidades cognitivas

Se distinguirán unas habilidades cognitivas generales y otras específicas para cada unidad didáctica.

Generales

CIC1: Establecer las conexiones con el resto de las asignaturas de la Titulación.

CIC2: Entender la utilidad aportada por los contenidos de la disciplina, así como desarrollar las capacidades, aptitudes y conocimientos que la asignatura aporta para el futuro profesional.

CIC3: Desarrollar interfaces usables, comprensibles y accesibles, así como la documentación necesaria para su utilización.

Específicas

Unidad didáctica I: La interacción persona-ordenador (IPO)

CIC4: Conocer la importancia de la disciplina.

CIC5: Conocer los objetivos de la IPO.

CIC6: Entender el significado de usabilidad de un sistema y aprender a analizar si un sistema es usable.

CIC7: Comprender la importancia de que los ordenadores sean accesibles a todas las personas.

CIC8: Conocer que hay muchas disciplinas que contribuyen a la IPO y valorar su aportación.

- CIC9: Tener una visión general del estado actual de los dispositivos de interacción.
- CIC10: Aprender y utilizar criterios para poder seleccionar los dispositivos adecuados para la resolución de un problema determinado.

Unidad didáctica II: La persona y su entorno de trabajo

- CIC11: Conocer los procesos de percepción más relevantes desde el punto de vista de la interacción.
- CIC12: Conocer cómo se realiza el proceso de comprensión por parte de la persona y los distintos modelos de memoria existentes.
- CIC13: Comprender que el modelo de memoria condiciona el diseño de las interfaces.
- CIC14: Establecer la importancia que tienen las limitaciones de la Memoria Operativa de la persona.
- CIC15: Conocer qué es y qué aplicaciones tiene el Modelo Mental.
- CIC16: Conocer los diferentes tipos de trabajo por computadora y analizar ejemplos de aplicación.
- CIC17: Analizar los entornos de trabajo de los seres humanos y la forma en que afectan a su rendimiento.
- CIC18: Valorar la importancia de la ergonomía en el entorno de trabajo.
- CIC19: Entender el impacto del trabajo en grupo por computadora.
- CIC20: Conocer los diferentes tipos de trabajo por computadora, así como conocer y estudiar ejemplos de aplicación
- CIC21: Aprender los conceptos básicos del diseño de interfaces, incidiendo en aquellos que permitan el trabajo en grupo.

Unidad didáctica III: Desarrollo de software centrado en la persona

- CIC22: Aprender qué es una metáfora y distinguir entre metáforas verbales y visuales.
- CIC23: Conocer las metáforas globales.
- CIC24: Analizar la metáfora del Escritorio, su desarrollo y su historia.
- CIC25: Ver el uso de las metáforas en las interfaces actuales.
- CIC26: Discutir los problemas y las ventajas del uso de las metáforas.
- CIC27: Aprender a diseñar metáforas, así como aplicarlas con la máxima efectividad.
- CIC28: Entender y aprender qué es un estilo de interacción.
- CIC29: Tener una visión general y comparativa de los estilos y paradigmas de interacción.

- CIC30: Conocer el estado actual y la futura evolución de los estilos de interacción.
- CIC31: Aprender a elegir entre los diferentes paradigmas y dentro de estos paradigmas, qué estilos de interacción utilizar para una determinada aplicación.
- CIC32: Comprender la necesidad de los sistemas de soporte en línea al usuario.
- CIC33: Conocer el significado de Hipertexto e Hipermedia.
- CIC34: Conocer los diferentes tipos de soporte al usuario que pueden tener las aplicaciones.
- CIC35: Conocer los métodos y las tecnologías apropiadas para evaluar sistemas de soporte en línea.
- CIC36: Conocer los procesos y tecnologías básicas en la elaboración de soportes en línea.

Unidad didáctica IV: Diseño de interfaces efectivas

- CIC37: Distinguir entre principios y directrices.
- CIC38: Conocer los diferentes estándares existentes relacionados con la IPO.
- CIC39: Conocer las diferentes guías de estilo existentes.
- CIC40: Ver la necesidad y utilidad de los estándares y las guías de estilo.
- CIC41: Aprender a percibir una imagen y reconocer su lenguaje, identificando los elementos morfológicos de la propia imagen.
- CIC42: Verificar la función expresiva de los signos básicos en la composición.
- CIC43: Valorar el diseño ergonómico en la realización de la interfaz de usuario.
- CIC44: Conocer el proceso de diseño de sistemas interactivos.
- CIC45: Estudiar notaciones y métodos para el análisis de la interfaz de usuario.
- CIC46: Conocer y aplicar análisis de tareas.
- CIC47: Introducir conceptos de diseño orientado a objetos.
- CIC48: Analizar estrategias de diseño.

Unidad didáctica V: Usabilidad

- CIC49: Entender el concepto de Diseño Universal.
- CIC50: Promover la concienciación en el diseño y programación de interfaces de usuario respecto a la necesidad de un diseño universal.
- CIC51: Conocer los diversos tipos de discapacidades y algunas de las soluciones disponibles más establecidas.
- CIC52: Motivar a la utilización de técnicas de usabilidad teniendo también en cuenta a usuarios con discapacidades.
- CIC53: Saber qué se conoce como evaluación de un diseño.

- CIC54: Conocer los diferentes métodos de evaluación y sus diferencias
- CIC55: Aprender a realizar evaluaciones, así como conocer cuándo utilizar los diferentes métodos de evaluación dentro del ciclo de vida del software.
- CIC56: Valorar el coste que puede suponer su aplicación.
- CIC57: Saber cómo obtener conclusiones y cómo mejorar la usabilidad de un sistema que se esté evaluando.
- CIC58: Valorar la importancia de la internacionalización de las interfaces.
- CIC59: Reconocer los problemas derivados de la traducción a otros lenguajes y los diferentes alfabetos existentes.
- CIC60: Saber cómo evaluar si el software está adecuadamente internacionalizado.
- CIC61: Conocer los recursos técnicos sobre como internacionalizar las interfaces.

3.1.2. Capacidades metodológicas

- CIM1: Ser capaz de tomar decisiones de manera razonada.
- CIM2: Adquirir capacidad de análisis y síntesis.
- CIM3: Ser capaz de manejar bibliografía y otros recursos relacionados con la asignatura.
- CIM4: Ser capaz de proporcionar una solución a un problema de diseño y desarrollo de interfaz de usuario, partiendo del análisis y aproximaciones que éste precise.

3.1.3. Destrezas tecnológicas

- CIT1: Manejo avanzado de la búsqueda de información y recursos a través de la Web y utilización de los servicios proporcionados por ésta, para ser capaz de ampliar conocimientos sobre la asignatura.
- CIT2: Conocer el funcionamiento de distintos sistemas operativos y distintas máquinas, con el fin de poder analizar las ventajas y los inconvenientes de las interfaces de usuario de estas.
- CIT3: Ser capaz de desarrollar interfaces de forma efectiva con las herramientas necesarias para ello.

3.1.4. Destrezas lingüísticas

- CIL1: Ser capaz de manejar con fluidez el lenguaje científico, siendo riguroso en las demostraciones de los desarrollos de la asignatura.
- CIL2: Adquirir el hábito de hablar y discutir temas científicos en público.

CIL3: Conocer y utilizar la terminología usual de la Hipermedia, tanto en español como en otros idiomas, especialmente inglés.

3.2. Competencias interpersonales

Las competencias interpersonales hacen referencia a la capacidad de trabajo en equipo del alumno y se dividen en competencias para las tareas colaborativas y competencias relativas al compromiso con el trabajo.

3.2.1. Competencias para tareas colaborativas

CIPTC1: Ser capaz de trabajar en equipo con el propósito de resolver problemas de desarrollo enunciados en los talleres prácticos de la asignatura.

CIPTC2: Ser capaz de presentar ante sus compañeros una solución a un determinado problema y mantener un debate con el resto de la clase sobre la solución planteada, para así buscar colaborativamente la mejor solución al problema.

CIPTC3: Ser capaz de realizar y defender las prácticas obligatorias de la asignatura en equipo.

3.2.2. Compromiso con el trabajo

CIPTR1: Se ha de definir un plan de trabajo en el que el volumen de trabajo de todos los miembros del equipo sea similar.

CIPTR2: Una vez finalizado el trabajo, todos los miembros del grupo deben conocer en profundidad todo el desarrollo realizado.

CIPTR3: Se debe cumplir el plazo de entrega de los trabajos.

CIPTR4: Se debe adquirir un compromiso ético entre todos los componentes del grupo.

3.3. Competencias sistémicas

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

- CS1: Capacidad de aplicar los conocimientos, métodos y herramientas vistos en la asignatura de Hipermedia a situaciones y problemas concretos del área de la Ingeniería Informática y de otras disciplinas relacionadas.
- CS2: Capacidad de aprender y aplicar, de forma autónoma e interdisciplinar, nuevos conceptos y métodos relacionados con la asignatura.
- CS3: Motivación por la calidad y por la creatividad.
- CS4: Capacidad de adoptar el proceso marcado por el método científico y de ingeniería en el planteamiento y realización de trabajos diversos, tanto a nivel académico como profesional.
- CS5: Capacidad de asimilación y adaptación a la evolución del estado del arte en el ámbito de la Hipermedia y de la Ingeniería Informática como profesión.

4. Prerrequisitos

4.1. Competencias y contenidos mínimos

- Conocer los elementos básicos de los lenguajes de programación.
- Entender la forma en que se trabaja a través de una interfaz de usuario.
- Tener manejo del uso de los distintos servicios Web existentes.
- Ser consciente de los problemas que se plantean en la interacción con un computador, a nivel de usuario.
- Conocer el ciclo de vida de un proyecto software.

4.2. Plan de trabajo para la consecución de los prerrequisitos

Los prerrequisitos para el estudio y entendimiento de la asignatura se cubren fundamentalmente con la asignatura Sistemas Informáticos de primer curso, aunque es conveniente haber cursado alguna asignatura de Programación y de Ingeniería del Software para comprender los problemas asociados a la interacción persona-ordenador y al diseño de interfaces, así como los conceptos fundamentales del ciclo de vida de un proyecto software.

5. Temario

5.1. Unidades didácticas

Unidad didáctica I: La interacción persona-ordenador

Tema 1.- Introducción a la interacción persona-ordenador (IPO)

Tema 2.- Dispositivos de interacción

Unidad didáctica II: La persona y su entorno de trabajo

Tema 3.- El factor humano

Tema 4.- Uso y contexto de los computadores

Tema 5.- Trabajo en red

Unidad didáctica III: Desarrollo de software centrado en la persona

Tema 6.- Metáforas

Tema 7.- Estilos de interacción

Tema 8.- Gestión de la información

Unidad didáctica IV: Diseño de interfaces efectivas

Tema 9.- Estándares y guías

Tema 10.- El diseño gráfico

Tema 11.- Ingeniería de la interfaz

Unidad didáctica V: Usabilidad

Tema 12.- Accesibilidad

Tema 13.- Diseño y Evaluación

Tema 14.- Diálogo persona-ordenador

Tema 15.- Internacionalización

5.2. Desarrollo de las unidades didácticas

Unidad didáctica I: La interacción persona-ordenador

Tema 1.- Introducción a la interacción persona-ordenador (IPO)

1.1 La disciplina

1.2 Historia de la IPO

- 1.3 Objetivos de la IPO
- 1.4 La interfaz de usuario
- 1.5 La interdisciplinariedad de la IPO
- 1.6 Usabilidad
- 1.7 El diseño centrado en el usuario

Tema 2.- Dispositivos de interacción

- 2.1 Gestión de los periféricos
- 2.2 Periféricos de entrada
- 2.3 Periféricos de salida
- 2.4 Realidad virtual y aumentada

Unidad didáctica II: La persona y su entorno de trabajo

Tema 3.- El factor humano

- 3.1 Lenguaje, comunicación e interacción
- 3.2 Cognición individual y distribuida
- 3.3 Procesamiento de información en la persona
- 3.4 Representación del conocimiento

Tema 4.- Uso y contexto de los computadores

- 4.1 Organización social y de trabajo de los seres humanos
- 4.2 Áreas de aplicación
- 4.3 Ergonomía

Tema 5.- Trabajo en red

- 5.1 Sistema cliente-servidor
- 5.2 Desarrollo de aplicaciones web
- 5.3 Computación sin cables y móvil
- 5.4 Trabajo cooperativo

Unidad didáctica III: Desarrollo de software centrado en la persona

Tema 6.- Metáforas

- 6.1 Metáforas
- 6.2 Metáfora del escritorio
- 6.3 Metodología de creación de metáforas

6.4 Lenguaje visual para el diseño de metáforas

6.5 Ejemplos de metáforas

Tema 7.- Estilos de interacción

7.1 ¿Qué es la interacción?

7.2 Estilos de Interacción

7.3 Paradigmas

7.4 Comparación de los paradigmas de interacción

Tema 8.- Gestión de la información

8.1 Hipertexto e Hipermedia.

8.2 Sistemas e información multimedia.

8.3 Ayuda y documentación

8.4 Tecnologías de datos multimedia

Unidad didáctica IV: Diseño de interfaces efectivas

Tema 9.- Estándares y guías

9.1 Principios y directrices

9.2 Estándares

9.3 Estándares de *iure* en IPO

9.4 Guías de estilo

9.5 Guías de estilo corporativas

9.6 Consideraciones sobre los estándares y las guías de estilo

Tema 10.- El diseño gráfico

10.1 La comunicación y los nuevos “media”

10.2 La experiencia psico-perceptual y el entendimiento humano

10.3 El sistema gráfico: la imagen

10.4 El presente del diseño aplicado a nuevos entornos

Tema 11.- Ingeniería de la interfaz

11.1 Análisis centrado en el usuario

11.2 Ciclo de vida de la interfaz de usuario

11.3 Aproximaciones al diseño

11.4 Análisis de tareas

11.5 Modelos arquitectónicos

- 11.6 Modelos abstractos
- 11.7 Estrategias de diseño

Unidad didáctica V: Usabilidad

Tema 12.- Accesibilidad

- 12.1 La importancia del diseño universal
- 12.2 ¿Que es el diseño universal?
- 12.3 Tipos de discapacidades y soluciones
- 12.4 Accesibilidad en la web
- 12.5 Comprobación de la accesibilidad

Tema 13.- Diseño y Evaluación

- 13.1 El diseño centrado en el usuario
- 13.2 La usabilidad
- 13.3 Prototipado
- 13.4 ¿Que es la evaluación?
- 13.5 Métodos de evaluación
- 13.6 Coste de la usabilidad
- 13.7 Laboratorio de usabilidad

Tema 14.- Diálogo persona-ordenador

- 14.1 Interacción usando el lenguaje
- 14.2 Realidad virtual
- 14.3 Interfaces hápticas
- 14.4 Realidad aumentada
- 14.5 Pantallas táctiles
- 14.6 Rastreo ocular
- 14.7 Ordenadores corporales

Tema 15.- Internacionalización

- 15.1 Internacionalización y localización
- 15.2 Elementos de la interfaz
- 15.3 Lenguajes
- 15.4 Zonas de internacionalización
- 15.5 Metodología de trabajo

6. Metodología y estrategias de aprendizaje

6.1. Metodología docente

El proceso de aprendizaje significativo exige que se produzca una conexión entre los conocimientos que tiene el alumno y los contenidos nuevos, de forma que la estructura o el esquema previo que el estudiante tiene se vea enriquecido, modificado y perfeccionado con las nuevas aportaciones. Aprender no es, por tanto, adquirir datos, conceptos o hechos aislados, sino adquirir esquemas y estructuras cada vez más amplios y ricos. Para modificar los esquemas de conocimiento del alumno, es necesario:

- Analizar su estructura cognitiva en el momento de entrar en contacto con unos nuevos contenidos. Es punto de partida imprescindible el diagnóstico previo, que el profesor llevará a cabo valiéndose de su experiencia docente, mediante la observación en el aula o con la formulación de preguntas concretas a los alumnos.
- Crear una cierta contradicción entre los conocimientos previos y los nuevos. Ese “desequilibrio” debe concluir en un “equilibrio”, un nuevo esquema más amplio y perfeccionado que el anterior.

En este contexto, el objetivo más ambicioso de la enseñanza es que el alumno aprenda por sí solo, es decir que *aprenda a aprender*, requisito imprescindible de cara a cualquier nuevo plan de estudios que se defina dentro del EEES. Por ello se ha habilitado un curso virtual en el cual el alumno tiene a su disposición todos los recursos de la asignatura y acceso directo a los docentes de la asignatura, que pueden resolverle mediante una “pizarra virtual” sus dudas en cualquier momento. Además se disponen de foros con los que los alumnos pueden expresar sus experiencias sobre la asignatura y plantear temas nuevos, enlaces a lecturas complementaras, ejercicios de autoevaluación, etc.

En la asignatura de Hipermedia el trabajo a realizar se va a centrar no sólo en la tradicional clase magistral o las prácticas de laboratorio, sino que éste estará complementado con sesiones de tutorías y seminarios, así como por los trabajos personales y en grupo de los propios alumnos, como se muestra en la Figura 4.

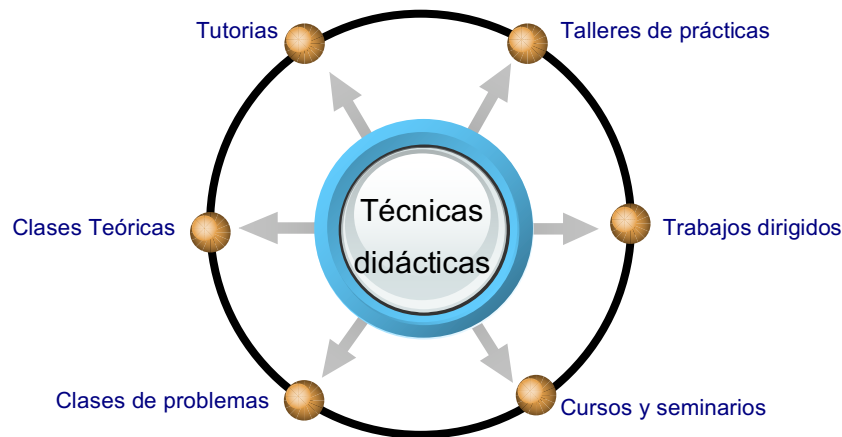


Figura 4. Metodología docente a utilizar.

Las actividades que se proponen son las siguientes:

- Clases de teoría con la ayuda de presentaciones multimedia. En ellas se muestran los contenidos básicos de cada tema, estructurados de la siguiente manera:
 - o Índice e introducción de los contenidos que se tratarán en la clase.
 - o Ubicación en el temario de la asignatura y enlaces con el resto de temas.
 - o Exposición de objetivos.
 - o Desarrollo del tema.
 - o Resumen.
 - o Referencias básicas y complementarias.

Como apoyo a las clases de teoría los alumnos dispondrán de una copia de las transparencias en formato electrónico y/o en formato papel, además de los apuntes de cada uno de los temas.

- Talleres de prácticas. En ellas se examinan supuestos prácticos extraídos del mundo real, analizando (a modo de debate) el funcionamiento de los esquemas tratados, buscando alternativas y realizando un informe de las posibles mejoras aplicables. Hay que tener en cuenta que ésta es una asignatura muy cambiante y que está muy a expensas de los avances tecnológicos, por lo cual la actualización de la misma ha de ser muy rápida.
- Clases de problemas. Se resolverán algunos de los ejercicios propuestos en la asignatura, los más representativos de cara al aprendizaje del alumno.
- Trabajos dirigidos. Los alumnos deberán realizar y defender una o varias prácticas relacionadas con el diseño de interfaces y el manejo de computadores y máquinas a través de servicios web.

- Tutorías. El alumnado tiene a su disposición seis horas de tutorías a la semana en las que puede consultar cualquier duda relacionada con los contenidos, organización y planificación de la asignatura. Además se realizarán tutorías grupales para resolver problemas relacionados con las actividades a realizar en grupo.
- Curso virtual. Es la parte más dinámica para la comunicación y almacenamiento de información de la asignatura, además de tener la ventaja de que es algo que se construye entre todos: docentes y alumnos.

6.2. Estrategias de aprendizaje

En los siguientes apartados se indican las fases a realizar para la aplicación y aprovechamiento de la metodología propuesta.

6.2.1. Documentación de la asignatura

- Los alumnos dispondrán en el curso virtual de la asignatura toda la información y recursos relacionados con la misma: temario, bibliografía, apuntes, programas útiles, enunciados de prácticas, trabajos, enlaces de interés, lecturas complementarias, avisos, criterios de evaluación, etc.
- Los docentes mantienen actualizada la información de este curso para que se convierta en un vehículo de comunicación con los alumnos.
- Los alumnos deben conocer y manejar con fluidez toda la información y la documentación que se integra en el curso virtual de la asignatura.
- El alumno debe consultar con frecuencia el curso virtual para estar actualizado.

6.2.2. Planificación de las clases teóricas

Las clases teóricas se dedicarán a la presentación de contenidos y a la discusión y resolución de las dudas que se puedan plantear durante las exposiciones.

La tarea de los docentes ante estas clases teóricas abarca tres puntos diferenciados:

- Preparación de la clase (contenido).
- Presentación y desarrollo de la clase (forma externa).
- Puesta en juego de recursos para la inserción activa del alumno en clase.

Respecto a la preparación de la clase, el contenido de la misma debe cumplir unas condiciones para que sea eficaz su captación y aprendizaje por parte del alumno. Dichas condiciones, básicamente, son las siguientes:

- Adecuación. La lección debe adaptarse a las posibilidades intelectuales, capacidad y preparación de base del alumno.
- Claridad de ideas. Para que una idea sea captada, ésta debe ser clara.
- Secuencialidad. Las ideas claras deben enlazarse para exponer coherentemente.
- Selección. Debe destacarse lo fundamental del contenido de las lecciones, evitando desviaciones o divagaciones secundarias.
- Ritmo. Debe forzarse el contenido de una clase, de manera que no obligue a adoptar un ritmo excesivamente rápido que los alumnos no puedan seguir.

En cuanto a la presentación y desarrollo de la clase, debe estructurarse en tres fases:

- Introducción. En esta fase se dispone a los alumnos para el aprendizaje de la lección. Debe esquematizarse brevemente el contenido de la clase a desarrollar, relacionándolo con los temas expuestos anteriormente y estableciendo el papel que, dentro de la asignatura, juega el tema a tratar. Deben, igualmente, presentarse los objetivos específicos que se persiguen alcanzar con la lección.
- Desarrollo. La fase de desarrollo constituye el cuerpo de la lección donde, además del contenido, deben estar presentes las técnicas de motivación. Debe desarrollarse la lección de forma clara, precisa y progresiva, con secuencias lógicas de contenidos y reforzando la exposición, caso de que el tema se preste a ello, con empleo de medios audiovisuales.
Durante el desarrollo de las clases, el profesor debe promover la participación activa del alumno, bien planteando directamente cuestiones que provoquen el comentario de los alumnos, bien, cuando se trate de un tema particularmente difícil, anticipando el contenido de dicho tema, así como la bibliografía pertinente, de modo que el alumno llegue a la clase con unas ideas previas y con los puntos oscuros localizados, lo cual inducirá al alumno a concentrar más su atención y a plantear preguntas con mayor conocimiento de causa.
- Conclusión. La clase debe finalizar con un resumen de los conceptos fundamentales asegurándonos que han sido captados. Deben proporcionarse a los alumnos las necesarias referencias complementarias de información, y debe anticiparse un marco de referencia de la próxima lección.

Sobre la puesta en juego de recursos para la inserción activa del alumno en clase, deben emplearse técnicas expositivas y utilizarse los recursos necesarios para tal fin. Es necesario estimular y orientar el interés de los alumnos en su trabajo intelectual y mantener ese interés cada día del curso.

Por su parte el alumno:

- Debe realizar una lectura previa de los contenidos que se van a tratar en la clase, teniendo en cuenta que no todos los contenidos se desarrollarán en profundidad.
- Una vez concluida la exposición teórica, ha de estudiar de forma autónoma su contenido y en caso de no entender algo intentar aclararlo utilizando la bibliografía recomendada o consultar a los profesores.
- Es conveniente también que profundice en algunos de los temas y busque y aporte nueva información.

6.2.3. Planificación de las clases prácticas

Las clases prácticas se dividen a su vez en dos tipos: clases de problemas y talleres de prácticas, cómo se indicó en el punto 6.1.

- Para las clases de problemas se sigue la estructura mostrada en la Figura 5. Primero se indica el contenido teórico en el cual se encuadra el ejercicio. Después se resuelven una serie de problemas básicos y por último una serie de problemas complementarios.
- En el caso de los talleres de prácticas, los distintos grupos de trabajo debe reunirse para leer y comprender bien el enunciado y realizar sus críticas y propuestas de solución. Los moderadores de estos talleres serán los profesores de la asignatura.

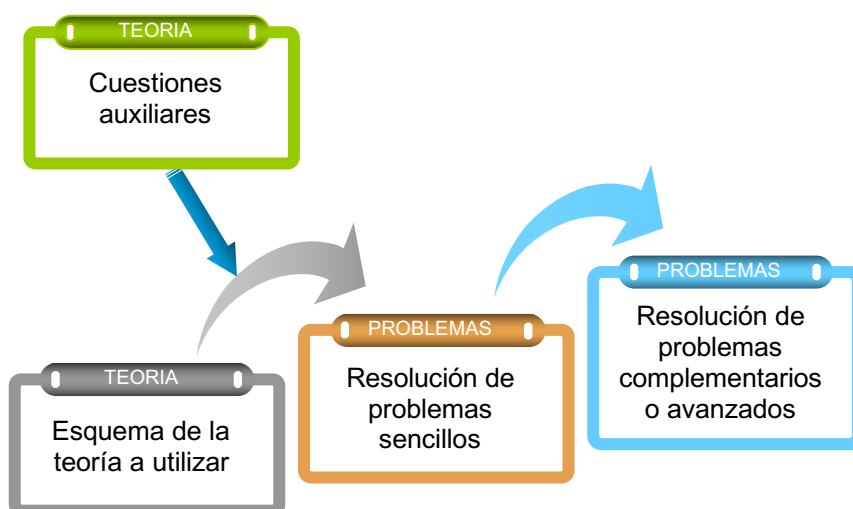


Figura 5. Esquema de las clases de problemas.

Respecto a las obligaciones del alumno:

- Cumplirán, sin excepciones, el calendario de entrega de las prácticas.
- Los informes de los talleres quedarán publicados en el curso virtual de la asignatura.
- Las prácticas obligatorias se defenderán en grupo. Cada grupo de trabajo debe leer y entender bien todos los puntos antes de realizarla.

6.2.4. Evaluación

- Se plantea una forma de evaluación continua.
- Para aquellos alumnos que no superen la evaluación continua o no quieran o no puedan acogerse a ella existirá una prueba final.

7. Planificación del tiempo y del esfuerzo

La asignatura de Hipermedia tiene en el plan de estudios actual 7,5 créditos LRU asignados, 3 créditos de teoría y 4,5 créditos de práctica. Para hacer su estudio en ECTS se ha tomado tomando la equivalencia de crédito LRU a crédito ECTS, con implicación de cambio de metodología docente.

Al hacer el estudio de carga de trabajo máxima, se tiene que:

- 7,5 ECTS → 187,5-225 horas de trabajo
- 3cr LRU → 30 horas presenciales + (30·1,5) horas de asimilación = 75 h.
- 4,5cr LRU → 45 horas presenciales + (45·2) horas de asimilación = 135 h.
- 75 + 135 = 210 horas de trabajo
- 187,5 < 210 < 225

Dos son los escenarios más probables: el resultante de tomar el crédito ECTS como 30 horas de trabajo y el resultante de tomarlo como 25 horas de trabajo. Por temas de organización y calendario, se ha optado por partir del escenario que asigna 25 horas de trabajo al ECTS.

Para poder elaborar las guías, cuyo destino final sería la realización de experiencias piloto, dentro de los planes de estudio vigentes, se debe tener en cuenta la definición del crédito del R.D. 1497/1987 y sus posteriores modificaciones. Así:

- La unidad de valoración de las enseñanzas se corresponde a diez horas de enseñanza teórica, práctica o de sus equivalencias entre las que podrán incluirse actividades académicas dirigidas, que habrán de preverse en el correspondiente plan docente junto con los mecanismos y medios objetivos de comprobación de los resultados académicos de las mismas. Todo ello sin perjuicio del cumplimiento del régimen de dedicación del profesorado, de conformidad con el Real Decreto 898/1985, de 30 de abril sobre régimen del profesorado universitario.
- En ningún caso, salvo que se trate de enseñanzas en Universidades a distancia, el porcentaje del crédito correspondiente a las actividades académicas dirigidas será superior al **30%**.

Tabla 3. Escenario ECTS = 25 horas de trabajo.

			A	B	C	D	E
	Técnica	Actividad	Horas equivalentes de clase	Factor de trabajo del alumno ¹⁵	Horas de trabajo personal del alumno	Horas totales (A+C)	ECTS (D ÷ 25)
Teoría	Seguimiento de la parte teórica de las	Asimila contenidos. Se plantea dudas que	20	1,5	30	50	2,00

¹⁵ Número de horas dedicadas por el alumno al trabajo personal (organización de apuntes, estudio, documentación...) por cada hora de clase.

	unidades didácticas	preguntará a los profesores en las tutorías					
Práctica	Seguimiento de la parte práctica de las unidades didácticas	Experimenta, practica, modela, simula	30	2	60	90	3,60
Ejercicios de apoyo	Ejercicios de repaso y afianzamiento de lo aprendido. Pueden computar o no para la superación de la asignatura, dependiendo de la tipología de éstos	Realiza, resuelve problemas, tests...	-	-	5	5	0,20
Prácticas obligatorias	Prácticas en grupo de obligada realización	Realiza un supuesto de un caso real	2	-	26	28	1,12
Examen	Controles de evaluación continua y/o examen final de la asignatura	Exámenes tipo test, supuestos prácticos...	2,5			2,5	0,10
Otras actividades	Tutorías personalizadas y grupales	Recibe orientación personalizada	2	-	-	2	0,08
	Búsquedas en la red, participación en foros especializados...	Busca elementos para completar los contenidos	-	-	10	10	0,40
TOTAL			56,5		131	187,5	7,50

Tabla 4. Reparto de horas presenciales.

NÚMERO DE HORAS PRESENCIALES		
Actividad	Horas Teoría	Horas Prácticas
Unidad Didáctica I: La interacción persona-ordenador		
Tema 1. Introducción a la interacción persona-ordenador (IPO)	1	
Tema 2. Dispositivos de interacción	1	
Unidad Didáctica II: La persona y su entorno de trabajo		
Tema 3. El factor humano	1	
Tema 4. Uso y contexto de los computadores	1,5	
Tema 5. Trabajo en red	1	6
Unidad Didáctica III: Desarrollo de software centrado en la persona		
Tema 6. Metáforas	1,5	
Tema 7. Estilos de interacción	1	
Tema 8. Gestión de la información	2	4
Unidad Didáctica IV: Diseño de interfaces efectivas		
Tema 9. Estándares y guías	1	
Tema 10. El diseño gráfico	2	6
Tema 11. Ingeniería de la interfaz	2	8
Unidad Didáctica V: Usabilidad		
Tema 12. Accesibilidad	2	6

Tema 13. Diseño y Evaluación	1	
Tema 14. Diálogo persona-ordenador	1	
Tema 15. Internacionalización	1	
Práctica obligatoria		2
Tutorías grupales		2
Examen	2,5	
TOTAL: 45	22,5	34

8. Bibliografía

8.1. Bibliografía básica

Abascal, J., Cañas, J., Gea, M., Gil, A. B., Lorés, J., Martínez, A. B., Ortega, M.; Valero, P., Vélez, M. (2002) *Introducción a la Interacción Persona-Ordenador*. Jesús Lorés (Editor). [<http://griho.udl.es/ipo/libroe.html>]

Dix, A., Finlay, J., Abowd G., Beale R. (1998) *Human Computer Interaction*. 2º edición. Prentice Hall.

Kriug, S. *No me hagas pensar .Una aproximación a la usabilidad en la web*. 2º edición. Prentice Hall.

Mandel, T. (1997) *The Elements of User Interface Design*. John Wiley & Sons.

Morrison, M. (2000) *XML al descubierto*. Prentice Hall.

Norman, D. (1990) *The design of everyday things*. Doubleday.

Phillips, L. A. (2000) *Descubre HTML*. Prentice Hall.

Rodríguez de la Fuente, S, Pérez Costoya, F., Carretero Pérez, J. y otros. (2003) *Programación de aplicaciones Web*. Thomson-Paraninfo.

Shneiderman, B., Plaisant, C. (2005) *Diseño de interfaces de usuario. Estrategias para una interacción persona-computadora efectiva*. Pearson Prentice-Hall 2005.

8.2. Bibliografía complementaria

Asleson, R., Schutta, N. (2005) *Foundations of Ajax*. Apress.

Diaper, D., Stanton, N. *The Handbook of Task Analysis for Human Computer Interaction*. (2004) Lawrence Erlbaum Associates, publishers.

Ghaoui, C. (2005) *Encyclopaedia of HCI*. Idea Group Reference.

Glass, M. y otros (2004). *Beginning PHP, Apache, MySQL Web Development*. John Wiley and Sons.

Gray, N. (2003) *Web Server Programming*. 2º Edición. John Wiley and Sons.

Hall, M., Brown, R. (2001) *Core Web Programming*. 2º Edición. Prentice Hall.

Holtzblatt, K., Burns, J., Wood, S. (1998) *Contextual Design. A How-to Guide to Key Techniques for User- Centered Design*. Morgan Kaufmann – Elsevier.

Raskin, J. (2003) *Diseño de sistemas interactivos*. 4º Edición. Addison Wesley.

Veen, J. (2001) *The Art & Science of Web Design*. New Riders.

8.3. Otros recursos

Apple Computer, Inc. (1993) *Macintosh Human Interface Guidelines*. Addison-Wesley.

Microsoft (1992). *The Windows Interface: An Application Design Guide*. Microsoft Windows Software Development Kit (SDK).

NeXT Computer, Inc (1992). *NeXTSTEP User Interface Guidelines: Release 3*. Addison-Wesley.

Petzold, C. (1993). *Programming Windows 3.1: Microsoft guide to writing applications for Windows 3.1*. Microsoft Press.

Sun Microsystems, Inc. (1990) *Open Look. Vol 1: GUI Application Style Guide*. Addison-Wesley.

Sun Microsystems, Inc. (1990) *Open Look. Vol 2: GUI Functional Specification*. Addison-Wesley.

8.4. Enlaces de interés

- **Advanced Common Senses**
 - <http://www.sensible.com/>

- **Asociación Interacción Persona-Ordenador**

- <http://griho.udl.es/aipo/>.
- **CHISPA - ACM SIGCHI en España**
 - <http://petra.euitio.uniovi.es/acm/sigchi/>
- **Computing Curricula**
 - <http://acm.org/education/curricula.html>
- **Curso de Introducción a la interacción persona-ordenador**
 - <http://griho.udl.es/ipo/>
- **Manual de diseño digital: diseño gráfico, diseño web, tipografía, creatividad**
 - <http://platea.cnice.mecd.es/~jmas/manual/html/intro.html>
- **Raskin Center for Humane Interfaces (RCHI)**
 - <http://www.raskincenter.org/index.html>
- **SIGCHI Curricula for Human-Computer Interaction**
 - <http://www.acm.org/sigchi/cdg/>
- **Usable Web**
 - <http://usableweb.com/>

8.5. Herramientas software

- **Amaya**
 - <http://www.w3.org/Amaya/Overview.html>
- **BEA | M7 product jump page**
 - [http://workshopstudio.bea.com/index.html"](http://workshopstudio.bea.com/index.html)
- **Instant Rails**
 - http://instantrails.rubyforge.org/wiki/wiki.pl?Instant_Rails

- **Nvu - The Complete Web Authoring System for Linux, Macintosh and Windows**
 - <http://www.nvu.com/>

- **WYSIWYG Web Builder**
 - <http://www.wysiwygwebbuilder.com/>

9. Evaluación de los procesos y resultados de aprendizaje

9.1. Sistema de evaluación

La evaluación de esta asignatura se divide en dos partes diferenciadas:

- **Parte de Teoría** (30% de la nota final).
 - Examen final que consta de una serie de preguntas cortas sobre el temario de la asignatura.
 - La parte teórica se guardará durante el curso académico actual.
- **Parte Práctica** (70% de la nota final).
 - Práctica o prácticas realizadas en grupos de trabajo.
 - Se realizará una defensa de dicho trabajo.
 - La parte práctica se guardará durante el curso académico actual.

De forma complementaria a este sistema de evaluación tradicional, el alumno se puede acoger opcionalmente a una evaluación continua.

- Se tendrá en cuenta la asistencia y la participación activa en clase.
- En relación con la parte de teoría:
 - Se realizarán trabajos que cubran parte de los contenidos de cada unidad didáctica.
 - Se eliminará la parte correspondiente del examen de teoría si se obtiene una calificación superior o igual a 5 en el conjunto de estas pruebas, obteniendo en todas ellas al menos una nota de 3.
- En relación con la parte práctica:
 - Todos los grupos obligatoriamente entregan el trabajo realizado al final de cada una de las sesiones (impreso y debidamente formateado), con los ficheros fuente del programa realizado (si los hubiese).
 - La participación activa, acertada y continuada puede aportar hasta 1 punto en la nota final.
 - En el caso de que se produzca un fraude o copia en los trabajos, se tomarán las medidas oportunas.

9.2. Criterios de evaluación

La calificación se hará conforme a las siguientes pautas:

Matrícula de honor

- La nota final es superior o igual a 9,75 puntos
- El conocimiento y la comprensión de la materia se extienden más allá del trabajo cubierto por el programa.
- La comprensión conceptual es sobresaliente.
- Los problemas relacionados con la asignatura son resueltos con soltura y se es capaz de razonar alternativas de solución estableciendo comparativas entre ellas.
- La participación en las diferentes actividades, incluyendo las clases, sobresale por su corrección y satisfacción.

Sobresaliente

- La nota final es superior o igual a 8,5 puntos
- El conocimiento y la comprensión de la materia son muy satisfactorios.
- La comprensión conceptual es sobresaliente.
- Los problemas relacionados con la asignatura son resueltos con soltura y se es capaz de razonar alternativas de solución.
- La participación en las diferentes actividades, incluyendo las clases, ha sido muy correcta y muy satisfactoria.

Notable

- La nota final es superior o igual a 6,75 puntos
- El conocimiento y la comprensión de la materia son satisfactorios.
- La comprensión conceptual es notable.
- Los problemas relacionados con la asignatura son resueltos con soltura.
- La participación en las diferentes actividades, incluyendo las clases, ha sido correcta y bastante satisfactoria.

Aprobado

- La nota final es superior o igual a 5 puntos
- El conocimiento y la comprensión de la materia son básicos.
- La comprensión conceptual es suficiente.
- Los problemas relacionados con la asignatura son resueltos con éxito razonable.

- La participación en las diferentes actividades, incluyendo las clases, ha sido correcta y pero no siempre satisfactoria.

Suspenso

- La nota final es inferior a 5 puntos
- El conocimiento y la comprensión de la materia no son aceptables.
- La comprensión conceptual es insuficiente.
- Los problemas relacionados con la asignatura no se resuelven de forma adecuada.
- La participación en las diferentes actividades, incluyendo las clases, ha sido escasa y deficiente.

10. Análisis de coherencia de la Guía Docente

Con este análisis se resumen las relaciones existentes entre objetivos y competencias a desarrollar con el plan de trabajo del alumnado, así como el sistema de evaluación empleado.

Tabla 5. Relaciones entre los objetivos y las competencias instrumentales.

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS		BLOQUES DE CONTENIDOS					PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Instrumentales		Unidad didáctica I	Unidad didáctica II	Unidad didáctica III	Unidad didáctica IV	Unidad didáctica V		
OI1 OI11 OI12 OI16	CIC3 CIC6 CIC11 CIC21 CIC58	-	4	6 8	9 10 11	12	Enseñanza presencial (Clases magistrales/Talleres de prácticas/Defensa de práctica) Enseñanza no presencial (Trabajo en grupos/Realización de prácticas obligatorias. Revisión bibliográfica/Consulta de recursos en Internet) Tutorías	Procedimientos Examen Práctica obligatoria Criterios Grado de comprensión y aplicación del concepto de interfaz.	
OI2 OI3 OI4	CIC7 CIC9 CIC10	1 2	3 4 5	6 7 8	9 10 11	12 13 14 15	Enseñanza presencial (Clases magistrales/Talleres de prácticas/Defensa de práctica) Enseñanza no presencial (Trabajo en grupos/Realización de prácticas obligatorias. Revisión bibliográfica/Consulta de recursos en Internet) Tutorías	Procedimientos Defensas de prácticas en grupo Criterios Grado de comprensión y aplicación de los conceptos básicos de la Hipermedia.	
OI5 OI7	CIC4 CIC5 CIC8	1 2	3 4 5	6 7 8	9 10 11	12 13 14 15	Enseñanza presencial (Clases magistrales/Talleres de prácticas/Defensa de práctica) Enseñanza no presencial (Trabajo en grupos/Realización de prácticas obligatorias. Revisión bibliográfica/Consulta de recursos en Internet) Tutorías	Procedimientos Tutorías Conferencias invitadas Criterios Detectar el conocimiento y visión que tienen del mercado laboral.	

OI6	CIC1 CIC2	-	5	6 8	9 10 11	12 13	Enseñanza presencial (Clases magistrales/Talleres de prácticas/Defensa de práctica) Enseñanza no presencial (Trabajo en grupos/Realización de prácticas obligatorias. Revisión bibliográfica/Consulta de recursos en Internet) Tutorías	Procedimientos Actividades en grupo Criterios Grado de destreza en trabajos en grupo. Evaluación del trabajo en equipo.
OI8 OI13	CIC37-52 CIC57-61	2	3	6 7 8	11	12 13 14 15	Enseñanza presencial (Clases magistrales/Talleres de prácticas/Defensa de práctica) Enseñanza no presencial (Trabajo en grupos/Realización de prácticas obligatorias. Revisión bibliográfica/Consulta de recursos en Internet) Tutorías	Procedimientos Examen Práctica obligatoria Criterios Grado de comprensión y aplicación del concepto de accesibilidad.
OI9 OI10 OI14	CIC12-20 CIC22-36	1	3 4	6 7 8	9 10 11	12 13	Enseñanza presencial (Clases magistrales/Talleres de prácticas/Defensa de práctica) Enseñanza no presencial (Trabajo en grupos/Realización de prácticas obligatorias. Revisión bibliográfica/Consulta de recursos en Internet) Tutorías	Procedimientos Examen Práctica obligatoria Criterios Grado de comprensión y aplicación del concepto de software centrado en la persona.
OI15	CIC53-56	2	3 4	-	-	14	Enseñanza presencial (Clases magistrales/Talleres de prácticas/Defensa de práctica) Enseñanza no presencial (Trabajo en grupos/Realización de prácticas obligatorias. Revisión bibliográfica/Consulta de recursos en Internet) Tutorías	Procedimientos Ejercicios Examen Talleres Práctica obligatoria Criterios Grado de comprensión y aplicación de un método estructurado.
Elementos transversales	CIM1-CIM6 CIT1-CIT4 CIL1-CIL2	1 2	3 4 5	6 7 8	9 10 11	12 13 14 15	Enseñanza presencial (Clases magistrales/Talleres de prácticas/Defensa de práctica) Enseñanza no presencial (Trabajo en grupos/Realización de prácticas obligatorias. Revisión bibliográfica/Consulta de recursos en Internet) Tutorías	Procedimientos Ejercicios Examen Talleres Práctica obligatoria Criterios Grado de destreza en la competencias transversales

Tabla 6. Relaciones entre los objetivos y las competencias interpersonales.

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS		BLOQUES DE CONTENIDOS					PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Interpersonales		Unidad didáctica I	Unidad didáctica II	Unidad didáctica III	Unidad didáctica IV	Unidad didáctica V		
OIP1	CIPTC1	1 2	3 4 5	6 7 8	9 10 11	12 13 14 15	Enseñanza presencial (Talleres de prácticas/Defensa de prácticas)	Procedimientos Actividades en grupo Criterios Grado de destreza en trabajos en grupo	

OIP2	CIPTC2 CIPTC3	1 2	3 4 5	6 7 8	9 10 11	12 13 14 15	Enseñanza presencial (Talleres de prácticas/Defensa de prácticas)	Procedimientos Actividades en grupo Criterios Grado de destreza en trabajos en grupo
OIP3	CIPTR1 CIPTR2 CIPTR3 CIPTR4	1 2	3 4 5	6 7 8	9 10 11	12 13 14 15		Enseñanza presencial (Talleres de prácticas/Defensa de prácticas)

Tabla 7. Relaciones entre los objetivos y las competencias sistémicas.

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS		BLOQUES DE CONTENIDOS					PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Sistémicas		Unidad didáctica I	Unidad didáctica II	Unidad didáctica III	Unidad didáctica IV	Unidad didáctica V		
OS1	CS1	1 2	3 4 5	6 7 8	9 10 11	12 13 14 15	Enseñanza presencial (Talleres de prácticas/Defensa de práctica/Exposición de ejercicios) Enseñanza no presencial (Realización de ejercicios/Revisión bibliográfica/Consulta de recursos en Internet) Tutorías	Procedimientos Prácticas Trabajos en grupo Ejercicios Criterios Grado de análisis y evaluación de los procedimientos relacionados con la resolución de problemas	
OS2 OS3 OS5	CS2 CS5	1 2	3 4 5	6 7 8	9 10 11	12 13 14 15	Enseñanza presencial (Talleres de prácticas/Defensa de práctica/Exposición de ejercicios) Enseñanza no presencial (Realización de ejercicios/Revisión bibliográfica/Consulta de recursos en Internet) Tutorías	Procedimientos Prácticas Trabajos en grupo Ejercicios Criterios Grado de análisis y evaluación de los procedimientos relacionados con la resolución de problemas	
OS4 OS6	CS3 CS4	1 2	3 4 5	6 7 8	9 10 11	12 13 14 15	Enseñanza presencial (Talleres de prácticas/Defensa de práctica/Exposición de ejercicios) Enseñanza no presencial (Realización de ejercicios/Revisión bibliográfica/Consulta de recursos en Internet) Tutorías	Procedimientos Prácticas Trabajos en grupo Ejercicios Criterios Grado de análisis y evaluación de los procedimientos relacionados con la resolución de problemas	

ANEXO

En este anexo se muestran las tablas del *Computing Curricula 2005* (CC 2005), que muestran las capacidades relativas alcanzadas por los titulados en Informática y la influencia de las materias en los cinco perfiles profesionales.

Tabla 8. *Capacidades relativas alcanzadas por los titulados en Informática, según el CC 2005.*

Area	Performance Capability	CE	CS	IS	IT	SE	
Algorithms	Prove theoretical results	3	5	1	0	3	
	Develop solutions to programming problems	3	5	1	1	3	
	Develop proof-of-concept programs	3	5	3	1	3	
	Determine if faster solutions possible	3	5	1	1	3	
Application programs	Design a word processor program	3	4	1	0	4	
	Use word processor features well	3	3	5	5	3	
	Train and support word processor users	2	2	4	5	2	
	Design a spreadsheet program (e.g., Excel)	3	4	1	0	4	
	Use spreadsheet features well	2	2	5	5	3	
Application programs	Train and support spreadsheet users	2	2	4	5	2	
	Computer programming	Do small-scale programming	5	5	3	3	5
Computer programming	Do large-scale programming	3	4	2	2	5	
	Do systems programming	4	4	1	1	4	
	Develop new software systems	3	4	3	1	5	
	Create safety-critical systems	4	3	0	0	5	
	Manage safety-critical projects	3	2	0	0	5	
	Hardware and devices	Design embedded systems	5	1	0	0	1
		Implement embedded systems	5	2	1	1	3
Design computer peripherals		5	1	0	0	1	
Design complex sensor systems		5	1	0	0	1	
Design a chip		5	1	0	0	1	
Program a chip		5	1	0	0	1	
Design a computer		5	1	0	0	1	
Human-computer interface	Create a software user interface	3	4	4	5	4	
	Produce graphics or game software	2	5	0	0	5	
	Design a human-friendly device	4	2	0	1	3	
Information systems	Define information system requirements	2	2	5	3	4	
	Design information systems	2	3	5	3	3	
	Implement information systems	3	3	4	3	5	
	Train users to use information systems	1	1	4	5	1	
	Maintain and modify information systems	3	3	5	4	3	
Information management (Database)	Design a database mgt system (e.g., Oracle)	2	5	1	0	4	
	Model and design a database	2	2	5	5	2	
	Implement information retrieval software	1	5	3	3	4	
	Select database products	1	3	5	5	3	
	Configure database products	1	2	5	5	2	
	Manage databases	1	2	5	5	2	
	Train and support database users	2	2	5	5	2	
IT resource planning	Develop corporate information plan	0	0	5	3	0	
	Develop computer resource plan	2	2	5	5	2	
	Schedule/budget resource upgrades	2	2	5	5	2	
	Install/upgrade computers	4	3	3	5	3	
	Install/upgrade computer software	3	3	3	5	3	
Intelligent systems	Design auto-reasoning systems	2	4	0	0	2	
	Implement intelligent systems	2	4	0	0	4	
Networking and communications	Design network configuration	3	3	3	4	2	
	Select network components	2	2	4	5	2	
	Install computer network	2	1	3	5	2	
	Manage computer networks	3	3	3	5	3	
	Implement communication software	5	4	1	1	4	
	Manage communication resources	1	0	3	5	0	
	Implement mobile computing system	5	3	0	1	3	
	Manage mobile computing resources	3	2	2	4	2	
Systems Development Through Integration	Manage an organization's web presence	2	2	4	5	2	
	Configure & integrate e-commerce software	2	3	4	5	4	
	Develop multimedia solutions	2	3	4	5	3	
	Configure & integrate e-learning systems	1	2	5	5	3	
	Develop business solutions	1	2	5	3	2	
Systems Development Through Integration	Evaluate new forms of search engine	2	4	4	4	4	

Tabla 9. Comparativa de la influencia de las materias en los cinco perfiles profesionales del CC 2005.

Knowledge Area	GE		CS		IS		IT		SE	
	min	max	min	max	min	max	min	max	min	max
Programming Fundamentals	4	4	4	5	2	4	2	4	5	5
Integrative Programming	0	2	1	3	2	4	3	5	1	3
Algorithms and Complexity	2	4	4	5	1	2	1	2	3	4
Computer Architecture and Organization	5	5	2	4	1	2	1	2	2	4
Operating Systems Principles & Design	2	5	3	5	1	1	1	2	3	4
Operating Systems Configuration & Use	2	3	2	4	2	3	3	5	2	4
Net Centric Principles and Design	1	3	2	4	1	3	3	4	2	4
Net Centric Use and configuration	1	2	2	3	2	4	4	5	2	3
Platform technologies	0	1	0	2	1	3	2	4	0	3
Theory of Programming Languages	1	2	3	5	0	1	0	1	2	4
Human-Computer Interaction	2	5	2	4	2	5	4	5	3	5
Graphics and Visualization	1	3	1	5	1	1	0	1	1	3
Intelligent Systems (AI)	1	3	2	5	1	1	0	0	0	0
Information Management (DB) Theory	1	3	2	5	1	3	1	1	2	5
Information Management (DB) Practice	1	2	1	4	4	5	3	4	1	4
Scientific computing (Numerical mthds)	0	2	0	5	0	0	0	0	0	0
Legal / Professional / Ethics / Society	2	5	2	4	2	5	2	4	2	5
Information Systems Development	0	2	0	2	5	5	1	3	2	4
Analysis of Business Requirements	0	1	0	1	5	5	1	2	1	3
E-business	0	0	0	0	4	5	1	2	0	3
Analysis of Technical Requirements	2	5	2	4	2	4	3	5	3	5
Engineering Foundations for SW	1	2	1	2	1	1	0	0	2	5
Engineering Economics for SW	1	3	0	1	1	2	0	1	2	3
Software Modeling and Analysis	1	3	2	3	3	3	1	3	4	5
Software Design	2	4	3	5	1	3	1	2	5	5
Software Verification and Validation	1	3	1	2	1	2	1	2	4	5
Software Evolution (maintenance)	1	3	1	1	1	2	1	2	2	4
Software Process	1	1	1	2	1	2	1	1	2	5
Software Quality	1	2	1	2	1	2	1	2	2	4
Comp Systems Engineering	5	5	1	2	0	0	0	0	2	3
Digital logic	5	5	2	3	1	1	1	1	0	3
Embedded Systems	2	5	0	3	0	0	0	1	0	4
Distributed Systems	3	5	1	3	2	4	1	3	2	4
Security: issues and principles	2	3	1	4	2	3	1	3	1	3
Security: implementation and mgt	1	2	1	3	1	3	3	5	1	3
Systems administration	1	2	1	1	1	3	3	5	1	2
Management of Info Systems Organ'tion	0	0	0	0	3	5	0	0	0	0
Systems integration	1	4	1	2	1	4	4	5	1	4
Digital media development	0	2	0	1	1	2	3	5	0	1
Technical support	0	1	0	1	1	3	5	5	0	1

Guía Docente de las asignaturas de Ingeniería del Software en la Ingeniería Técnica de Gestión

Versión 2.0 – 24-7-2006

María Dolores Muñoz Vicente

Departamento de Informática y Automática

Escuela Politécnica Superior de Zamora – Universidad de Salamanca

Avda. Requejo N° 33, 49022, Zamora, España

Tfno. +34 980 545000. Ext. 3635

Fax. +34 980 545001

mariado@usal.es

1. Contexto

1.1. Perfil de los créditos y su adecuación al perfil profesional y académico de la titulación

Para un ingeniero técnico en informática es imprescindible tener unos conocimientos robustos de cómo desarrollar un sistema software, teniendo en cuenta todas las fases de su ciclo de vida, desde que surge la necesidad de realizar dicho sistema hasta su implementación.

La *Ingeniería del Software*, que tradicionalmente ha aparecido dentro de la Ciencia de la Computación, en los últimos años está siendo considerada como una entidad curricular independiente, aunque con profundas raíces en la Ciencia de la Computación y las Matemáticas. De hecho, se ha definido un cuerpo de conocimiento para la Ingeniería del Software (*Software Engineering Body of Knowledge – SWEBOK*), y más recientemente se ha publicado el currículo público (2004) del *Computing Curriculum – Software Engineering* por parte de la acción conjunta entre IEEE-CS y ACM, que se mantiene como uno de los cinco perfiles profesionales formando parte también del reciente borrador (2005) del *Computing Curricula*, junto a los perfiles de *Computer Engineering*, *Computer Science*, *Information Systems* e *Information Technology*.

El Plan de Estudios de Ingeniero Técnico en Informática de Gestión destina 12 créditos para el estudio de la Ingeniería de Software. El estudio de dicha materia se lleva a cabo a través de dos asignaturas troncales, Ingeniería de Software I e Ingeniería de Software II. El cometido de estas dos asignaturas es presentar las actividades técnicas e ingenieriles que se llevan a cabo en el ciclo de vida de un producto software. Además, describe los problemas, principios, métodos y tecnologías asociadas con la Ingeniería del Software. No se puede obviar la importancia que tienen los requisitos en el éxito de un proyecto software, por ello la ingeniería de requisitos debe ser objeto de un estudio detallado, introduciendo las técnicas básicas de obtención, documentación, especificación y prototipado de los requisitos de un sistema software. Por otra parte, se debe introducir al alumno en la sistematización de la creación del software mediante el uso de métodos de ingeniería del software, tanto desde el punto de vista estructurado, típicamente tomando el método de Yourdon como referencia principal, como desde un punto de vista orientado a objetos, tomando UML (Unified Modeling Language) y el

Proceso Unificado como bases fundamentales. En las mencionadas asignaturas es demasiado ambicioso afrontar en profundidad los métodos de análisis y diseño, por ello se enfoca mayormente desde el análisis de sistemas y de requisitos, dejando los aspectos más relacionados con el diseño para otras asignaturas. Para la parte práctica de estas asignaturas se debe hacer uso de herramientas CASE (Computer Aided/Assisted Software Engineering).

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

1.2. Ubicación y relaciones en el plan de estudios

En el momento en que el alumno toma contacto con la Ingeniería del Software se encuentra en el ecuador de sus estudios, habiendo cursado un total de 67.5 créditos de materias troncales y obligatorias, lo que supone haber cursado el 42.45 % de la troncalidad y la obligatoriedad de créditos, ya que las asignaturas optativas se imparten entre el segundo cuatrimestre del segundo curso y el último año de los estudios.

Hasta ese instante el alumno ha recibido una formación en materias exclusivamente informáticas de 27 créditos, repartidos entre Programación I (7.5 créditos); Programación II (7.5 créditos); Estructuras de Datos (6 créditos) y Sistemas Informáticos (6 créditos).

La formación en materias relacionadas con los fundamentos matemáticos de la Informática, en la mitad del periodo formativo, es de 24.5 créditos (6 de Álgebra, 6 de Cálculo Diferencial, 6 de Álgebra Computacional, 7.5 de Cálculo Integral). Los 27 créditos restantes se reparten entre materias de Organización y Gestión Empresarial (7.5 créditos de Técnicas Contables Informatizadas) y 7.5 créditos de Fundamentos Físicos de la Informática.

Entre el segundo cuatrimestre del segundo curso y el tercer curso, el alumno debe superar un total de 56 créditos de materias troncales y obligatorias con contenidos puramente informáticos.

Respecto a las materias optativas hay que señalar que se ofrecen un total de 54 créditos distribuidos en 12 asignaturas, todas ellas de 4.5 créditos, debiendo cursar un máximo de 18 créditos en los dos últimos años de los estudios. Por bloques formativos el reparto es, como en el caso de las materias troncales y obligatorias, marcadamente favorable al ámbito informático (22.5 créditos) y en menor medida al estadístico (9). Los restantes 22.5 créditos se reparten de manera equitativa entre Lógica Matemática (4.5); Introducción a la Nueva Economía (4.5); Dirección Estratégica de la empresa (4.5); Derecho e Informática (4.5) y Comercio Electrónico (4.5).

La figura siguiente representa la dependencia entre las asignaturas de Ingeniería de Software en el Plan de Estudios. En el establecimiento de estas dependencias se han tenido en cuenta el factor tiempo, que claramente establece el orden lógico en el que se van a cursar así como las aportaciones a la unidad docente de las mismas, en forma de contenidos y objetivos a conseguir.

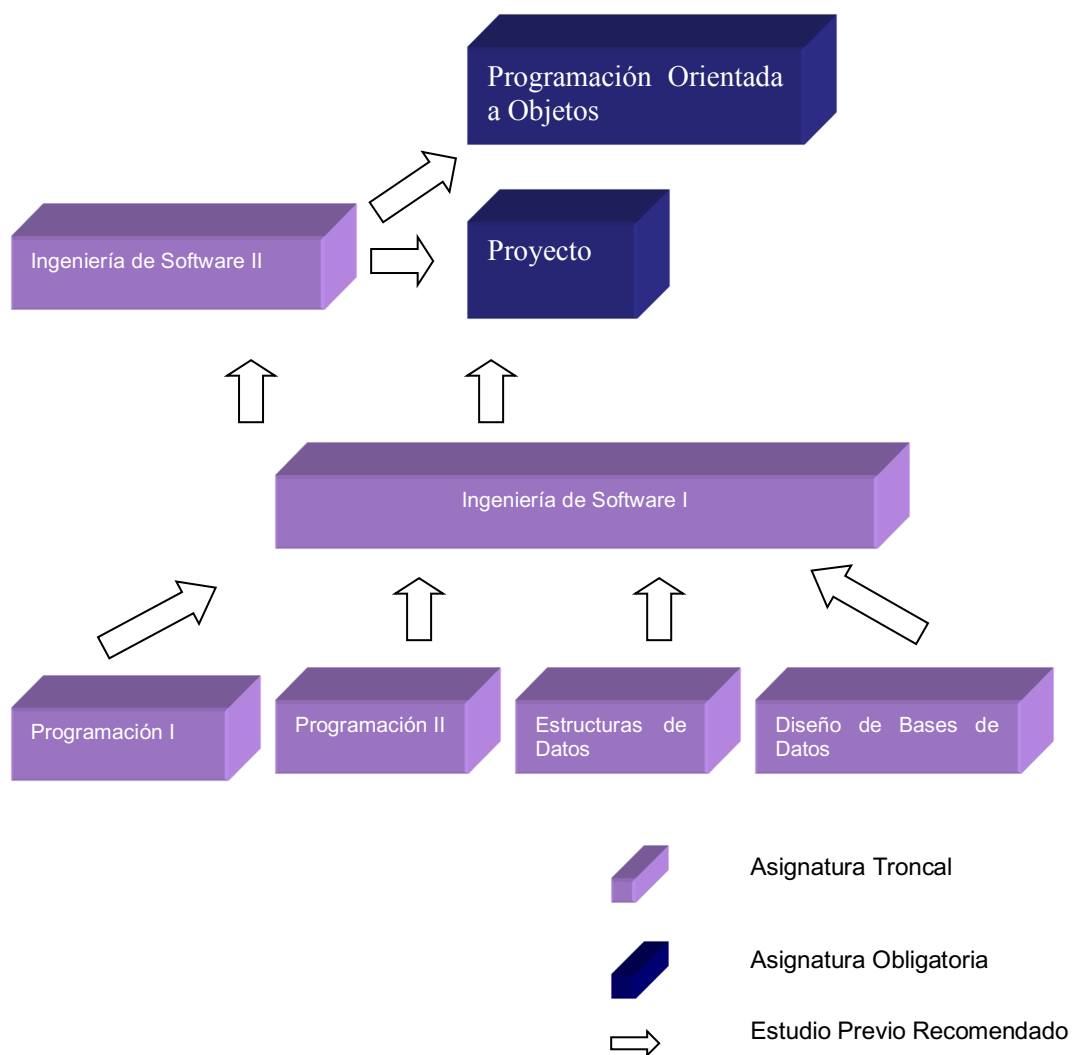


Figura 1. Relaciones de la Ingeniería del Software con otras asignaturas

2. Objetivos

2.1. Objetivos instrumentales generales

- OI1: Obtener una visión genérica y práctica del software, estudiando sus particularidades y los problemas más comunes en el proceso de construcción de Sistemas de Información.
- OI2: Identificar y establecer las diferentes etapas del desarrollo de un sistema de información, centrándose en la materia principal del curso en la fase de análisis y en el diseño del sistema de información.
- OI3: Introducir al estudiante en la problemática del desarrollo de grandes programas e incidir en la necesidad de emplear procedimientos de ingeniería para obtener software de calidad.
- OI4: Conocer las actividades técnicas e ingenieriles que se llevan a cabo en el ciclo de vida de un producto software.

- OI5: Aprender como se describen los problemas, principios, métodos y tecnologías asociadas con la Ingeniería del Software.
- OI6: Comprender la importancia de los requisitos en el ciclo de vida del software.
- OI7: Conocer las fases de obtención, documentación, especificación y prototipado de los requisitos de un sistema software.
- OI8: Aprender las especificaciones formales de requisitos.
- OI9: Aprender los métodos de análisis/diseño estructurado.
- OI10: Conocer el método de análisis/diseño orientado a objetos.
- OI11: Comprender los fundamentos del diseño de sistemas software.
- OI12: Gestión de proyectos software: definición de objetivos, gestión de recursos, estimación de esfuerzo y coste, planificación y gestión de riesgos.
- OI13: Uso de métricas software para el apoyo a la gestión de proyectos software y aseguramiento de la calidad del software.
- OI14: Conceptos, métodos, procesos y técnicas destinadas al mantenimiento y evolución de los sistemas software.
- OI15: Conocimiento sobre el uso de la Ingeniería del Software en dominios de aplicación específicos.
- OI16: Aplicar de forma práctica los conceptos teóricos sobre el desarrollo estructurado.
- OI17: Aplicar de forma práctica los conceptos teóricos de la Orientación a Objetos.
- OI18: Aplicar de forma práctica los conceptos teóricos sobre gestión de proyectos.
- OI19: Utilización de herramientas CASE para la gestión y desarrollo de sistemas software.
- OI20: Conocer y aplicar los principios de un método de análisis y diseño orientado a objetos.
- OI21: Recolección de diferentes métricas en el desarrollo de sistemas software reales.
- OI22: Construcción de sistemas software de entidad superior a una práctica de laboratorio, a ser posible partiendo de unas especificaciones reales obtenidas de clientes y/o usuarios reales.

2.2. *Objetivos interpersonales generales*

- OIP1: Destrezas para la participación responsable: capacidad de coordinación, asistencia, contribuciones al grupo...
- OIP2: Capacidad de trabajar en equipo adquiriendo y mejorando las habilidades sociales y la inteligencia emocional.
- OIP3: Comprometerse de forma ética con el trabajo, con el resto de los integrantes del grupo y consigo mismo.

2.3. *Objetivos sistémicos generales*

- OS1: El desarrollo de actitudes que sostengan la comprensión de los principios fundamentales, más valioso que la acumulación o la adquisición de habilidades y técnicas concretas.
- OS2: Aprender a pensar y razonar sobre los distintos aspectos de la asignatura, desarrollando hábitos críticos que permitan al alumno formar un juicio ponderado. El objetivo será lograr la adquisición de la capacidad de valorar un objeto, una idea o un método de conducta.
- OS3: El desarrollo de la habilidad suficiente para enfrentarse, aplicando los conocimientos adquiridos, con la resolución de casos prácticos similares a los que se le plantearán en su vida profesional.
- OS4: La capacidad no sólo de comprender la extensión y significado de lo que ya se conoce en el campo donde se encuadra la disciplina, sino de ser receptivo ante lo nuevo, de afrontarlo y de trabajar confiadamente de forma individual.
- OS5: Reforzar el hábito de desarrollar diferentes alternativas, cuestionando las características, riegos y viabilidad de cada una, para cada problema planteado.

3. Competencias

3.1. Competencias instrumentales

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

3.1.1. Habilidades cognitivas

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

Generales

- CIC1: Conocer y entender las relaciones que mantiene la asignatura con el resto de asignaturas de la titulación a partir de la introducción explícita de ejemplos y comentarios, aprovechando puntos relevantes del temario.
- CIC2: Conocer y comprender algunas de las capacidades, aptitudes y conocimientos que la asignatura aporta para el desarrollo de los diversos perfiles profesionales.
- CIC3: Ser capaz de crear documentaciones técnicas completas, correctas y legibles.

Ingeniería del Software I

Unidad Didáctica I: Conceptos básicos

- CIC4: Reconocerá el rol de la información y los sistemas de información como red de vinculación en las decisiones organizacionales.
- CIC5: Sabrá identificar las diferentes componentes de un sistema de información.
- CIC6: Será capaz de distinguir las diferentes actividades técnicas e ingenieriles que se llevan a cabo en el ciclo de vida de un producto software.

- CIC7: Conocerá la forma de describir los problemas, principios, métodos y tecnologías asociadas con la Ingeniería del Software.
- CIC8: Aprenderá las características más significativas de los sistemas software.
- CIC9: Podrá establecer los conceptos de ciclo de vida del desarrollo del software, bajo los distintos planteamientos que se han ido desarrollando.
- CIC10: Comprenderá la necesidad del empleo de metodología en el desarrollo del software, como única forma de desarrollar software profesional y de calidad.

Unidad Didáctica II: Paradigma estructurado de desarrollo

- CIC11 Será capaz de comprender la importancia de los requisitos en el ciclo de vida del software.
- CIC12 Conocerá como se debe documentar y especificar los diferentes requisitos de un sistema software.
- CIC13 Adquirirá una visión en torno a la problemática de la obtención, gestión, análisis, documentación y especificación de los requisitos de un sistema software.
- CIC14 Será capaz de comprender y realizar un análisis utilizando técnicas estructuradas, en las que se le introduce en sus aspectos teóricos y prácticos más relevantes.
- CIC15 Entenderá las ventajas de utilizar técnicas de especificación gráficas, que facilitan la obtención de modelos y favorezcan la comunicación con clientes y/o usuarios.
- CIC16 Conocerá la diferencia entre la postura mantenida inicialmente por los partidarios de la descomposición funcional estricta, defendida sobre todo por el método de Gane-Sarson, y el método de los estímulos que es sostenido por Yourdon.
- CIC17 Estará familiarizado con la aplicación de un método sistemático en la aplicación de las técnicas estructuradas estudiadas en el tema anterior.
- CIC18 Aprenderá la importancia del diseño en la obtención de un software de calidad.
- CIC19 Entenderá la necesidad de que los diseños deben respetar una serie de principios, como única forma de abordar un trabajo de forma profesional, alejado de la artesanía que tradicionalmente se venía utilizando en la construcción de los sistemas software.
- CIC20 Comprenderá que un criterio de calidad consiste en buscar la independencia modular, desarrollando unos módulos altamente cohesionados, con bajo acoplamiento, y que hagan acopio del principio de ocultación de la información, comunicándose a través de unas interfaces precisas y de pequeño tamaño.

Unidad Didáctica III: Introducción al paradigma objetual

- CIC21 Aprenderá los conceptos fundamentales de la tecnología de objetos, conociendo las diferencias que presenta esta manera de desarrollar software respecto a los planteamientos estructurados.
- CIC22 Entenderá el hecho de que los objetos representan elementos del mundo real dentro del dominio del problema que se debe resolver, siendo por tanto, familiares al usuario. Los objetos llevan incorporado el comportamiento habitual que el experto en el dominio espera de ellos.
- CIC23 Comprenderá dentro de los conceptos fundamentales, el especial relieve que tienen para la obtención de un software de calidad los conceptos de clasificación (jerarquía) y encapsulación.
- CIC24 Será capaz de comprender el aumento de productividad que se alcanza gracias al polimorfismo y la herencia, pero también el significado avanzado de estos conceptos en las fases de análisis y diseño.
- CIC25 Comprenderá la trascendencia de la orientación a objetos en el análisis de aplicaciones.
- CIC26 Será capaz de comentar y discutir los conceptos básicos del modelo objeto basándose en las aportaciones de varios autores, y el modo en el que pueden soslayarse las posibles limitaciones tanto conceptuales, como de notación.
- CIC27 Comprenderá el concepto de reutilización del software, su importancia para el aumento de la productividad en el desarrollo del software y su relación con la orientación a objetos. En este sentido debe hacerse hincapié en que la reutilización del software está presente en todo el ciclo de vida, no sólo en la implementación, debiendo empezar a pensar en ella en las primeras fases del ciclo de vida.

Ingeniería del Software II

Unidad Didáctica I: Paradigma Objetual

- CIC28 Entenderá qué es y qué no es UML, estudiando de forma somera sus orígenes e historia.
- CIC29 Habrá tenido una primera toma de contacto con el vocabulario, las reglas, formas de empleo de UML y, en general, con el manejo de la simbología y el adecuado alcance de los conceptos.
- CIC30 Conocerá el modo de aplicar los modelos de UML para resolver los problemas implícitos en el desarrollo del software.
- CIC31 Conocerá el modelado estructural a un nivel básico.
- CIC32 Aprenderá el concepto de caso de uso y entenderá su importancia en las corrientes metodológicas actuales, bien como herramienta de extracción y documentación de los requisitos del sistema software, bien como herramienta para modelar el comportamiento esencial de un sistema o subsistema en conjunción con otros diagramas (colaboración, secuencia, transición de estados...).

Unidad Didáctica II: Ingeniería del Software Asistido por Computador

- CIC33 Entenderá el uso de las herramientas CASE como apoyo y automatización al proceso de desarrollo de software.
- CIC34 Conocerá la diferente tipología de herramientas CASE que existen en el mercado y la importancia de su integración para conseguir entornos avanzados de Ingeniería del Software.
- CIC35 Entenderá la importancia del intercambio de datos entre diferentes herramientas CASE para favorecer la integración, reutilización e intercambio de información entre proyectos.
- CIC36 Conocerá la arquitectura que se encuentra detrás de una herramienta CASE.

Unidad Didáctica III: Gestión de proyectos de software

- CIC37 Conocerá la relación entre las actividades llevadas a cabo en la planificación de un proyecto software.
- CIC38 Comprenderá por qué planear proyectos es esencial en todos los proyectos de software.
- CIC39 Sabrá especificar las tareas necesarias llevadas a cabo en la planificación de un proyecto.

3.1.2. Capacidades metodológicas

- CIM1: Ser capaz de tomar decisiones de manera razonada.
- CIM2: Tener capacidad de análisis y síntesis.
- CIM3: Ser capaz de manejar bibliografía relacionada con la Ingeniería del Software.
- CIM4: Ser capaz de documentar la solución a un problema dado en las diferentes fases del ciclo de vida con los diferentes niveles de abstracción y aproximación que esto requiere.
- CIM5: Ser capaz de aplicar correctamente y manejar con comodidad los diferentes elementos de los que dispone un lenguaje de modelado concreto, ya sea orientado a objetos u orientado a procesos.
- CIM6: Ser capaz de modelar un sistema de información desde las diferentes perspectivas que exige un determinado proceso.

3.1.3. Destrezas tecnológicas

- CIT1: Habilidades básicas de navegación por la Web y uso del resto de servicios de red para la obtención y manejo de la información relacionada con la asignatura.
- CIT2: Manejar las operaciones básicas y de administración de diferentes sistemas operativos para la instalación de las diferentes herramientas CASE a utilizar en la asignatura.
- CIT3: Manejar con fluidez diferentes herramientas CASE.
- CIT4: Manejo avanzado de un sistema de procesamiento de textos para la realización de los informes de las prácticas y la documentación de la práctica obligatoria.

3.1.4. Destrezas lingüísticas

- CIL1: Adquirir y utilizar con fluidez un buen lenguaje científico, tanto oral como escrito, siendo riguroso en las explicaciones de cualquier proceso dentro de la asignatura.
- CIL2: Conocer y utilizar la terminología usual de la Ingeniería del Software, tanto en español como en inglés.

3.2. Competencias interpersonales

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

3.2.1. Competencias para tareas colaborativas

- CIPTC1: Ser capaz de realizar de trabajar en equipo para resolver los problemas planteados en los enunciados de las partes prácticas de las asignaturas.
- CIPTC2: Ser capaz de presentar en público una solución a un problema planteado y mantener un debate con el resto de la clase sobre la solución planteada, para así buscar colaborativamente la mejor solución al problema.
- CIPTC3: Ser capaz de realizar y defender la práctica obligatoria de la asignatura en equipo.

3.2.2. Compromiso con el trabajo

- CIPTR1: Se ha de definir un plan de trabajo en el que el volumen de trabajo de todos los miembros del equipo sea similar.
- CIPTR2: Una vez finalizado el trabajo, todos los miembros del grupo deben conocer en profundidad todo el desarrollo realizado.
- CIPTR3: Se debe cumplir el plazo de entrega de los trabajos.
- CIPTR4: Se debe adquirir un compromiso ético entre todos los componentes del grupo.

3.3. Competencias sistémicas

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

- CS1: Capacidad de aplicar los conocimientos, métodos y herramientas vistos en las asignaturas de Ingeniería del Software a situaciones y problemas concretos del área de la Ingeniería Informática y de otras disciplinas relacionadas.
- CS2: Capacidad de aprender y aplicar, de forma autónoma e interdisciplinar, nuevos conceptos y métodos relacionados con cada asignatura.
- CS3: Motivación por la calidad y por la creatividad.
- CS4: Capacidad de adoptar el proceso marcado por el método científico y de ingeniería en el planteamiento y realización de trabajos diversos, tanto a nivel académico como profesional.
- CS5: Capacidad de asimilación y adaptación a la evolución del estado del arte en el ámbito de la Ingeniería del Software y de la Ingeniería Informática como profesión.

4. Prerrequisitos

4.1. Competencias y contenidos mínimos

- Conocer los elementos básicos de un lenguaje de programación de 3ª generación.
- Conocer las principales formas de representar los algoritmos.
- Conocer el concepto de tipo abstracto de datos.
- Conocer los principios fundamentales del diseño de estructuras de datos.
- Conocer los principios fundamentales del modelado conceptual de datos mediante el diagrama entidad/relación.
- Conocer los mecanismos de transformación de un modelo conceptual de datos a un modelo lógico de datos.

4.2. Plan de trabajo para la consecución de los prerrequisitos

Los prerrequisitos para el estudio y entendimiento de las asignaturas de Ingeniería del Software se cubren fundamentalmente en las asignaturas de programación I del primer curso, y en las asignaturas Programación II, Estructura de Datos y Bases de Datos del segundo curso.

A través de la asignatura Diseño de Bases de Datos al alumno se le forma en conceptos relacionados con los modelos de datos conceptuales y lógicos (diagramas entidad/relación y modelos relacionales típicamente), lo que supone una importante base, a la vez que una descarga, para las asignaturas de Ingeniería del Software donde estos modelos serán utilizados de forma práctica sin necesidad de tener que incluirlos en la parte teórica de dichas asignaturas.

Por otra parte el alumno en el primer año de la carrera cursa la asignatura Programación I, en la que aprende someramente ciertas características destacables de la Ingeniería del Software, ya que se destina un tema de los once que conforman el temario de esta asignatura, para su estudio.

También deberá estar familiarizado con la teoría y la práctica del diseño y codificación de lenguajes procedurales conceptos adquiridos en las asignaturas relacionadas con los lenguajes de programación: Programación I y Programación II y Estructuras de Datos.

5. Temario

5.1. Unidades didácticas

Ingeniería del Software I

Unidad Didáctica I: Conceptos básicos

Tema 1. Introducción a la Ingeniería del Software

Unidad Didáctica II: Paradigma estructurado de desarrollo

Tema 2. Análisis y especificación de requisitos

Tema 3. Análisis estructurado

Tema 4. Diseño del software

Unidad Didáctica III: Introducción al paradigma objetual

Tema 5. Introducción a la orientación a objetos

Ingeniería del Software II

Unidad Didáctica I: Paradigma objetual

Tema 1. UML

Unidad Didáctica II: Ingeniería del software asistido por computador

Tema 2. Herramientas CASE

Unidad Didáctica III: Gestión de proyectos de software

Tema 3. Gestión de proyectos de software

Tema 4. Control de calidad del software

Tema 5. Evolución y mantenimiento del software

5.2. Desarrollo de las unidades didácticas

Ingeniería del software I

Unidad Didáctica I: Conceptos básicos

Tema 1. Introducción a la Ingeniería del Software

1.1. Conceptos generales

1.2. Sistemas de información

1.3. Ingeniería del software

1.4. Ciclo de vida del software

1.5. Paradigmas de la ingeniería del software

1.6. Metodologías

1.7. CASE

Unidad Didáctica II: Paradigma estructurado de desarrollo

Tema 2. Análisis y especificación de requisitos

2.1. Introducción al análisis

2.2. Especificación de requisitos del software

2.3. Técnicas de especificación

2.4. Modelado funcional

2.5. Modelado de información

2.6. Modelado del comportamiento

2.7. Balanceo de modelos

Tema 3. Análisis estructurado

3.1. Introducción al análisis

3.2. Enfoque de modelado clásico

3.3. Enfoque de modelado de Yourdon

Tema 4. Diseño del software

- 4.1. Introducción
- 4.2. Proceso de diseño
- 4.3. Actividades del diseño
- 4.4. Fundamentos de diseño
 - 4.5. Diseño modular
 - 4.6. Diseño estructurado

Unidad Didáctica III: Introducción al paradigma objetual

Tema 5. Introducción a la orientación a objetos

- 5.1. Introducción y evolución a la orientación a objetos
- 5.2. Modelo objeto
- 5.3. Análisis y diseño orientado a objetos

Ingeniería del Software II

Unidad Didáctica I: Paradigma objetual

Tema 1. UML

- 1.1. Conceptos generales
- 1.2. Una visión general de UML
- 1.3. Diagramas de estructuras
- 1.4. Diagramas de iteración
- 1.5. Casos de uso

Unidad Didáctica II: Ingeniería del software asistido por computador

Tema 2. Herramientas CASE

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

Unidad Didáctica III: Gestión de proyectos del software

Tema 3. Gestión de proyectos

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

Tema 4. Control de calidad del software

- 4.1. Introducción
- 4.2. Conceptos de calidad

- 4.3. Tendencias de la calidad
- 4.4. Garantía de calidad del software
 - 4.5. Revisiones del software
 - 4.6. Revisiones técnicas formales
- 4.7. Estándar de calidad ISO 9001
- 4.8. El plan de calidad del software

Tema 5. Evolución y mantenimiento del software

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

6. Metodología y estrategias de aprendizaje

6.1. Metodología docente

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

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

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

Siendo congruentes con lo anterior, y con carácter general, el Ingeniero en Informática debe estar capacitado para aprender a conocer, hacer, convivir y ser, en su ámbito

personal, profesional y social, de acuerdo con lo recogido en el informe de la UNESCO sobre las perspectivas de la educación en el siglo XXI.

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

- *La exposición teórica.* Dentro de los tipos de lecciones magistrales, se utiliza la lección magistral participativa. Este modelo de lección magistral facilita el aprendizaje activo y cooperativo de los estudiantes. Utilizando ejemplificaciones de la realidad, podremos analizar, evaluar e incluso modificarla. En este caso el papel del profesor cambia, no solamente transmitirá la información que los alumnos necesitan, sino que a lo largo de la clase el profesor actuará como animador convirtiendo en reflexión colectiva las aportaciones individuales. De esta forma los alumnos ya no serán simples y pasivos receptores del mensaje emitido por el profesor, sino que serán capaces de discutir, analizar y reflexionar sobre los temas tratados en clase. Se considerará por tanto, una buena lección magistral participativa si promueve el conocimiento por comprensión, crea la necesidad de seguir aprendiendo, proporciona un ambiente de trabajo personal y colaborativo entre los alumnos y si el alumno consigue asumir la responsabilidad y el protagonismo del aprendizaje.

Es importante usar las nuevas tecnologías para la exposición de las clases pero no se puede dejar de lado la importancia de la utilización de la pizarra. El uso de la pizarra es fundamental ya que permite realizar dinámicamente la exposición de conceptos, en el sentido de que lo representado en la pizarra puede evolucionar, siendo muy útil para la ejemplificación de procesos dinámicos y para la realización de modificaciones sobre lo representado

- *Clases de problemas.* El verdadero objetivo de estas clases consiste en potenciar en el alumno la capacidad de reflexionar sobre los principios teóricos adquiridos con anterioridad y aplicar éstos sobre un problema concreto para obtener su resolución. En este tipo de clase el papel predominante lo debe tener el alumno, siendo el profesor el que guía la clase y propone la resolución de aquellos problemas más interesantes.

En las primeras clases el profesor realizará ejercicios incidiendo en la necesidad de relacionar los conceptos teóricos con los prácticos y a lo largo de la clase irá haciendo reflexiones conjuntas con el alumnado de manera que éste consiga comprender los razonamientos seguidos por el profesor. Una vez que se han hecho los primeros ejercicios el alumno se encuentra en disposición para poder realizar de manera autónoma los ejercicios propuestos por el profesor, que se irán corrigiendo a lo largo de las clases de problemas. A la hora de la corrección será el alumno el que expondrá su solución al resto de la clase de manera que le servirá al profesor para darse cuenta de las carencias en el aprendizaje de los alumnos.

- *Prácticas de laboratorio:* la estructura de las clases prácticas es la siguiente:

- *Introducción:* El profesor repasa los conceptos fundamentales que se van a tratar en la sesión, haciendo hincapié en los aspectos más novedosos o que por la experiencia se sabe que causan más problemas. Esta introducción permite centrar el esfuerzo del alumno en lo realmente importante y anticiparse a dudas y errores comunes.
- *Práctica guiada:* El profesor haciendo uso de los medios disponibles tales como la pantalla de cristal líquido y la pizarra realizara un ejemplo simple y relacionado con la parte de la práctica que en este momento se esté realizando. De este modo el alumno podrá ir practicando lo aprendido hasta el momento.
- *Práctica libre:* Finalizado el tiempo de explicación por parte del profesor el alumno ayudado de su guión deberá resolver la práctica propuesta. En este momento el alumno deberá demostrar que ha entendido las técnicas estudiadas y que sabe aplicarlas en la solución de problemas nuevos.
- *Supervisión y evaluación:* A lo largo de toda la sesión el profesor supervisará el trabajo de los alumnos y atenderá las dudas que se vayan planteando. En ciertos momentos si las dudas se repiten, es aconsejable que el profesor pare la clase con el fin de resolver las dudas a toda la clase y no de manera individual. De este modo, el estudiante recibe respuesta a sus dudas en el momento en que más receptivo está: cuando éstas se plantean.
- *Tutorías.* El alumnado tiene a su disposición seis horas de tutorías a la semana en las que puede consultar cualquier duda relacionada con los contenidos, organización y planificación de la asignatura. Las tutorías pueden ser individualizadas, pero se admite tutorías grupales para resolver problemas relacionados con las actividades a realizar en grupo.
- *Página web.* Se convierte en el vehículo de comunicación y registro de información de la asignatura.

6.2. Estrategias de aprendizaje

Se detalla las actuaciones concretas a realizar para la aplicación y aprovechamiento de la metodología propuesta. Se estructura en las siguientes fases:

6.2.1. Recopilación de la documentación de la asignatura

- Los alumnos dispondrán en la página web de la asignatura de toda la información y normativa relacionada con la asignatura: temario, criterios de evaluación, bibliografía, apuntes, enunciados de prácticas, trabajos, enlaces de interés, lecturas complementarias, avisos...
- El profesor mantiene actualizada la información de esta página para que se convierta en un vehículo de comunicación con los alumnos.
- Los alumnos deben conocer y manejar con fluidez toda la información y la documentación que se integra en la página de la asignatura.
- Asiduamente, el alumno debe consultar la página para estar actualizado.

6.2.2. Planificación de las clases teóricas

Las exposiciones teóricas se organizan en sesiones de una hora, cuyo desarrollo se ajusta al siguiente esquema:

1. *Introducción:* En esta parte se muestra el esquema de lo que va a ser tratado a lo largo de la clase, sirviendo a su vez de enlace con lo estudiado en clases anteriores. Se fijan los objetivos a lograr justificando el orden propuesto y se da la bibliografía específica para complementar lo expuesto. La introducción por lo tanto, deberá servir para fijar la atención del estudiante sobre los conceptos que se van a explicar, a la vez de permitir que recuerde y relacione la exposición de la clase anterior con la actual.
 2. *Desarrollo:* Una vez introducido el tema, se desarrollara la explicación siguiendo la secuencialidad mostrada en los items fijados como objetivos. La exposición debe ser clara y debe estar bien estructurada de forma que le permita al alumno identificar y comprender los conceptos más importantes. Se debe utilizar un vocabulario preciso pero conocido y comprensible para el alumno. En todo momento se deberá crear un clima distendido que favorezca el aprendizaje y la participación por parte del alumno. A lo largo de la explicación deberán producirse pausas en las que se planteen una serie de cuestiones orientadas a conocer el grado de seguimiento del alumno y le permita tanto al profesor como al alumno valorar los conocimientos adquiridos. El profesor aludiendo a problemas reales o hipotéticos, experiencias personales...deberá despertar en los alumnos el interés por el tema tratado. Está demostrado que el uso de analogías y comparaciones con conceptos y situaciones cotidianas agiliza la comprensión de los conceptos abstractos y ameniza la clase facilitando el seguimiento.
 3. *Resumen y conclusiones:* Esta parte de la clase servirá para concretar las ideas fundamentales que se hayan expuesto a lo largo de la clase así como para sugerir nuevos problemas y cuestiones relacionadas. Si todavía no se ha llegado a conclusiones el profesor, deberá remarcar hacia dónde se pretende llegar y mostrar el camino recorrido.
- Una vez terminada la clase magistral, se debe estudiar de forma autónoma su contenido y en caso de no entender algo intentar aclararlo utilizando la bibliografía recomendada o cualquiera de los materiales adicionales recomendados en la asignatura.

6.2.3. Planificación de las clases prácticas

- El profesor publicará en la página de la asignatura los enunciados, con indicación de plazos de entrega, con suficiente antelación.

Para la realización de las prácticas el grupo deberá seguir el procedimiento siguiente:

- Leer el problema en equipo y encontrar las definiciones de los términos desconocidos.
- Identificar los conocimientos necesarios para resolver el problema.
- Clasificar los conocimientos.
- Establecer las prioridades de búsqueda y estudio.
- Repartir el trabajo.
- Buscar documentación y estudiar individualmente conforme a las prioridades establecidas.

Durante la revisión de la práctica el profesor moderará el debate que se origine sobre el problema y la solución presentada, dando argumentos a cada una de las decisiones o alternativas que se planteen.

Los alumnos deben cumplir el calendario de entrega de prácticas y defenderlas en grupo.

6.2.4. Evaluación

- Se plantea una forma de evaluación continua.
- Para aquellos alumnos que no superen la evaluación continua o no se quieran o no puedan acogerse a ella existirá una prueba final.

7. Planificación del tiempo y del esfuerzo

La carga docente asignada a las asignaturas de Ingeniería del Software es la siguiente:

Ingeniería del Software I: 6 créditos (4.5 teóricos y 1.5 prácticos)

Ingeniería del Software II: 6 créditos (3 teóricos y 3 prácticos)

Para hacer su estudio en ECTS se ha tomado la equivalencia de crédito LRU a crédito ECTS, con implicación de cambio de metodología docente.

Al hacer el estudio de carga de trabajo máxima, se tiene que:

- 6 ECTS -> 150-180 horas de trabajo
- 4,5cr LRU -> 45 horas presenciales + (45*1,5) horas de asimilación = 112,5 horas
- 1,5cr LRU -> 15 horas presenciales + (15*2) horas de asimilación = 45 horas
- $112,5 + 45 = 157,5$ horas de trabajo
- $150 < 157,5 < 180$

Dos son los escenarios más probables: el resultante de tomar el crédito ECTS como 30 horas de trabajo y el resultante de tomarlo como 25 horas de trabajo.

Por temas de organización y calendario, se ha optado por partir del escenario que asigna 25 horas de trabajo al ECTS.

Para poder elaborar las guías, cuyo destino final sería la realización de experiencias piloto, dentro de los planes de estudio vigentes, se debe tener en cuenta la definición del crédito del R.D. 1497/1987 y sus posteriores modificaciones. Así:

- La unidad de valoración de las enseñanzas se corresponde a diez horas de enseñanza teórica, práctica o de sus equivalencias entre las que podrán incluirse actividades académicas dirigidas, que habrán de preverse en el correspondiente plan docente junto con los mecanismos y medios objetivos de comprobación de los resultados académicos de las mismas. Todo ello sin perjuicio del cumplimiento del régimen de dedicación del profesorado, de conformidad con el Real Decreto 898/1985, de 30 de abril sobre régimen del profesorado universitario.
- En ningún caso, salvo que se trate de enseñanzas en Universidades a distancia, el porcentaje del crédito correspondiente a las actividades académicas dirigidas será superior al **30%**.

De esta forma, si 6 créditos LRU equivalen a 60 horas, se debe asegurar el 70% como mínimo de presencialidad, esto es, 42 horas entre horas de teoría y de práctica.

En las Tablas 1 y 2 se presentan la organización del esfuerzo del alumno para cubrir los 6 ECTS asociados a las dos asignaturas destinadas al estudio de la Ingeniería del Software en la titulación. Por su parte en las Tablas 3 y 4 se distribuyen las horas presenciales entre los diferentes temas y el examen.

			A	B	C	D	E
	Técnica	Actividad	Horas equivalentes de clase	Factor de trabajo del alumno [†]	Horas de trabajo personal del alumno	Horas totales (A+C)	ECTS (D ÷ 25)
Teoría	Seguimiento de la parte teórica de las unidades didácticas	Asimila contenidos. Se plantea dudas que planteará a los profesores en las tutorías	36	1,5	54	90	3,6
Práctica	Seguimiento de la parte práctica de las unidades didácticas	Experimenta, practica, modela, simula	6	2	12	18	0,72
Ejercicios de apoyo	Ejercicios de repaso y afianzamiento de lo aprendido.	Realiza, resuelve problemas, tests...	-	-	5	5	0,2
Práctica obligatoria	Prácticas de obligada realización	Realiza una ERS siguiendo distintos enfoques de modelado	-	-	30	30	1,2
Examen	Controles de evaluación continua y/o examen final de la asignatura	Exámenes tipo test, supuestos prácticos...	3			3	0,12
Otras actividades	Tutorías personalizadas y grupales	Recibe orientación personalizada	-	-	2	2	0,08
	Búsquedas en la red, participación en foros especializados...	Busca elementos para completar los contenidos	-	-	2	2	0,08
TOTAL			45		104	149	6

Tabla 1. Ingeniería del Software I-Escenario ECTS = 25 horas de trabajo

[†] Número de horas dedicadas por el alumno al trabajo personal (organización de apuntes, estudio, documentación...) por cada hora de clase.

			A	B	C	D	E
	Técnica	Actividad	Horas equivalentes de clase	Factor de trabajo del alumno†	Horas de trabajo personal del alumno	Horas totales (A+C)	ECTS (D ÷ 25)
Teoría	Seguimiento de la parte teórica de las unidades didácticas	Asimila contenidos. Se plantea dudas que planteará a los profesores en las tutorías	28	1,5	45	75	3
Práctica	Seguimiento de la parte práctica de las unidades didácticas	Experimenta, practica, modela, simula	15	2	30	45	1,8
Ejercicios de apoyo	Ejercicios de repaso y afianzamiento de lo aprendido.	Realiza, resuelve problemas, tests...	-	-	2	2	0,08
Práctica obligatoria	Práctica en grupo de obligada realización	Realiza una ERS completada con elementos de diseño	-	-	24	24	0,96
Examen	Controles de evaluación continua y/o examen final de la asignatura	Exámenes tipo test, supuestos prácticos...	2			2	0,08
Otras actividades	Tutorías personalizadas y grupales	Recibe orientación personalizada	-	-	1	1	0,04
	Búsquedas en la red, participación en foros especializados...	Busca elementos para completar los contenidos	-	-	1	1	0,04
TOTAL			45		113	160	6

Tabla 2. Ingeniería del Software II-Escenario ECTS = 25 horas de trabajo

NÚMERO DE HORAS PRESENCIALES		
Actividad	Horas Teoría	Horas Prácticas
Unidad Didáctica I: Conceptos básicos		
Tema 1	8	
Unidad Didáctica II: Paradigma estructurado de desarrollo		
Tema 2	10	6
Tema 3	2	

† Número de horas dedicadas por el alumno al trabajo personal (organización de apuntes, estudio, documentación...) por cada hora de clase.

Tema 4	8	
Unidad Didáctica III: Introducción al paradigma objetual		
Tema 5	8	
Examen	3	
TOTAL: 45	39	6

Tabla 3. Reparto de horas presenciales Ingeniería del Software I

NÚMERO DE HORAS PRESENCIALES		
Actividad	Horas Teoría	Horas Prácticas
Unidad Didáctica I: Paradigma objetual		
Tema 1	11	5
Unidad Didáctica II: Ingeniería del software asistido por computador		
Tema 2	3	
Unidad Didáctica III: Gestión de proyectos de software		
Tema 3	7	10
Tema 4	4	
Tema 5	3	
Examen	2	
TOTAL: 45	30	15

Tabla 4. Reparto de horas presenciales Ingeniería del Software II

8. Bibliografía

Ingeniería del software I

8.1. Bibliografía básica

Booch, G., Rumbaugh, J., Jacobson, I. (1999) *El Lenguaje Unificado de Modelado*. Addison Wesley.

Meyer B.(1999) *Construcción de Software Orientado a Objetos*. 2ª Edición. Prentice Hall,

Piattini, M. G., Calvo-Manzano, J. A., Cervera, J., Fernández, L. (2004) *Análisis y Diseño de Aplicaciones Informáticas de Gestión. Una perspectiva de Ingeniería del Software*. Ra-ma.

Pressman, R. S. (2005) *Ingeniería del Software: Un Enfoque Práctico*. 6ª Edición. McGraw-Hill.

Rumbaugh, J., Jacobson, I., Booch, G. (2000) *El Lenguaje Unificado de Modelado. Manual de Referencia*. Addison-Wesley.

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

Yourdon, E. (1993) *Análisis Estructurado Moderno*. Prentice-Hall Hispanoamericana.

8.2. Bibliografía complementaria

- Andreu, R., Ricart, J., Valor, J. (1996) *Estrategia y Sistemas de I Información*. 2ª Edición, McGraw-Hill (serie de management).
- Amescua Seco, A. de, García Sánchez, L., Martínez Fernández, P. y Díaz Pérez, P.(1995) *Ingeniería del Software de Gestión. Análisis y Diseño de Aplicaciones*, Paraninfo.
- Booch, G. (1996) *Análisis y Diseño Orientado a Objetos con Aplicaciones*. 2ª Edición. Addison-Wesley/Díaz de Santos.
- Fairley, R.(1998) *Ingeniería del Software*. McGraw Hill.
- Martin, J., Odell, J. J.(1997) *Métodos Orientados a Objetos: Conceptos Fundamentales*,. Prentice Hall.
- Martin, J., Odell, J. J.(1997) *Métodos Orientados a Objetos: Consideraciones Prácticas*.Prentice Hall Hispanoamericana.
- Pfleeger, S. L. (2002) *Ingeniería del Software. Teoría y Práctica*. Prentice Hall.

Ingeniería del software II

8.3. Bibliografía básica

- Booch, G. (1996) *Análisis y Diseño Orientado a Objetos con Aplicaciones*. 2ª Edición. Addison-Wesley/Díaz de Santos.
- Muller, P. A.(1997) *Modelado de objetos con UML. Eyrolles-Ediciones Gestión 2000*.
- Rumbaugh, J., Jacobson, I., Booch, G (2000). *El Lenguaje Unificado de Modelado. Manual de Referencia*. Addison-Wesley.
- Piattini M.G, Calvo-Manzano J.A., Cervera J., Fernández L. (2004).*Análisis y diseño de aplicaciones informáticas de gestión*. Rama.
- Pfleeger, S. L. (2002) *Ingeniería del Software. Teoría y Práctica*. Prentice Hall.
- Pressman, R. S.(2005)*Ingeniería del Software: Un Enfoque Práctico*. 5ª Edición. McGraw-Hill.
- Sommerville, I. (2005) *Ingeniería del Software*. 7ª Edición, Pearson Addison-Wesley.

8.4. Bibliografía complementaria

- Fowler, M., Scott, K (1999).. *UML Gota a Gota*. Addison Wesley Longman (Pearson).
- Jacobson, I. Ericsson, M., Jacobson(1994) A. *The Object Advantage-Business Process Reengineering with Object Technology*. Addison Wesley.
- Larman, C.(1999) *UML y Patrones. Introducción al Análisis y Diseño Orientado a Objetos*. Pearson.
- Puig (1994) *Proyectos Informáticos. Planificación, Desarrollo y Control*. Paraninfo.
- Quang (1994) *Dirección de Proyectos Informáticos*. Enrolles.
- Stevens, P., Pooley, R.(2002) *Utilización de UML en Ingeniería del Software con Objetos y Componentes*. Addison-Wesley.

8.3. Otros recursos

- Durán, A., Bernárdez, B. (2002) Metodología para la Elicitación de Requisitos de Sistemas Software (versión 2.3). Informe Técnico LSI-2000-10, Universidad de Sevilla. <http://www.lsi.us.es/~amador>. [Última vez visitado, 22-9-2005].
- Durán, A., Bernárdez, B. (2001) Metodología para el Análisis de Requisitos de Sistemas Software (versión 2.2). Universidad de Sevilla. <http://www.lsi.us.es/~amador>. [Última vez visitado, 22-9-2005].
- OMG (2003) OMG Unified Modeling Language Specification. Version 1.5. Object Management Group Inc. Document formal/03-03-01. March 2003. <http://www.omg.org/docs/formal/03-03-01.pdf>. [Última vez visitado, 22-9-2005]
- OMG (2005) Unified Modeling Language: Superstructure. Version 2.0. Object Management Group Inc. Document formal/05-07-04. August 2005. <http://www.omg.org/cgi-bin/doc?formal/05-07-04>. [Última vez visitado, 22-9-2005].
- Pohl, K. (1997) Requirements Engineering: An Overview. En M. Dekker (Ed.), *Encyclopedia of Computer Science and Technology*, 36. Disponible en <ftp://sunsite.informatik.rwth-aachen.de/pub/CREWS/CREWS-96-02.pdf>. [Última vez visitado, 22-9-2005].

8.4. Enlaces de interés

- **Association for Computing Machinery (ACM)**
 - <http://www.acm.org>.
 - Fundada en 1947 fue la primera sociedad científica y de educación del mundo. El portal de información que presenta es impresionante, tanto en cuanto a enlaces de interés, grupos de trabajo, documentos electrónicos, conferencias como por su biblioteca digital conteniendo revistas y actas de congresos (<http://portal.acm.org>).
- **Cetus Links - Object-Oriented**
 - <http://www.cetus-links.org>.
 - La colección de enlaces más completa sobre Orientación a Objetos. A fecha de 17 de marzo de 2004 se tenían contabilizados 18193 enlaces.
- **Consejo Superior de Informática**
 - <http://www.csi.map.es/>.
 - Web del Consejo Superior de Informática donde, entre otras cosas, se puede obtener la documentación y herramientas sobre Métrica 3 (<http://www.csi.map.es/csi/metrica3/index.html>).
- **DACS (Data & Analysis Center for Software)**
 - <http://www.dacs.dtic.mil>.
 - Sitio perteneciente al DoD de EEUU con numerosa información y enlaces a lugares relacionados con la Ingeniería del Software.
- **European Software Institute (ESI)**
 - <http://www.esi.es>.

- El Instituto del Software Europeo tiene su sede en Bilbao (España). Dispone tanto de documentación privada para los miembros del instituto como documentación pública con los análisis de proyectos, necesidades de empresas y software europeo.
- **Guide to the Software Engineering Body of Knowledge (SWEBOK)**
 - <http://www.swebok.org>.
 - Proyecto para establecer un cuerpo de conocimiento común para la Ingeniería del Software.
- **Institute of Electrical and Electronics Engineers (IEEE)**
 - <http://www.ieee.org>.
 - Otra prestigiosa organización compuesta por diversas sociedades, donde la que más relación tiene con los temas abordados en el presente curso es la IEEE Computer Society (<http://computer.org>).
 - A semejanza de ACM, ofrece información sobre conferencias, estándares, educación y mantiene otra biblioteca digital con revistas y actas de congresos.
- **Object Management Group (OMG)**
 - <http://www.omg.org>.
 - Es un consorcio internacional de industrias con el fin de promover el uso de la Orientación a Objetos en la Ingeniería del Software. A diferencia de organizaciones como ISO o IEEE, OMG desarrolla estándares de “*facto*” como consenso entre las empresas que la forman.
 - Dicho servidor ofrece publicaciones electrónicas y enlaces a estándares y herramientas del sector relacionado con la tecnología de objetos.
 - En <http://www.omg.org/uml> se encuentran las especificaciones de la versión actual (así como de las anteriores) de UML.
- **IBM Rational Software**
 - <http://www-306.ibm.com/software/rational/>.
 - En esta dirección se tiene valiosa información sobre UML y RUP (documentos, informes, artículos, presentaciones, bibliografía recomendada...).
 - Además, se pueden obtener versiones de demostración de diferentes herramientas que comercializan, siendo Rational Rose la más difundida.
- **R. S. Pressman & Associates, Inc.**
 - <http://www.rspa.com>.
 - Bajo la dirección de Roger S. Pressman y la difusión internacional de su afamado libro sobre Ingeniería del Software, aparece una empresa de consultoría en Ingeniería del Software.
 - Lo más interesante que ofrece esta dirección es un portal que da entrada a otras fuentes de información relacionadas con cada uno de los capítulos tratados en su libro.

- **Software Engineering Institute (SEI)**
 - <http://www.sei.cmu.edu>.
 - El Instituto de Ingeniería del Software en la Universidad Carnegie Mellon, es uno de los lugares más activos en pro de la Ingeniería del Software.
 - Se pueden encontrar documentos asociados a módulos curriculares en Ingeniería del Software, informes técnicos sobre diferentes áreas de la Ingeniería del Software, una revista en línea...
 - Incluye además enlaces a otras organizaciones relacionadas con la Ingeniería del Software.
- **The World Wide Web Consortium**
 - <http://www.w3.org/>.
 - Sitio central donde se recogen las especificaciones de los diversos lenguajes relacionados con la Web (HTML, XML, RDF, SOAP...).
- **The World Wide Web Virtual Library Formal Methods**
 - <http://www.afm.sbu.ac.uk>.
 - Sitio especializado en información sobre métodos formales.
- **UML Resource Center**
 - <http://www-306.ibm.com/software/rational/uml/>.
 - Sitio donde se centraliza gran parte de la información sobre UML.
- **Unified Modeling Language (UML) Dictionary**
 - <http://usecasedriven.com/UML.htm>.
 - Diccionario sobre términos de UML.
- **Webliography Software Engineering**
 - http://polaris.umuc.edu/~skerby/help/wbib_swe.htm.
 - Colección de enlaces sobre Ingeniería del Software.
- **WWW Virtual Library - Software Engineering**
 - <http://mingo.info-science.uiowa.edu/soft-eng>.
 - Colección de enlaces sobre Ingeniería del Software.

8.5. Herramientas CASE

- **ArgoUML**
 - <http://argouml.tigris.org>.
- **Enterprise Architect**
 - <http://www.deiser.com>.
- **Left CASE**
 - <http://zarza.fis.usal.es>.

- **Microsoft Visio**
 - <http://www.microsoft.com/office/visio>.
- **PowerDesigner**
 - <http://www.sybase.com/products/developmentintegration/powerdesigner>.
- **Rational Rose**
 - <http://www-306.ibm.com/software/rational/>.
- **REM**
 - http://www.lsi.us.es/descargas/descarga_programas.php?id=3.
- **Studio CASE**
 - <http://zarza.fis.usal.es/~fgarcia>.
- **Visual Paradigm**
 - <http://www.visual-paradigm.com/>.
- **Visual UML**
 - <http://www.visualobjectmodelers.com/products.htm>.

9. Evaluación de los procesos y resultados de aprendizaje

9.1. Sistema de evaluación

La evaluación de esta asignatura se divide en dos partes diferenciadas:

- **Parte de Teoría** (50% de la nota final).
 - Un examen final que consta de dos partes que hay que aprobar independientemente:
 - Un test.
 - Un conjunto de supuestos teórico/prácticos.
- **Parte Práctica** (50% de la nota final).
 - Práctica realizada en grupos de trabajo.
 - Se realizará una defensa de dicho trabajo.
 - Todo grupo tiene derecho a dos defensas
 - La parte práctica se guardará durante el curso académico actual.

La nota final de las asignaturas se calculará conforme a la siguiente fórmula:

Si (Teoría $\geq 5,0$) y (Práctica $\geq 5,0$)

Nota Final = (Teoría*0,5) + (Práctica*0,5)

Si no



Fin si

9.2. Criterios de evaluación

La calificación se hará conforme a las siguientes pautas:

Matrícula de honor

- La nota final es superior o igual a 9,75 puntos
- El conocimiento y la comprensión de la materia se extienden más allá del trabajo cubierto por el programa.
- La comprensión conceptual es sobresaliente.
- Los problemas relacionados con la asignatura son resueltos con soltura y se es capaz de razonar alternativas de solución estableciendo comparativas entre ellas.

Sobresaliente

- La nota final es superior o igual a 8,5 puntos
- El conocimiento y la comprensión de la materia son muy satisfactorios.
- La comprensión conceptual es sobresaliente.
- Los problemas relacionados con la asignatura son resueltos con soltura y se es capaz de razonar alternativas de solución.

Notable

- La nota final es superior o igual a 6,75 puntos
- El conocimiento y la comprensión de la materia son satisfactorios.
- La comprensión conceptual es notable.
- Los problemas relacionados con la asignatura son resueltos con soltura.

Aprobado

- La nota final es superior o igual a 5 puntos
- El conocimiento y la comprensión de la materia son básicos.
- La comprensión conceptual es suficiente.
- Los problemas relacionados con la asignatura son resueltos con éxito razonable.

Suspenso

- La nota final es inferior a 5 puntos
- El conocimiento y la comprensión de la materia no son aceptables.
- La comprensión conceptual es insuficiente.
- Los problemas relacionados con la asignatura no se resuelven de forma adecuada.

10. Análisis de coherencia de la guía docente

Este análisis permite condensar las diferentes relaciones existentes entre los objetivos y competencias a desarrollar con el plan de trabajo del alumnado y el sistema de evaluación empleado. El análisis se ha estructurado en tres tablas separadas en función del tipo de objetivo y competencia analizados en cada caso: instrumental, interpersonal y sistémico.

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS		BLOQUES DE CONTENIDOS			PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Instrumentales		Unidad I	Unidad II	Unidad III		
O11 O12	CIC1 CIC4-CIC10		1			Enseñanza presencial (Clases magistrales) Enseñanza no presencial (Revisión bibliográfica/Consulta de recursos en Internet) Tutorías	Procedimientos Examen Criterios Grado de comprensión y aplicación de los conceptos básicos de la Ingeniería del Software y del proceso de Software
O13 O15 O16 O17 O18	CIC11-CIC13			2		Enseñanza presencial (Clases magistrales/Clases de prácticas/Defensa de práctica/Exposición de ejercicios) Enseñanza no presencial (Trabajo en grupos/Realización de la práctica obligatoria/Revisión bibliográfica/Consulta de recursos en Internet) Tutorías	Procedimientos Ejercicios Práctica obligatoria Examen Criterios Grado de comprensión y aplicación del concepto de requisito
O19	CIC14-CIC16			3		Enseñanza presencial (Clases magistrales/Clases de prácticas/Defensa de práctica/Exposición de ejercicios) Enseñanza no presencial (Trabajo en grupos/Realización de la práctica obligatoria/Revisión bibliográfica/Consulta de recursos en Internet) Tutorías	Procedimientos Ejercicios Práctica obligatoria Examen Criterios Grado de comprensión y aplicación de un método estructurado
O111 O116	CIC16 CIC17-CIC20			4		Enseñanza presencial (Clases magistrales/Talleres de prácticas/Defensa de práctica) Enseñanza no presencial (Trabajo en grupos/Realización de la práctica obligatoria/Revisión bibliográfica/Consulta de recursos en Internet) Tutorías	Procedimientos Examen Práctica obligatoria Criterios Grado de comprensión y aplicación del concepto de proceso software

OI10 OI17 OI20	CIC21-CIC27			5	Enseñanza presencial (Clases magistrales/Talleres de prácticas/Defensa de práctica/Exposición de ejercicios) Enseñanza no presencial (Trabajo en grupos/Realización de la práctica obligatoria/ Realización de ejercicios/Revisión bibliográfica/Consulta de recursos en Internet) Tutorías	Procedimientos Ejercicios Examen Práctica obligatoria Criterios Grado de comprensión y aplicación del modelo objeto
Elementos transversales	CIM1-CIM6 CIT1-CIT4 CIL1-CIL2	1	2,3,4	5	Enseñanza presencial (Clases magistrales/Talleres de prácticas/Defensa de práctica/Exposición de ejercicios) Enseñanza no presencial (Trabajo en grupos/Realización de la práctica obligatoria/Realización de ejercicios/Revisión bibliográfica/Consulta de recursos en Internet) Tutorías	Procedimientos Ejercicios Examen Práctica obligatoria Criterios Grado de destreza en la competencias transversales

Ingeniería del Software I

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS		BLOQUES DE CONTENIDOS			PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Instrumentales		Unidad I	Unidad II	Unidad III		
OI10	CIC28-CIC32	1				Enseñanza presencial (Clases magistrales/Clases de prácticas) Enseñanza no presencial (Revisión bibliográfica/Consulta de recursos en Internet) Tutorías	Procedimientos Ejercicios Examen Criterios Grado de comprensión y aplicación del modelo objeto
OI19	CIC33-CIC36		2			Enseñanza presencial (Clases magistrales/Clases de prácticas/Defensa de práctica/Exposición de ejercicios) Enseñanza no presencial (Trabajo en grupos/Realización de la práctica obligatoria/Revisión bibliográfica/Consulta de recursos en Internet) Tutorías	Procedimientos Ejercicios Examen Práctica obligatoria Criterios Grado de comprensión y aplicación de las técnicas y herramientas de la Ingeniería del Software

OI16 OI18 OI22	CIC37-CIC39		3		<p>Enseñanza presencial (Clases magistrales/Clases de prácticas/Defensa de práctica/Exposición de ejercicios)</p> <p>Enseñanza no presencial (Trabajo en grupos/Realización de la práctica obligatoria/Revisión bibliográfica/Consulta de recursos en Internet)</p> <p>Tutorías</p>	<p>Procedimientos</p> <p>Práctica obligatoria</p> <p>Examen</p> <p>Criterios</p> <p>Grado de comprensión y aplicación del desarrollo de un proyecto software</p>
OI13 OI14 OI21	CIC37-CIC39		4		<p>Enseñanza presencial (Clases magistrales/Talleres de prácticas/Defensa de práctica)</p> <p>Enseñanza no presencial (Trabajo en grupos/Realización de la práctica obligatoria/Revisión bibliográfica/Consulta de recursos en Internet)</p> <p>Tutorías</p>	<p>Procedimientos</p> <p>Examen</p> <p>Práctica obligatoria</p> <p>Criterios</p> <p>Grado de comprensión y aplicación del concepto de proceso software</p>
OI14 OI15 OI22	CIC37-CIC39			5	<p>Enseñanza presencial (Clases magistrales/Talleres de prácticas/Defensa de práctica/Exposición de ejercicios)</p> <p>Enseñanza no presencial (Trabajo en grupos/Realización de la práctica obligatoria/ Realización de ejercicios/Revisión bibliográfica/Consulta de recursos en Internet)</p> <p>Tutorías</p>	<p>Procedimientos</p> <p>Examen</p> <p>Práctica obligatoria</p> <p>Criterios</p> <p>Grado de comprensión y aplicación de los métodos aplicados para el mantenimiento y la evolución del software</p>
Elementos transversales	CIM1-CIM6 CIT1-CIT4 CIL1-CIL2	1	2,3,4	5	<p>Enseñanza presencial (Clases magistrales/Talleres de prácticas/Defensa de práctica/Exposición de ejercicios)</p> <p>Enseñanza no presencial (Trabajo en grupos/Realización de la práctica obligatoria/Realización de ejercicios/Revisión bibliográfica/Consulta de recursos en Internet)</p> <p>Tutorías</p>	<p>Procedimientos</p> <p>Ejercicios</p> <p>Examen</p> <p>Talleres</p> <p>Práctica obligatoria</p> <p>Criterios</p> <p>Grado de destreza en la competencias transversales</p>

Ingeniería del Software II

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS	BLOQUES DE CONTENIDOS

	Interpersonales	Unidad I	Unidad II	Unidad III	PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
OIP1 OIP2	CIPTC1 CIPTC2 CIPTC3		2,3,4		Enseñanza presencial (Práctica obligatoria/Exposición de ejercicios)	Procedimientos Actividades en grupo Criterios Grado de destreza en trabajos en grupo
OIP3	CIPTR1-CIPTR4	2	3	4,6	Enseñanza presencial (Práctica obligatoria /Defensa de práctica/Exposición de ejercicios)	Procedimientos Actividades en grupo Criterios Grado de destreza en trabajos en grupo

Ingeniería del Software I

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS		BLOQUES DE CONTENIDOS			PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Interpersonales		Unidad I	Unidad II	Unidad III		
OIP1 OIP2 OIP3	CIPTC1 CIPTC2 CIPTC3		1	2	3,4,5	Enseñanza presencial (Práctica obligatoria/Exposición de ejercicios)	Procedimientos Actividades en grupo Criterios Grado de destreza en trabajos en grupo
OIP3	CIPTR1-CIPTR4			2	3,4,5	Enseñanza presencial (Talleres de prácticas/Defensa de práctica/Exposición de ejercicios)	Procedimientos Actividades en grupo Criterios Grado de destreza en trabajos en grupo

Ingeniería del Software II

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS		BLOQUES DE CONTENIDOS			PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Sistémicas		Unidad I	Unidad II	Unidad III		
OS1 OS2 OS3 OS4 OS5	CS1	1	2,3,4	5	<p>Enseñanza presencial (Clases magistrales/Clases prácticas/Defensa de práctica/Exposición de ejercicios)</p> <p>Enseñanza no presencial (Trabajo en grupos/Realización de la práctica obligatoria/Realización de ejercicios/Revisión bibliográfica/Consulta de recursos en Internet)</p> <p>Tutorías</p>	<p>Procedimientos</p> <p>Ejercicios</p> <p>Examen</p> <p>Práctica obligatoria</p> <p>Trabajos y prácticas de otras asignaturas</p> <p>Proyecto fin de carrera</p> <p>Criterios</p> <p>Nivel de actuación en destrezas transferibles</p>	
OS1 OS2 OS3 OS4 OS5	CS2 CS5	1	2,3,4		<p>Enseñanza presencial (Clases prácticas/Defensa de práctica/Exposición de ejercicios)</p> <p>Enseñanza no presencial (Realización de ejercicios/Revisión bibliográfica/Consulta de recursos en Internet)</p> <p>Tutorías</p>	<p>Procedimientos</p> <p>Prácticas</p> <p>Trabajos en grupo</p> <p>Ejercicios</p> <p>Criterios</p> <p>Nivel de precisión y eficiencia relacionados con la resolución de problemas nuevos</p>	
OS1 OS2 OS3 OS4 OS5	CS3 CS4		2,3,4		<p>Enseñanza presencial (Clases prácticas/Defensa de práctica/Exposición de ejercicios)</p> <p>Enseñanza no presencial (Realización de ejercicios/Revisión bibliográfica/Consulta de recursos en Internet)</p> <p>Tutorías</p>	<p>Procedimientos</p> <p>Prácticas</p> <p>Trabajos en grupo</p> <p>Ejercicios</p> <p>Criterios</p> <p>Grado de análisis y evaluación de los procedimientos relacionados con la resolución de problemas</p>	

Ingeniería del Software I

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS		BLOQUES DE CONTENIDOS			PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Sistémicas		Unidad I	Unidad II	Unidad III		
OS1	CS1	1	2	3,4,5	<p>Enseñanza presencial (Clases magistrales/Clases Prácticas/Defensa de</p>	<p>Procedimientos</p>	

OS2					práctica/Exposición de ejercicios)	Ejercicios
OS3					Enseñanza no presencial (Trabajo en grupos/Realización de la práctica obligatoria/Realización de ejercicios/Revisión bibliográfica/Consulta de recursos en Internet)	Examen
OS4					Tutorías	Práctica obligatoria
OS5						Trabajos y prácticas de otras asignaturas
						Proyecto fin de carrera
						Criterios
						Nivel de actuación en destrezas transferibles
OS1					Enseñanza presencial (Clases Prácticas /Defensa de práctica/Exposición de ejercicios)	Procedimientos
OS2	CS2		2	3,4,5	Enseñanza no presencial (Realización de ejercicios/Revisión bibliográfica/Consulta de recursos en Internet)	Prácticas
OS3	CS5				Tutorías	Trabajos en grupo
OS4						Ejercicios
OS5						Criterios
						Nivel de precisión y eficiencia relacionados con la resolución de problemas nuevos
OS1					Enseñanza presencial (Clases prácticas/Defensa de práctica/Exposición de ejercicios)	Procedimientos
OS2	CS3		2	3,4,5	Enseñanza no presencial (Realización de ejercicios/Revisión bibliográfica/Consulta de recursos en Internet)	Prácticas
OS3	CS4				Tutorías	Trabajos en grupo
OS4						Ejercicios
OS5						Criterios
						Grado de análisis y evaluación de los procedimientos relacionados con la resolución de problemas

Ingeniería del Software II

Guía Docente de Ingeniería del Software II

Versión 1.0 – 15-05-2006

Jaime Calvo Gallego

Departamento de Informática y Automática
E. P. S. de Zamora – Universidad de Salamanca
Av. Requejo, 33, 46022, Zamora, Spain
Tfno. +34 980 545000. Ext. 3636
jaime.calvo@usal.es

1. Contexto

1.1. Perfil de los créditos y su adecuación al perfil profesional y académico de la titulación

Para un Ingeniero Técnico en Informática de Gestión es imprescindible tener unos conocimientos robustos de cómo desarrollar un sistema software, teniendo en cuenta todas las fases de su ciclo de vida, desde que surge la necesidad de realizar dicho sistema hasta su implementación.

La Ingeniería del Software II como asignatura dentro del Plan de Estudios actual de la Ingeniería Técnica en Informática de Gestión, es troncal y consta de 6 créditos LRU (3 teóricos y 3 prácticos). Su cometido es presentar las actividades técnicas e ingenieriles que se llevan a cabo en el ciclo de vida de un producto software. Además, describe los problemas, métodos y tecnologías asociadas con la Ingeniería del Software, en concreto con el desarrollo de proyectos de software y específicamente, con las partes de gestión, control de la calidad, y evolución y mantenimiento del software, entre otras. Por otra parte, se debe afianzar al alumno en la sistematización de la creación del software mediante el uso de métodos de ingeniería del software, tanto desde el punto de vista estructurado, típicamente tomando el método de Yourdon como referencia principal, como desde un punto de vista orientado a objetos, tomando UML (*Unified Modeling Language*) y el Proceso Unificado como bases fundamentales. En esta asignatura es demasiado ambicioso afrontar en profundidad los métodos de análisis y diseño, por ello se enfoca mayormente desde el punto de vista del el uso de métodos de ingeniería del software y el desarrollo de proyectos software; partiendo de la base adquirida en la asignatura de Ingeniería del Software I. Para la parte práctica de esta asignatura se debe hacer uso de herramientas CASE (*Computer Aided/Assisted Software Engineering*).

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

1.2. Ubicación y relaciones en el plan de estudios

La asignatura de Ingeniería del Software II se imparte en el primer cuatrimestre del tercer curso de la Ingeniería Técnica en Informática de Gestión, cuyo Plan de Estudios actual data de 2003. Esta asignatura se relaciona directamente con otras de este mismo Plan de Estudios, pero además con otras de la titulación de segundo ciclo Ingeniería en Informática, cuyo Plan de Estudio data de 1998.

Las dependencias e interrelaciones entre estas asignaturas se muestran en la Figura 1. En el establecimiento de estas dependencias se ha tenido en cuenta el factor tiempo, que claramente establece el orden lógico en el que se van a cursar las asignaturas.

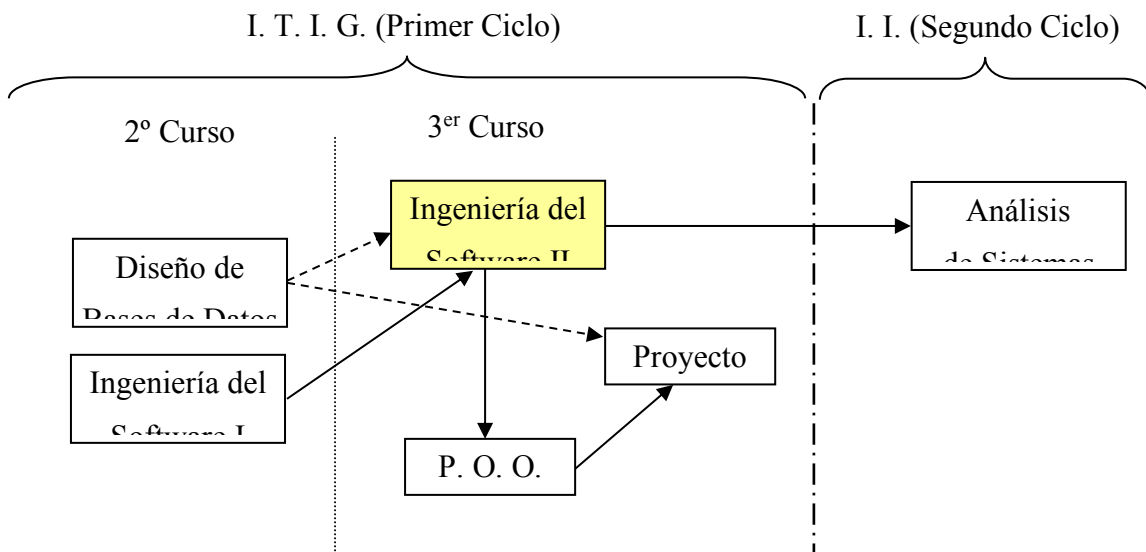


Figura 1. Relaciones de la Ingeniería del Software II con otras asignaturas

2. Objetivos

2.1. Objetivos instrumentales generales

- OI1: Aplicar los conocimientos adquiridos mediante la resolución de problemas y una práctica obligatoria.
- OI2: Utilizar con fluidez herramientas CASE.
- OI3: Adquirir y emplear un buen lenguaje formal, tanto oral como escrito, siendo riguroso en las explicaciones de cualquier proceso.
- OI4: Reforzar el conocimiento y utilización de la terminología utilizada en Ingeniería del Software.
- OI5: Adquirir un buen manejo de la bibliografía recomendada en la asignatura, de forma que se potencia la autosuficiencia a la hora de completar la formación.
- OI6: Comprender el ámbito de la Ingeniería del Software dentro de la Ingeniería Técnica Informática de Gestión y dentro de los perfiles profesionales.
- OI7: Tomar conciencia de las implicaciones del trabajo de ingeniero de software, tanto individualmente como formando parte de un equipo
- OI8: Reforzar la visión inicial del campo de la Ingeniería del Software.
- OI9: Tener una perspectiva global del proceso asociado al ciclo de vida de un producto software.
- OI10: Conocer los problemas, principios, métodos y herramientas propios de la Ingeniería del Software.
- OI11: Conocer los elementos básicos de un modelo objeto, soportado en el lenguaje unificado de modelado (UML).

- OI12: Conocer la importancia y la influencia de los requisitos en el éxito de un proceso software.
- OI13: Aplicar las técnicas básicas de ingeniería de requisitos.
- OI14: Conocer los principios de un método de análisis y diseño orientado a objetos.
- OI15: Conocer los principios de un método de análisis y diseño estructurado.
- OI16: Adquirir la capacidad de crear documentaciones técnicas completas, correctas y legibles.

2.2. Objetivos interpersonales generales

- OIP1: Destrezas para la participación responsable: capacidad de coordinación, asistencia, contribuciones al grupo, etc.
- OIP2: Capacidad de trabajar en equipo adquiriendo y mejorando las habilidades sociales y la inteligencia emocional.
- OIP3: Comprometerse de forma ética con el trabajo, con el resto de los integrantes del grupo y consigo mismo.

2.3. Objetivos sistémicos generales

- OS1: Capacidad de integrar los conocimientos y destrezas prácticas de las diferentes asignaturas del plan de estudio para resolver situaciones reales relacionadas con la Ingeniería Técnica Informática de Gestión, así como con otras disciplinas relacionadas.
- OS2: Reforzar el hábito de plantearse interrogantes. Ante un problema preguntarse por el número de soluciones, la relación entre ellas, cómo afectaría a las condiciones iniciales alguna modificación, etc.
- OS3: Capacidad de aplicar y relacionar, de forma autónoma, los contenidos de Ingeniería del Software de forma interdisciplinar.
- OS4: Adquirir una comprensión del método científico, a través de las diversas actividades realizadas en la asignatura de Ingeniería del Software II, y asimilar su importancia como manera de pensar y actuar en la labor de científico e ingeniero, fomentando la capacidad de abstracción y el espíritu crítico.
- OS5: Desarrollar la madurez necesaria en el proceso de abstracción para abordar problemas reales y plantear modelos y soluciones de forma razonada y correcta.
- OS6: Reforzar el hábito de desarrollar diferentes alternativas, cuestionando las características, riesgos y viabilidad de cada una, para cada problema planteado.

3. Competencias

3.1. Competencias instrumentales

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

3.1.1. Habilidades cognitivas

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

Generales

- CIC1: Conocer y entender las relaciones que mantiene la asignatura con el resto de asignaturas de la titulación a partir de la introducción explícita de ejemplos y comentarios, aprovechando puntos relevantes del temario.
- CIC2: Conocer y comprender algunas de las capacidades, aptitudes y conocimientos que la asignatura aporta para el desarrollo de los diversos perfiles profesionales.
- CIC3: Ser capaz de crear documentaciones técnicas completas, correctas y legibles.

Unidad Didáctica I: Conceptos básicos

- CIC4: Conocer el alcance de un producto software. Razonar acerca de la calidad del software.
- CIC5: Entender el método de resolución de problemas que sustenta la Ingeniería del Software.
- CIC6: Diferenciar las perspectivas propias del dominio de problema y del dominio de la solución.
- CIC7: Entender el concepto de proceso software y conocer los principales modelos de proceso o paradigmas de ciclo de vida del software. Razonar acerca de su necesidad en la producción de software de calidad.
- CIC8: Entender la noción de metodología y conocer los diferentes tipos existentes.
- CIC9: Conocer la estructura de un proceso iterativo e incremental, que esté dirigido por casos de uso.
- CIC10: Conocer el concepto y alcance de las herramientas CASE.
- CIC11: Comprender los elementos que conforman un modelo objeto.
- CIC12: Conocer y comprender el lenguaje de modelado unificado (UML), con especial atención a las vistas estática y de interacción.
- CIC13: Ser capaz de aplicar las vistas estáticas y de interacción en modelos concretos y de complejidad baja y media, correspondientes a diversos artefactos software.
- CIC14: Conocer y comprender la vista de casos de uso de UML.
- CIC15: Ser capaz de aplicar la vista de casos de uso en proyectos software de complejidad baja y media.

Unidad Didáctica II: Gestión de proyectos

- CIC16: Conocer los conceptos implicados en la gestión de un proyecto de construcción del software.
- CIC17: Ser capaz de trazar las actividades para la planificación de un proyecto de construcción del software.

Unidad Didáctica III: Control de calidad del software

- CIC18: Conocer los conceptos implicados en la calidad del software.
- CIC19: Conocer las tendencias de los sistemas de calidad.
- CIC20: Conocer los métodos para garantizar de calidad del software.
- CIC21: Resaltar la importancia de las revisiones del software y de las revisiones técnicas formales.
- CIC22: Conocer lo estándares de calidad.
- CIC23: Ser capaz de entender y aplicar planes de calidad del software.

Unidad Didáctica IV: Evolución y mantenimiento del software

- CIC24: Tomar conciencia de la evolución y mantenimiento del software, y ser capaz de trazar planificaciones para tales fines.
- CIC25: Ser capaz de planificar actividades de mantenimiento.
- CIC26: Conocer las líneas principales actuación en el proceso de mantenimiento software, de asegurar la calidad del mismo, así como una planificar una posible acción preventiva de los problemas que pudieran surgir.
- CIC27: Ser capaz de plasmar y planificar los costes de mantenimiento.
- CIC28: Conocer las herramientas y técnicas.
- CIC29: Introducir el desarrollo de la capacidad de para efectuar procesos de ingeniería inversa y reingeniería.

3.1.2. Capacidades metodológicas

- CIM1: Ser capaz de tomar decisiones de manera razonada.
- CIM2: Tener capacidad de análisis y síntesis.
- CIM3: Ser capaz de manejar bibliografía relacionada con la Ingeniería del Software.
- CIM4: Ser capaz de documentar la solución a un problema dado en las diferentes fases del ciclo de vida con los diferentes niveles de abstracción y aproximación que esto requiere.
- CIM5: Ser capaz de aplicar correctamente y manejar con comodidad los diferentes elementos de los que dispone un lenguaje de modelado concreto, ya sea orientado a objetos u orientado a procesos.
- CIM6: Ser capaz de modelar un sistema de información desde las diferentes perspectivas que exige un determinado proceso.

3.1.3. Destrezas tecnológicas

- CIT1: Habilidades básicas de navegación por la Web y uso del resto de servicios de red para la obtención y manejo de la información relacionada con la asignatura.
- CIT2: Manejar las operaciones básicas y de administración de diferentes sistemas operativos para la instalación de las diferentes herramientas CASE a utilizar en la asignatura.
- CIT3: Manejar con fluidez diferentes herramientas CASE.

CIT4: Manejo avanzado de un sistema de procesamiento de textos para la realización de los informes de las prácticas y la documentación de la práctica obligatoria.

3.1.4. Destrezas lingüísticas

CIL1: Adquirir y utilizar con fluidez un buen lenguaje científico, tanto oral como escrito, siendo riguroso en las explicaciones de cualquier proceso dentro de la asignatura.

CIL2: Conocer y utilizar la terminología usual de la Ingeniería del Software, tanto en español como en inglés.

3.2. Competencias interpersonales

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

3.2.1. Competencias para tareas colaborativas

CIPTC1: Ser capaz de realizar de trabajar en equipo para resolver los problemas de modelado enunciados en los talleres/laboratorios prácticos de la asignatura.

CIPTC2: Ser capaz de presentar en público una solución a un problema planteado y mantener un debate con el resto de la clase sobre la solución planteada, para así buscar colaborativamente la mejor solución al problema.

CIPTC3: Ser capaz de realizar y defender la práctica obligatoria de la asignatura en equipo.

3.2.2. Compromiso con el trabajo

CIPTR1: Se ha de definir un plan de trabajo en el que el volumen de trabajo de todos los miembros del equipo sea similar.

CIPTR2: Una vez finalizado el trabajo, todos los miembros del grupo deben conocer en profundidad todo el desarrollo realizado.

CIPTR3: Se debe cumplir el plazo de entrega de los trabajos.

CIPTR4: Se debe adquirir un compromiso ético entre todos los componentes del grupo.

3.3. Competencias sistémicas

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

CS1: Capacidad de aplicar los conocimientos, métodos y herramientas vistos en la asignatura de Ingeniería del Software II a situaciones y problemas concretos del área de la Ingeniería Informática y de otras disciplinas relacionadas.

CS2: Capacidad de aprender y aplicar, de forma autónoma e interdisciplinar, nuevos conceptos y métodos relacionados con cada asignatura.

CS3: Motivación por la calidad y por la creatividad.

CS4: Capacidad de adoptar el proceso marcado por el método científico y de ingeniería en el planteamiento y realización de trabajos diversos, tanto a nivel académico como profesional.

CS5: Capacidad de asimilación y adaptación a la evolución del estado del arte en el ámbito de la Ingeniería del Software y de la Ingeniería Informática como profesión.

4. Prerrequisitos

4.1. Competencias y contenidos mínimos

- Conocer los elementos básicos de un lenguaje de programación de 3ª generación.
- Conocer las principales formas de representar los algoritmos.
- Conocer el concepto de tipo abstracto de datos.
- Conocer los principios fundamentales del diseño de estructuras de datos.
- Conocer los principios fundamentales del modelado conceptual de datos mediante el diagrama entidad/relación.
- Conocer los mecanismos de transformación de un modelo conceptual de datos a un modelo lógico de datos.
- Conocer los conceptos básicos de Ingeniería del Software.

4.2. Plan de trabajo para la consecución de los prerrequisitos

Los prerrequisitos para el estudio y entendimiento de la asignatura Ingeniería del Software II se cubren fundamentalmente en las asignaturas de programación del primer curso, y en las asignaturas de Ingeniería del Software I y Bases de Datos del segundo curso.

5. Temario

5.1. Unidades didácticas

Unidad Didáctica I: Conceptos básicos

Tema 0: Sumario de la asignatura

Tema 1. Lenguaje unificado de modelado.

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

Unidad Didáctica II: Gestión de proyectos

Tema 3. Gestión de proyectos.

Unidad Didáctica III: Control de calidad del software

Tema 4. Control de calidad del software.

Unidad Didáctica IV: Evolución y mantenimiento del software

Tema 5. Evolución y mantenimiento del software

5.2. Desarrollo de las unidades didácticas

Unidad Didáctica I: Conceptos básicos

Tema 0: Sumario de la asignatura

0.1. Motivación.

0.2. Enfoque de ingeniería en el desarrollo del software.

- 0.3. Docencia de la Ingeniería del Software II.
- 0.4. Asignatura Ingeniería del Software II en ITIG.
 - 0.4.1. Objetivos.
 - 0.4.2. Temario.
 - 0.4.3. Evaluación.
 - 0.4.4. Bibliografía básica recomendada.
 - 0.4.5. Otras fuentes.
 - 0.4.6. Enlaces.
 - 0.4.7. Herramientas CASE.
 - 0.4.8. Tutorías.

0.5. Prerrequisitos: Conocimiento de los conceptos desarrollados en la asignatura Ingeniería del Software I.

Tema 1. Modelo Objeto. Una descripción de UML

- 1.1. Introducción a la orientación a objetos
- 1.2. Modelo objeto
- 1.3. ¿Qué es UML?
- 1.4. Historia de UML
- 1.5. Visión global de UML
- 1.6. Vista estática
- 1.7. Vista de interacción
- 1.8. Vista de casos de uso
- 1.9. Vista de máquina de estados
- 1.10. Vista de actividad
- 1.11. Vistas físicas

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

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

Unidad Didáctica II: Gestión de proyectos

Tema 3. Gestión de proyectos

- 3.1. Introducción.
- 2. Gestión de un proyecto de construcción del software.
- 3. Actividades para la planificación de un proyecto de construcción del software.

Unidad Didáctica III: Control de calidad del software

Tema 4. Control de calidad del software

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

Unidad Didáctica IV: Evolución y mantenimiento del software

Tema 5. Evolución y mantenimiento del software

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

6. Metodología y estrategias de aprendizaje

6.1. Metodología docente

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

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

- Convertirse en el responsable de su aprendizaje, debiendo gestionar y controlar la forma en que se produce.
- Desarrollar un espíritu crítico y una actitud abierta ante los cambios científico-técnicos de su especialidad, con curiosidad intelectual y rigor científico.
- Fomentar actitudes y adquirir técnicas para trabajar en equipo con eficacia.
- Basar en principios deontológicos su futura actividad profesional.

- Valorar el sentido humanístico de la ciencia y de la técnica como resultado del esfuerzo de generaciones anteriores, que debe servir para resolver los problemas del hombre actual y futuro.
- Estimular el perfeccionamiento profesional y la formación continua a lo largo de la vida.

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

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

- *Clases de teoría con apoyo de material audiovisual.* En estas clases se presentarán los contenidos básicos de un cierto tema. Las clases comenzarán con una breve introducción de los contenidos que se pretenden transmitir en la clase, así como con un breve comentario a los conceptos vistos en clases anteriores y que sirven de enlace a los que se pretenden desarrollar. El desarrollo de la clase se llevará a cabo con medios audiovisuales, textos, transparencias, etc. que permitan un adecuado nivel de motivación e interés en los alumnos. Se debe intentar motivar a los alumnos a intervenir en cualquier momento en las clases para hacer éstas más dinámicas y facilitar el aprendizaje. Es importante intentar terminar la exposición con las conclusiones más relevantes del tema tratado.
- *Talleres/laboratorios de prácticas.* Las clases prácticas presenciales estarán dedicadas a la resolución colaborativa de problemas de modelado.
- *Práctica obligatoria.* Los grupos de prácticas deberán realizar una especificación de requisitos del software (completada con algunas partes de diseño) sobre un tema libre (previamente acordado con profesor y contando con el visto bueno de éste para poder empezar) que ellos han debido de cerrar. Esa práctica tendrá que defenderse en grupo y representará una importante parte de la nota final de la asignatura.
- *Entrega de ejercicios.* Un alumno individualmente puede entregar ejercicios resueltos por él, cuyos enunciados debe encontrar en la bibliografía (sin que estén resueltos). Éstos serán discutidos con el profesor en horas de tutoría.
- *Tutorías.* El alumnado tiene a su disposición seis horas de tutorías a la semana en las que puede consultar cualquier duda relacionada con los contenidos, organización y planificación de la asignatura. Las tutorías pueden ser individualizadas, pero se admite tutorías grupales para resolver problemas relacionados con las actividades a realizar en grupo.
- *Página web.* Se convierte en el vehículo de comunicación principal y registro de información de la asignatura.

6.2. Estrategias de aprendizaje

Se detalla las actuaciones concretas a realizar para la aplicación y aprovechamiento de la metodología propuesta. Se estructura en las siguientes fases:

6.2.1. Recopilación de la documentación de la asignatura

- Los alumnos dispondrán en la página web de la asignatura de toda la información y normativa relacionada con la asignatura: temario, criterios de evaluación, bibliografía, apuntes, enunciados de prácticas, trabajos, enlaces de interés, lecturas complementarias, avisos...
- El profesor mantiene actualizada la información de esta página para que se convierta en un vehículo de comunicación con los alumnos.
- Los alumnos deben conocer y manejar con fluidez toda la información y la documentación que se integra en la página de la asignatura.
- Asiduamente, el alumno debe consultar la página para estar actualizado.

6.2.2. Planificación de las clases teóricas

- Las clases teóricas se dedicarán a la presentación de contenidos y a la discusión sobre las dudas que surjan durante las exposiciones.
- Las clases se desarrollarán según el siguiente esquema:
 1. El profesor presenta los objetivos a conseguir y el contexto en el que se va a desarrollar la clase.
 2. Se presentan los conceptos.
 3. Se aclaran las dudas cuando éstas surjan. El profesor puede incentivar el debate con preguntas para hacer la clase más participativa.
 4. El profesor termina con un resumen de los principales conceptos tratados en la clase y puede introducir la siguiente clase.
- El alumno debe haber realizado una lectura previa de los contenidos que se van a tratar en la clase, máxime cuando no todos los contenidos se desarrollan en las clases magistrales.
- Una vez terminada la clase magistral, se debe estudiar de forma autónoma su contenido y en caso de no entender algo intentar aclararlo utilizando la bibliografía recomendada o cualquiera de los materiales adicionales recomendados en la asignatura.

6.2.3. Planificación de las clases prácticas

- El profesor hará público los plazos de entrega con suficiente antelación.
- En el caso de los talleres/laboratorios el grupo de trabajo debe reunirse con antelación para leer y comprender bien el enunciado y realizar su propuesta de solución.
- En el caso de la práctica obligatoria el grupo de trabajo debe leer y entender bien todos los puntos antes de ponerse a llevarla a cabo.
- Durante los talleres/laboratorios el profesor moderará el debate que se origine sobre el problema y la solución presentada, dando argumentos a cada una de las decisiones o alternativas que se planteen.
- Los alumnos deben cumplir el calendario de entrega de prácticas.

- Las prácticas obligatorias se defenderán en grupo.

6.2.4. Evaluación

- Se plantea una forma de evaluación continua.
- Para aquellos alumnos que no superen la evaluación continua o no se quieran o no puedan acogerse a ella existirá una prueba final.

7. Planificación del tiempo y del esfuerzo

La asignatura de Ingeniería del Software II tiene en el plan de estudios actual 6 créditos LRU asignados, 4,5 créditos de teoría y 1,5 créditos de práctica. Para hacer su estudio en ECTS se ha tomado tomando la equivalencia de crédito LRU a crédito ECTS, con implicación de cambio de metodología docente.

Al hacer el estudio de carga de trabajo máxima, se tiene que:

- 6 ECTS -> 150-180 horas de trabajo
- 3cr LRU (teóricos) -> 30 horas presenciales + (30*2) horas de asimilación = 90 horas
- 3cr LRU (prácticos) -> 30 horas presenciales + (30*1,5) horas de asimilación = 75 horas
- $112,5 + 45 = 165$ horas de trabajo
- $150 < 165 < 180$

Dos son los escenarios más probables: el resultante de tomar el crédito ECTS como 30 horas de trabajo y el resultante de tomarlo como 25 horas de trabajo.

Por temas de organización y calendario, se ha optado por partir del escenario que asigna 25 horas de trabajo al ECTS.

Para poder elaborar las guías, cuyo destino final sería la realización de experiencias piloto, dentro de los planes de estudio vigentes, se debe tener en cuenta la definición del crédito del R.D. 1497/1987 y sus posteriores modificaciones. Así:

- La unidad de valoración de las enseñanzas se corresponde a diez horas de enseñanza teórica, práctica o de sus equivalencias entre las que podrán incluirse actividades académicas dirigidas, que habrán de preverse en el correspondiente plan docente junto con los mecanismos y medios objetivos de comprobación de los resultados académicos de las mismas. Todo ello sin perjuicio del cumplimiento del régimen de dedicación del profesorado, de conformidad con el Real Decreto 898/1985, de 30 de abril sobre régimen del profesorado universitario.
- En ningún caso, salvo que se trate de enseñanzas en Universidades a distancia, el porcentaje del crédito correspondiente a las actividades académicas dirigidas será superior al **30%**.

De esta forma, si 6 créditos LRU equivalen a 60 horas, se debe asegurar el 70% como mínimo de presencialidad, esto es, 42 horas entre horas de teoría y de práctica.

En la Tabla 1 se presenta la organización del esfuerzo del alumno para cubrir los 6 ECTS asociados a la asignatura. Por su parte en la Tabla 2 se distribuyen las horas presenciales entre los diferentes temas y el examen.

			A	B	C	D	E
	Técnica	Actividad	Horas equivalentes de clase	Factor de trabajo del alumno [†]	Horas de trabajo personal del alumno	Horas totales (A+C)	ECTS (D ÷ 25)
Teoría	Seguimiento de la parte teórica de las unidades didácticas	Asimila contenidos. Se plantea dudas que planteará a los profesores en las tutorías	34	1,5	51	85	3,4
Práctica	Seguimiento de la parte práctica de las unidades didácticas	Experimenta, practica, modela, simula	8	2	16	24	0,96
Ejercicios de apoyo	Ejercicios de repaso y afianzamiento de lo aprendido. Pueden computar o no para la superación de la asignatura, dependiendo de la tipología de éstos	Realiza, resuelve problemas, tests...	-	-	5	5	0,2
Práctica obligatoria	Práctica en grupo de obligada realización	Realiza una ERS completada con elementos de diseño	-	-	30	30	1,2
Examen	Controles de evaluación continua y/o examen final de la asignatura	Exámenes tipo test, supuestos prácticos...	3			3	0,12
Otras actividades	Tutorías personalizadas y grupales	Recibe orientación personalizada	-	-	2	2	0,08
	Búsquedas en la red, participación en foros especializados...	Busca elementos para completar los contenidos	-	-	1	1	0,04
TOTAL			45		105	150	6

Tabla 1. Escenario ECTS = 25 horas de trabajo

[†] Número de horas dedicadas por el alumno al trabajo personal (organización de apuntes, estudio, documentación...) por cada hora de clase.

NÚMERO DE HORAS PRESENCIALES		
Actividad	Horas Teoría	Horas Prácticas
Unidad Didáctica I: Conceptos básicos		
Tema 0	1	
Tema 1	7	
Tema 2	2	
Unidad Didáctica II: Gestión de proyectos		
Tema 3	11	4
Unidad Didáctica III: Control de calidad del software		
Tema 4	6	2
Unidad Didáctica IV: Evolución y mantenimiento del software		
Tema 5	7	2
Examen	3	
TOTAL: 45	37	8

Tabla 2. Reparto de horas presenciales

8. Bibliografía

8.1. Bibliografía básica

- Booch, G., Rumbaugh, J., Jacobson, I. (1999) *El Lenguaje Unificado de Modelado*. Addison Wesley.
- Jacobson, I., Booch, G., Rumbaugh, J. (2000) *El Proceso Unificado de Desarrollo de Software*. Addison-Wesley.
- Larman, C. (2003) *UML y Patrones*. 2ª Edición. Prentice-Hall.
- Pfleeger, S. L. (2002) *Ingeniería del Software*. Teoría y Práctica. Prentice Hall.
- Piattini, M. G., Calvo-Manzano, J. A., Cervera, J., Fernández, L. (2004) *Análisis y Diseño de Aplicaciones Informáticas de Gestión. Una perspectiva de Ingeniería del Software*. Ra-ma.
- Pressman, R. S. (2005) *Ingeniería del Software: Un Enfoque Práctico*. 6ª Edición. McGraw-Hill.
- Rumbaugh, J., Jacobson, I., Booch, G. (2000) *El Lenguaje Unificado de Modelado. Manual de Referencia*. Addison-Wesley.
- Sommerville, I. (2005) *Ingeniería del Software*. 7ª Edición, Pearson Addison-Wesley.
- Yourdon, E. (1993) *Análisis Estructurado Moderno*. Prentice-Hall Hispanoamericana.

8.2. Bibliografía complementaria

- Gamma, E., Helm, R., Johnson, R., Vlissides, J. (2003) *Patrones de Diseño*. Addison-Wesley.
- Larman, C. (2005) *Applying UML and Patterns. An Introduction to Object-Oriented Analysis and Design and Iterative Development*. 3rd Edition. Prentice Hall.
- Meyer, B. (1999) *Construcción de Software Orientado a Objetos*. 2ª Edición. Prentice Hall.

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

8.3. Otros recursos

- Durán, A., Bernárdez, B. (2002) Metodología para la Elicitación de Requisitos de Sistemas Software (versión 2.3). Informe Técnico LSI-2000-10, Universidad de Sevilla. <http://www.lsi.us.es/~amador>. [Última vez visitado, 22-9-2005].
- Durán, A., Bernárdez, B. (2001) Metodología para el Análisis de Requisitos de Sistemas Software (versión 2.2). Universidad de Sevilla. <http://www.lsi.us.es/~amador>. [Última vez visitado, 22-9-2005].
- OMG (2003) OMG Unified Modeling Language Specification. Version 1.5. Object Management Group Inc. Document formal/03-03-01. March 2003. <http://www.omg.org/docs/formal/03-03-01.pdf>. [Última vez visitado, 22-9-2005]
- OMG (2005) Unified Modeling Language: Superstructure. Version 2.0. Object Management Group Inc. Document formal/05-07-04. August 2005. <http://www.omg.org/cgi-bin/doc?formal/05-07-04>. [Última vez visitado, 22-9-2005].
- Pohl, K. (1997) Requirements Engineering: An Overview. En M. Dekker (Ed.), *Encyclopedia of Computer Science and Technology*, 36. Disponible en <ftp://sunsite.informatik.rwth-aachen.de/pub/CREWS/CREWS-96-02.pdf>. [Última vez visitado, 22-9-2005].

8.4. Enlaces de interés

- **Association for Computing Machinery (ACM)**
 - <http://www.acm.org>.
 - Fundada en 1947 fue la primera sociedad científica y de educación del mundo. El portal de información que presenta es impresionante, tanto en cuanto a enlaces de interés, grupos de trabajo, documentos electrónicos, conferencias como por su biblioteca digital conteniendo revistas y actas de congresos (<http://portal.acm.org>).
- **Cetus Links - Object-Oriented**
 - <http://www.cetus-links.org>.
 - La colección de enlaces más completa sobre Orientación a Objetos. A fecha de 17 de marzo de 2004 se tenían contabilizados 18193 enlaces.
- **Consejo Superior de Informática**
 - <http://www.csi.map.es/>.
 - Web del Consejo Superior de Informática donde, entre otras cosas, se puede obtener la documentación y herramientas sobre Métrica 3 (<http://www.csi.map.es/csi/metrica3/index.html>).
- **DACS (Data & Analysis Center for Software)**
 - <http://www.dacs.dtic.mil>.
 - Sitio perteneciente al DoD de EEUU con numerosa información y enlaces a lugares relacionados con la Ingeniería del Software.
- **European Software Institute (ESI)**
 - <http://www.esi.es>.

- El Instituto del Software Europeo tiene su sede en Bilbao (España). Dispone tanto de documentación privada para los miembros del instituto como documentación pública con los análisis de proyectos, necesidades de empresas y software europeo.
- **Guide to the Software Engineering Body of Knowledge (SWEBOK)**
 - <http://www.swebok.org>.
 - Proyecto para establecer un cuerpo de conocimiento común para la Ingeniería del Software.
- **Institute of Electrical and Electronics Engineers (IEEE)**
 - <http://www.ieee.org>.
 - Otra prestigiosa organización compuesta por diversas sociedades, donde la que más relación tiene con los temas abordados en el presente curso es la IEEE Computer Society (<http://computer.org>).
 - A semejanza de ACM, ofrece información sobre conferencias, estándares, educación y mantiene otra biblioteca digital con revistas y actas de congresos.
- **Object Management Group (OMG)**
 - <http://www.omg.org>.
 - Es un consorcio internacional de industrias con el fin de promover el uso de la Orientación a Objetos en la Ingeniería del Software. A diferencia de organizaciones como ISO o IEEE, OMG desarrolla estándares de “*facto*” como consenso entre las empresas que la forman.
 - Dicho servidor ofrece publicaciones electrónicas y enlaces a estándares y herramientas del sector relacionado con la tecnología de objetos.
 - En <http://www.omg.org/uml> se encuentran las especificaciones de la versión actual (así como de las anteriores) de UML.
- **IBM Rational Software**
 - <http://www-306.ibm.com/software/rational/>.
 - En esta dirección se tiene valiosa información sobre UML y RUP (documentos, informes, artículos, presentaciones, bibliografía recomendada...).
 - Además, se pueden obtener versiones de demostración de diferentes herramientas que comercializan, siendo Rational Rose la más difundida.
- **R. S. Pressman & Associates, Inc.**
 - <http://www.rspa.com>.
 - Bajo la dirección de Roger S. Pressman y la difusión internacional de su afamado libro sobre Ingeniería del Software, aparece una empresa de consultoría en Ingeniería del Software.
 - Lo más interesante que ofrece esta dirección es un portal que da entrada a otras fuentes de información relacionadas con cada uno de los capítulos tratados en su libro.

- **Software Engineering Institute (SEI)**
 - <http://www.sei.cmu.edu>.
 - El Instituto de Ingeniería del Software en la Universidad Carnegie Mellon, es uno de los lugares más activos en pro de la Ingeniería del Software.
 - Se pueden encontrar documentos asociados a módulos curriculares en Ingeniería del Software, informes técnicos sobre diferentes áreas de la Ingeniería del Software, una revista en línea...
 - Incluye además enlaces a otras organizaciones relacionadas con la Ingeniería del Software.
- **The World Wide Web Consortium**
 - <http://www.w3.org/>.
 - Sitio central donde se recogen las especificaciones de los diversos lenguajes relacionados con la Web (HTML, XML, RDF, SOAP...).
- **The World Wide Web Virtual Library Formal Methods**
 - <http://www.afm.sbu.ac.uk>.
 - Sitio especializado en información sobre métodos formales.
- **UML Resource Center**
 - <http://www-306.ibm.com/software/rational/uml/>.
 - Sitio donde se centraliza gran parte de la información sobre UML.
- **Unified Modeling Language (UML) Dictionary**
 - <http://usecasedriven.com/UML.htm>.
 - Diccionario sobre términos de UML.
- **Webliography Software Engineering**
 - http://polaris.umuc.edu/~skerby/help/wbib_swe.htm.
 - Colección de enlaces sobre Ingeniería del Software.
- **WWW Virtual Library - Software Engineering**
 - <http://mingo.info-science.uiowa.edu/soft-eng>.
 - Colección de enlaces sobre Ingeniería del Software.

8.5. Herramientas CASE

- **ArgoUML**
 - <http://argouml.tigris.org>.
- **Enterprise Architect**
 - <http://www.deiser.com>.
- **Left CASE**
 - <http://zarza.fis.usal.es>.

- **Microsoft Visio**
 - <http://www.microsoft.com/office/visio>.
- **PowerDesigner**
 - <http://www.sybase.com/products/developmentintegration/powerdesigner>.
- **Rational Rose**
 - <http://www-306.ibm.com/software/rational/>.
- **REM**
 - http://www.lsi.us.es/descargas/descarga_programas.php?id=3.
- **Studio CASE**
 - <http://zarza.fis.usal.es/~fgarcia>.
- **Visual Paradigm**
 - <http://www.visual-paradigm.com/>.
- **Visual UML**
 - <http://www.visualobjectmodelers.com/products.htm>.

9. Evaluación de los procesos y resultados de aprendizaje

9.1. Sistema de evaluación

La evaluación de esta asignatura se divide en dos partes diferenciadas:

- **Parte de Teoría** (50% de la nota final).
 - Un examen final que consta de dos partes que hay que aprobar independientemente:
 - Un examen de desarrollo y/o de tipo test.
 - Un conjunto de supuestos teórico/prácticos.
 - La parte de teoría se guardará durante el curso académico actual.
- **Parte Práctica** (50% de la nota final).
 - Práctica realizada en grupos de trabajo.
 - Se realizará una defensa de dicho trabajo.
 - Todo grupo tiene derecho a dos defensas, siempre que la primera se haya hecho antes del mes de diciembre.
 - La parte práctica se guardará durante el curso académico actual.

De forma complementaria a este sistema de evaluación tradicional, el alumno se puede acoger opcionalmente a una evaluación continua.

- Se tendrá en cuenta la asistencia y la participación activa en clase.
- Todos los alumnos que se acojan a esta modalidad deberán haber entregado su ficha en el primer mes de clase.
- En relación con la parte del examen de desarrollo y/o de tipo test:
 - Se realizarán pruebas de evaluación de durante las clases de teoría:

- Prueba 1: Unidad II.
- Prueba 2: Unidades III.
- Prueba 3: Unidad IV
- Se eliminará la parte del examen de teoría si se obtiene una calificación superior o igual a 5 en el conjunto de estas pruebas, obteniendo en todas ellas al menos una nota de 4:
 - $\text{Nota test} = (\text{Prueba 1} * 0,4) + (\text{Prueba 2} * 0,3) + (\text{Prueba 3} * 0,3)$
- En relación con la parte del conjunto de supuestos teórico/prácticos del examen de teoría:
 - En los talleres/laboratorios
 - Todos los grupos obligatoriamente entregarán las prácticas y/o trabajos que se precisen, y los informes y/o memorias correspondientes en las fechas indicadas.
 - Si se detecta fraude dará lugar a la invalidación del práctica/trabajo o prácticas/trabajos correspondientes. Para su recuperación se deberá realizar un nuevo supuesto propuesto por el profesor.
 - La entrega de informes será obligatoria y deberá atenerse a las normas de presentación.
 - Todos los grupos presentarán y defenderán obligatoriamente las prácticas y/o trabajos en las fechas que se indiquen.
 - Por la defensa se obtiene entre 0 y 0,5 puntos.
 - La participación activa, acertada y continuada puede aportar hasta 0,5 puntos.
 - Entrega de ejercicios
 - Entregar un mínimo de tres ejercicios de los cuales, como mínimo, habrá un ejercicio de cada una de las unidades didácticas II, III y IV. Los enunciados se obtendrán de libros (y no resueltos en ellos). La revisión de los ejercicios se efectuará en horario de tutorías, y podrá aportar hasta 0,75 puntos

La nota final de la asignatura se calculará conforme a la siguiente fórmula:

Si (Teoría \geq 5,0) y (Práctica \geq 5,0)

Nota Final = (Teoría*0,5) + (Práctica*0,5)

Si no

Suspense

Fin si

9.2. Criterios de evaluación

La calificación se hará conforme a las siguientes pautas:

Matrícula de honor

- La nota final es superior o igual a 9,75 puntos
- El conocimiento y la comprensión de la materia se extienden más allá del trabajo cubierto por el programa.
- La comprensión conceptual es sobresaliente.
- Los problemas relacionados con la asignatura son resueltos con soltura y se es capaz de razonar alternativas de solución estableciendo comparativas entre ellas.
- La participación en las diferentes actividades, incluyendo las clases, sobresale por su corrección y satisfacción.

Sobresaliente

- La nota final es superior o igual a 8,5 puntos
- El conocimiento y la comprensión de la materia son muy satisfactorios.
- La comprensión conceptual es sobresaliente.
- Los problemas relacionados con la asignatura son resueltos con soltura y se es capaz de razonar alternativas de solución.
- La participación en las diferentes actividades, incluyendo las clases, ha sido muy correcta y muy satisfactoria.

Notable

- La nota final es superior o igual a 6,75 puntos
- El conocimiento y la comprensión de la materia son satisfactorios.
- La comprensión conceptual es notable.
- Los problemas relacionados con la asignatura son resueltos con soltura.
- La participación en las diferentes actividades, incluyendo las clases, ha sido correcta y bastante satisfactoria.

Aprobado

- La nota final es superior o igual a 5 puntos
- El conocimiento y la comprensión de la materia son básicos.
- La comprensión conceptual es suficiente.
- Los problemas relacionados con la asignatura son resueltos con éxito razonable.
- La participación en las diferentes actividades, incluyendo las clases, ha sido correcta y pero no siempre satisfactoria.

Suspenso

- La nota final es inferior a 5 puntos
- El conocimiento y la comprensión de la materia no son aceptables.
- La comprensión conceptual es insuficiente.

- Los problemas relacionados con la asignatura no se resuelven de forma adecuada.
- La participación en las diferentes actividades, incluyendo las clases, ha sido escasa y deficiente.

10. Análisis de coherencia de la guía docente

Este análisis permite condensar las diferentes relaciones existentes entre los objetivos y competencias a desarrollar con el plan de trabajo del alumnado y el sistema de evaluación empleado. El análisis se ha estructurado en tres tablas separadas en función del tipo de objetivo y competencia analizados en cada caso: instrumental, interpersonal y sistémico.

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS		BLOQUES DE CONTENIDOS				PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Instrumentales		Unidad I	Unidad II	Unidad III	Unidad IV		
O11 O12 O16	CIC3 CIC5		2	3	4,6	7	Enseñanza presencial (Talleres/laboratorios de prácticas) Enseñanza no presencial (Trabajo en grupos/Realización de la práctica obligatoria) Tutorías	Procedimientos Talleres/laboratorios prácticos Práctica obligatoria Criterios Grado de comprensión y aplicación de los conceptos de la Ingeniería del Software
O13 O14 O15	CIC1 CIC2		0,1,2	3	4,5,6	7,8	Enseñanza presencial (Clases magistrales/Talleres/laboratorios de prácticas/Defensa de práctica/Exposición de ejercicios) Enseñanza no presencial (Trabajo en grupos/Realización de la práctica obligatoria/Revisión bibliográfica/Consulta de recursos en Internet) Tutorías	Procedimientos Defensas de prácticas en grupo Criterios Grado de comprensión y aplicación de los conceptos de la Ingeniería del Software Evaluación del trabajo en equipo
O16 O17 O18	CIC4 CIC5 CIC6		0,1				Enseñanza presencial (Clases magistrales/Talleres/laboratorios de prácticas) Enseñanza no presencial (Trabajo en grupos/Realización de la práctica obligatoria/Revisión bibliográfica/Consulta de recursos en Internet) Tutorías	Procedimientos Tutorías Conferencias invitadas Criterios Detectar el conocimiento y visión que tienen del mercado laboral

OI9	CIC7 CIC8	1				<p>Enseñanza presencial (Clases magistrales/Talleres/laboratorios de prácticas/Defensa de práctica)</p> <p>Enseñanza no presencial (Trabajo en grupos/Realización de la práctica obligatoria/Revisión bibliográfica/Consulta de recursos en Internet)</p> <p>Tutorías</p>	<p>Procedimientos</p> <p>Examen</p> <p>Práctica obligatoria</p> <p>Criterios</p> <p>Grado de comprensión y aplicación del concepto de proceso software</p>
OI10	CIC8 CIC10	1,2	3	4,6	7,8	<p>Enseñanza presencial (Clases magistrales/Talleres/laboratorios de prácticas/Defensa de práctica/Exposición de ejercicios)</p> <p>Enseñanza no presencial (Trabajo en grupos/Realización de la práctica obligatoria/ Realización de ejercicios/Revisión bibliográfica/Consulta de recursos en Internet)</p> <p>Tutorías</p>	<p>Procedimientos</p> <p>Ejercicios</p> <p>Examen</p> <p>Talleres/laboratorios</p> <p>Práctica obligatoria</p> <p>Criterios</p> <p>Grado de comprensión y aplicación de las técnicas y herramientas de la Ingeniería del Software</p>
OI11	CIC11-CIC15	2				<p>Enseñanza presencial (Clases magistrales/Talleres/laboratorios de prácticas/Defensa de práctica/Exposición de ejercicios)</p> <p>Enseñanza no presencial (Trabajo en grupos/Realización de la práctica obligatoria/Realización de ejercicios/Revisión bibliográfica/Consulta de recursos en Internet)</p> <p>Tutorías</p>	<p>Procedimientos</p> <p>Ejercicios</p> <p>Examen</p> <p>Talleres/laboratorios</p> <p>Práctica obligatoria</p> <p>Criterios</p> <p>Grado de comprensión y aplicación del modelo objeto</p>
OI12 OI13	CIC16-CIC17		3			<p>Enseñanza presencial (Clases magistrales/Talleres/laboratorios de prácticas/Defensa de práctica/Exposición de ejercicios)</p> <p>Enseñanza no presencial (Trabajo en grupos/Realización de la práctica obligatoria/Realización de ejercicios/Revisión bibliográfica/Consulta de recursos en Internet)</p> <p>Tutorías</p>	<p>Procedimientos</p> <p>Ejercicios</p> <p>Examen</p> <p>Talleres/laboratorios</p> <p>Práctica obligatoria</p> <p>Criterios</p> <p>Grado de comprensión y aplicación del concepto de requisito</p>
OI14	CIC18-CIC23			4,6		<p>Enseñanza presencial (Clases magistrales/Talleres/laboratorios de prácticas/Defensa de práctica/Exposición de ejercicios)</p> <p>Enseñanza no presencial (Trabajo en grupos/Realización de la práctica obligatoria/Realización de ejercicios/Revisión bibliográfica/Consulta de recursos)</p>	<p>Procedimientos</p> <p>Ejercicios</p> <p>Examen</p> <p>Talleres/laboratorios</p> <p>Práctica obligatoria</p> <p>Criterios</p>

						en Internet) Tutorías	Grado de comprensión y aplicación de un método de ADOO
OI15	CIC24-CIC29				7,8	Enseñanza presencial (Clases magistrales/Talleres/laboratorios de prácticas/Exposición de ejercicios) Enseñanza no presencial (Trabajo en grupos/Realización de ejercicios/Revisión bibliográfica/Consulta de recursos en Internet) Tutorías	Procedimientos Ejercicios Examen Talleres/laboratorios Práctica obligatoria Criterios Grado de comprensión y aplicación de un método estructurado
Elementos transversales	CIM1-CIM6 CIT1-CIT4 CIL1-CIL2	0,1,2	3	4,5,6	7,8	Enseñanza presencial (Clases magistrales/Talleres/laboratorios de prácticas/Defensa de práctica/Exposición de ejercicios) Enseñanza no presencial (Trabajo en grupos/Realización de la práctica obligatoria/Realización de ejercicios/Revisión bibliográfica/Consulta de recursos en Internet) Tutorías	Procedimientos Ejercicios Examen Talleres/laboratorios Práctica obligatoria Criterios Grado de destreza en la competencias transversales

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS	BLOQUES DE CONTENIDOS				PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Interpersonales	Unidad I	Unidad II	Unidad III	Unidad IV		
OIP1	CIPTC1	2	3	4,6	7	Enseñanza presencial (Talleres/laboratorios de prácticas/Exposición de ejercicios)	Procedimientos Actividades en grupo Criterios Grado de destreza en trabajos en grupo
OIP2	CIPTC2 CIPTC3	2	3	4,6	7	Enseñanza presencial (Talleres/laboratorios de prácticas/Defensa de práctica/Exposición de ejercicios)	Procedimientos Actividades en grupo Criterios Grado de destreza en trabajos en grupo
OIP3	CIPTR1-CIPTR4	2	3	4,6	7	Enseñanza presencial (Talleres/laboratorios de prácticas/Defensa de práctica/Exposición de ejercicios)	Procedimientos Actividades en grupo Criterios Grado de destreza en trabajos en grupo

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS	BLOQUES DE CONTENIDOS				PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Sistémicas	Unidad I	Unidad II	Unidad III	Unidad IV		
OS1	CS1	0,1,2	3	4,5,6	7,8	Enseñanza presencial (Clases magistrales/Talleres/laboratorios de prácticas/Defensa de práctica/Exposición de ejercicios) Enseñanza no presencial (Trabajo en grupos/Realización de la práctica obligatoria/Realización de ejercicios/Revisión bibliográfica/Consulta de recursos en Internet) Tutorías	Procedimientos Ejercicios Examen Talleres/laboratorios Práctica obligatoria Trabajos y prácticas de otras asignaturas Proyecto fin de carrera Criterios Nivel de actuación en destrezas transferibles
OS2 OS3 OS5	CS2 CS5	0,1,2	3	4,5,6	7,8	Enseñanza presencial (Talleres/laboratorios de prácticas/Defensa de práctica/Exposición de ejercicios) Enseñanza no presencial	Procedimientos Prácticas Trabajos en grupo Ejercicios

						(Realización de ejercicios/Revisión bibliográfica/Consulta de recursos en Internet) Tutorías	Criterios Nivel de precisión y eficiencia relacionados con la resolución de problemas nuevos
OS4 OS6	CS3 CS4	1,2	3	4,6	7,8	Enseñanza presencial (Talleres/laboratorios de prácticas/Defensa de práctica/Exposición de ejercicios) Enseñanza no presencial (Realización de ejercicios/Revisión bibliográfica/Consulta de recursos en Internet) Tutorías	Procedimientos Prácticas Trabajos en grupo Ejercicios Criterios Grado de análisis y evaluación de los procedimientos relacionados con la resolución de problemas

Guía Docente de Sistemas Informáticos

Versión 2.0 – 19-7-2006

Ana Belén González Rogado

Departamento de Informática y Automática

Escuela Politécnica Superior de Zamora – Universidad de Salamanca

Avda. Cardenal Cisneros, 34 "Campus Viriato", 49022 – Zamora, España

Tfno. +34 923 29 44 00 Ext. 3635

Fax +34 980 545002

abgr@usal.es

1. Contexto

1.1. Perfil de los créditos y su adecuación al perfil profesional y académico de la titulación

Los estudiantes que inician sus estudios de Ingeniería Técnica Informática son de muy diversa procedencia. Por ello tal como indica el currículum MSIS'2000 (*Model Curriculum and Guidelines for Graduate Degree Programs in Information Systems ACM/AIS/AITP*) debe establecerse en la titulación un bloque dedicado a fundamentos, que tenga como función unificar los conocimientos de estudiantes de diversas procedencias y con niveles distintos. En particular el modelo especifica las habilidades sobre sistemas de información y sobre empresa como prerrequisito para el resto del currículum. Esta asignatura forma parte del Plan de Estudios para asegurarse que todos los estudiantes tienen unos conocimientos mínimos básicos en sistemas informáticos, para el posterior desarrollo en el resto de las asignaturas de la titulación.

La asignatura Sistemas Informáticos dentro del Plan de Estudios actual es una asignatura obligatoria que consta de 6 Créditos LRU (1,5 teóricos y 4,5 prácticos) y que se imparte en el primer cuatrimestre del primer Curso de la Titulación. Sus objetivos van a tener dos vertientes: sobre contenido y sobre metodología.

Respecto al contenido se busca que el estudiante conozca, (o en algunos casos recuerde, dependiendo de la procedencia del alumnado), los fundamentos de los sistemas informáticos desde los puntos de vista tanto del hardware como del software. Por una parte se pretende que el estudiante conozca el funcionamiento de un ordenador haciendo especial énfasis en los sistemas de numeración y sistemas de codificación de datos; y que sea capaz de identificar los elementos que componen la máquina, así como aspectos relacionados con la interconexión de las mismas mediante redes informáticas e Internet. En cuanto al manejo, se trata de aprender a utilizar diversos sistemas operativos, principalmente GNU/Linux.

En resumen, se trata de adquirir los conceptos básicos de informática que creen una base homogénea en todo el alumnado para poder cursar el resto de asignaturas de su titulación.

Respecto a la vertiente metodológica, se busca como objetivo global un cambio en el aprendizaje del estudiante, esto significará ir hacia una programación centrada en el aprendizaje activo. Esta programación, centrada en el aprendizaje, se encuentra entre los

aspectos de innovación que se han marcado en los acuerdos dirigidos a la consecución de un Espacio Europeo de Educación Superior. Se trata de que el estudiante participe activamente en el aprendizaje de la materia, llevando a cabo un trabajo día a día, y finalizando etapas claras a lo largo del curso, en diferentes ámbitos de trabajo, con una participación en equipo y un desarrollo continuo de las habilidades.

Este planteamiento pretende comenzar a desarrollar en nuestros estudiantes aptitudes tan importantes para un Ingeniero de hoy día, como las que aparecen recogidas en el Libro Blanco para el Título de Grado en Ingeniero Informático de la ANECA:

"Las nuevas competencias que las empresas exigen a los profesionales están relacionadas con el manejo de equipos tecnológicos pero, además, precisan nuevos conocimientos, competencias sociales y emocionales, capacidades estratégicas, organizativas, de planificación, etc. Es decir, se requieren profesionales multifuncionales con una buena actitud ante el cambio y con una amplia capacidad de aprendizaje."

1.2. Ubicación y relaciones en el plan de estudios

La asignatura Sistemas Informáticos es una asignatura obligatoria, que se imparte en el primer cuatrimestre del primer Curso de la Ingeniería Técnica Informática de Gestión, cuyo Plan de Estudios actual data de 2003.

En cuanto a sus relaciones dentro del Plan de Estudios la tenemos reflejada en la Figura 1.

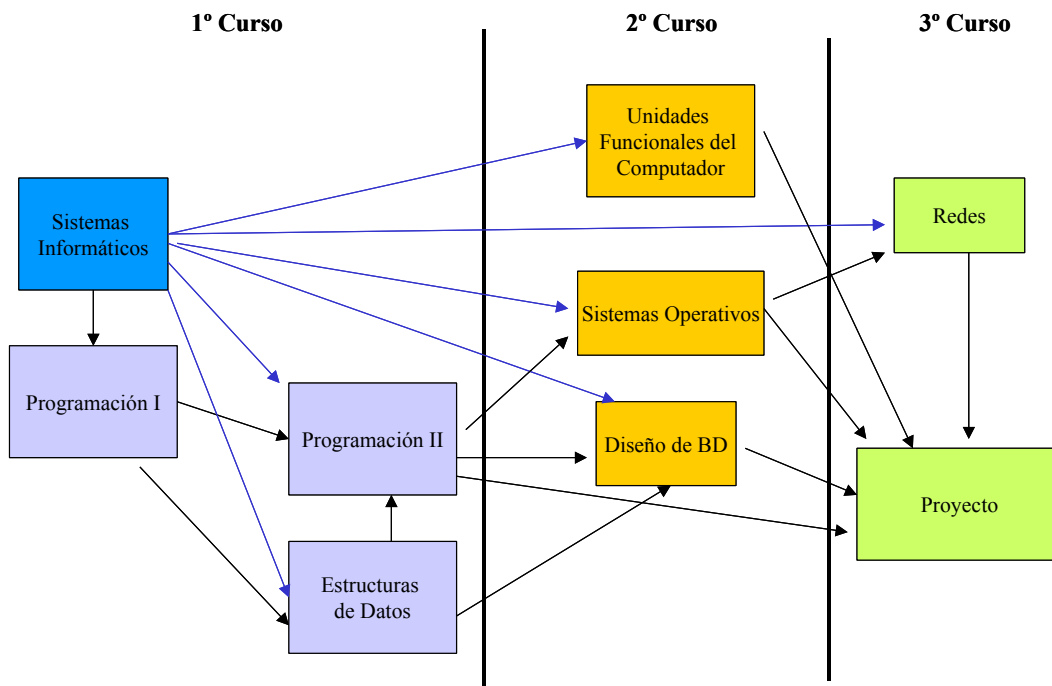


Figura 1. Relaciones de Sistemas Informáticos con otras asignaturas

2. Objetivos

2.1. Objetivos instrumentales generales

- OI1: Adquirir conocimientos generales básicos sobre sistemas informáticos.
- OI2: Conocer y utilizar los diversos sistemas de numeración utilizados en sistemas informáticos.
- OI3: Conocer y utilizar diversos métodos de codificación de la información utilizados en sistemas informáticos.
- OI4: Adquirir capacidad para emplear la lengua propia en la comprensión de los sistemas informáticos, tanto oral como escrito, siendo riguroso en las explicaciones de cualquier proceso.
- OI5: Adquirir un buen manejo de la bibliografía recomendada en la asignatura, de forma que se potencia la autosuficiencia a la hora de completar la formación.
- OI6: Adquirir capacidad de lectura comprensiva de artículos científicos técnicos.
- OI7: Tomar conciencia de las implicaciones del trabajo individual formando parte de un equipo.
- OI8: Adquirir capacidad de análisis y síntesis de información.
- OI9: Adquirir capacidad de toma de decisiones en cuanto a la selección de información.
- OI10: Conocer los problemas, principios, métodos y herramientas propios del manejo de entornos GNU/Linux.
- OI11: Adquirir fluidez en la instalación, manejo y programación del sistema operativo GNU/Linux.
- OI12: Adquirir la capacidad de crear documentos completos, correctos y legibles.

2.2. Objetivos interpersonales generales

- OIP1: Destrezas para la participación responsable: capacidad de coordinación, asistencia, contribuciones al grupo...
- OIP2: Capacidad de trabajar en equipo adquiriendo y mejorando las habilidades sociales y la inteligencia emocional.
- OIP3: Comprometerse de forma ética con el trabajo, con el resto de los integrantes del grupo y consigo mismo.
- OIP4: Capacidad de crítica y autocrítica, respecto al trabajo realizado por el resto de los compañeros y el suyo propio.

2.3. Objetivos sistémicos generales

- OS1: Capacidad de integrar los conocimientos y destrezas prácticas adquiridos para resolver situaciones reales relacionadas con los sistemas informáticos, así como con otras disciplinas relacionadas.
- OS2: Reforzar el hábito de plantearse interrogantes. Ante un problema preguntarse por el número de soluciones, la relación entre ellas, cómo afectaría a las condiciones iniciales alguna modificación...

- OS3: Capacidad de aplicar y relacionar, de forma autónoma, los contenidos de Sistemas Informáticos de forma interdisciplinar.
- OS4: Adquirir una comprensión del método científico, a través de las diversas actividades realizadas en la asignatura de Sistemas Informáticos, y asimilar su importancia como manera de pensar y actuar en la labor de un científico e ingeniero, fomentando la capacidad de abstracción y el espíritu crítico.
- OS5: Desarrollar la capacidad de aprender a aprender, para poder aplicarlo a lo largo de su vida tanto de estudiante, como profesional.
- OS6: Incentivar la preocupación por la calidad del trabajo realizado, así como la búsqueda de motivaciones para alcanzar los diversos logros.

3. Competencias

3.1. Competencias instrumentales

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

3.1.1. Habilidades cognitivas

Se distinguirán unas habilidades cognitivas generales y otras agrupadas por unidades didácticas.

Generales

- CIC1: Conocer y entender las relaciones que mantiene la asignatura con el resto de asignaturas de la titulación a partir de la introducción explícita de ejemplos y comentarios, aprovechando puntos relevantes del temario.
- CIC2: Conocer y comprender algunas de las capacidades, aptitudes y conocimientos que la asignatura aporta para el desarrollo de los diversos perfiles profesionales.
- CIC3: Ser capaz de crear documentos completos, correctos y legibles.

TEORÍA

Unidad Didáctica I: Introducción

- CIC4: Conocer los conceptos de información e informática.
- CIC5: Diferenciar los componentes básicos de la máquina.
- CIC6: Comprender la evolución de la tecnología informática.
- CIC7: Conocer los distintos tipos de computadores y sus aplicaciones.
- CIC8: Entender el método científico y su aplicación en informática.
- CIC9: Entender la noción de metodología.

Unidad Didáctica II: La máquina y los datos

- CIC10: Conocer cada uno de los componentes del soporte físico.
- CIC11: Diferenciar memoria de sistemas de almacenamiento.
- CIC10: Conocer el fundamento y los diferentes niveles en el soporte lógico.
- CIC12: Comprender la importancia del Sistema Operativo.

- CIC13: Comprender el flujo de información entre los distintos componentes del ordenador.
- CIC14: Comprender la necesidad de las redes de computadores.
- CIC15: Conocer los fundamentos y principios de la red de redes, Internet.
- CIC16: Comprender la necesidad de utilización de sistemas de numeración diferentes al sistema decimal en el ordenador.
- CIC17: Ser capaz de trabajar con los sistemas de numeración utilizados con el computador.
- CIC18: Comprender la necesidad de codificar.
- CIC19: Comprender los distintos métodos de codificación.
- CIC20: Comprender la necesidad de proteger la información.
- CIC21: Entender la codificación en función del soporte físico.

Unidad Didáctica III: Aprendiendo a aprender

- CIC21: Profundizar en alguno de los campos presentados genéricamente en las unidades anteriores.

PRACTICA

- CIC22: Comprender la filosofía GNU
- CIC23: Comprender el proyecto GNU/Linux
- CIC24: Comprender la arquitectura del software de Linux
- CIC25: Conocer que es una distribución
- CIC26: Conocer el manejo del Sistema Operativo GNU/Linux

3.1.2. Capacidades metodológicas

- CIM1: Ser capaz de tomar decisiones de manera razonada.
- CIM2: Tener capacidad de análisis y síntesis.
- CIM3: Ser capaz de manejar bibliografía relacionada con Sistemas Informáticos.
- CIM4: Ser capaz de realizar un trabajo continuo.
- CIM5: Ser capaz de manejar revistas electrónicas, tanto de divulgación como científicas.

3.1.3. Destrezas tecnológicas

- CIT1: Habilidades básicas de navegación por la Web y uso del resto de servicios de red para la obtención y manejo de la información relacionada con la asignatura.
- CIT2: Manejar las operaciones básicas de diferentes sistemas operativos para la instalación de las distintas herramientas.
- CIT3: Manejar con fluidez las operaciones de administración del Sistema Operativo GNU/Linux.

- CIT4: Manejo avanzado de un sistema de procesamiento de textos para la realización de los diferentes trabajos de teoría, informes de las prácticas, documentación de la práctica obligatoria.
- CIT5: Manejo avanzado de un sistema de tratamiento gráfico para la realización del trabajo Tipo 4 (póster).
- CIT6: Manejo del correo electrónico para la entrega de trabajos de envío electrónico obligatorio.
- CIT7: Manejo avanzado de programas de presentaciones gráficas para las exposiciones orales de los trabajos Tipo 3 y 4.

3.1.4. Destrezas lingüísticas

- CIL1: Adquirir y utilizar con fluidez un buen lenguaje científico, tanto oral como escrito, siendo riguroso en las explicaciones de cualquier proceso dentro de la asignatura.
- CIL2: Conocer y utilizar la terminología usual de los sistemas informáticos, tanto en español como en inglés.

3.2. Competencias interpersonales

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

3.2.1. Competencias para tareas colaborativas

- CIPTC1: Ser capaz de trabajar en equipo para resolver los diferentes trabajos planteados en grupo en la asignatura.
- CIPTC2: Ser capaz de presentar en público los resultados de los trabajos realizados y mantener un debate con el resto de la clase sobre el tema trabajado, aclarando las posibles dudas de forma colaborativa.
- CIPTC3: Ser capaz de realizar la práctica obligatoria de la asignatura.
- CIPTC4: Ser capaz de calificar el trabajo realizado por otros compañeros.

3.2.2. Compromiso con el trabajo

- CIPTR1: Se ha de definir un plan de trabajo en el que el volumen de trabajo de todos los miembros del equipo sea similar.
- CIPTR2: Una vez finalizado el trabajo, todos los miembros del grupo deben conocer en profundidad todo el desarrollo realizado.
- CIPTR3: Se debe cumplir el plazo de entrega de los trabajos.
- CIPTR4: Se debe adquirir un compromiso ético entre todos los componentes del grupo.

3.3. Competencias sistémicas

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

- CS1: Capacidad de aplicar los conocimientos, métodos y herramientas vistos en la asignatura Sistemas Informáticos a situaciones y problemas concretos del área de la Ingeniería Informática y de otras disciplinas relacionadas.

- CS2: Capacidad de aprender y aplicar, de forma autónoma e interdisciplinar, nuevos conceptos y métodos relacionados con cada asignatura.
- CS3: Motivación por la calidad y por la creatividad.
- CS4: Capacidad de adoptar el proceso marcado por el método científico y de ingeniería en el planteamiento y realización de trabajos diversos, tanto a nivel académico como profesional.
- CS5: Capacidad para criticar, tanto las tareas realizadas por el profesor como las de los compañeros.
- CS6: Capacidad de autocrítica.

4. Prerrequisitos

No hay.

5. Temario

5.1. Unidades didácticas

Tema 0: Sumario de la asignatura

Tema 1: Conceptos básicos.

Unidad Didáctica II. La máquina y los datos

Tema 2: Visión global de la máquina.

Tema 3: Sistemas de numeración.

Tema 4: Codificación de la información.

Unidad Didáctica III. Aprendiendo a aprender

Tema 5. Soporte físico

Tema 6. Soporte lógico

Tema 7. Teleinformática

Team 8. Historia de la informática

Prácticas (0 T + 3 P):

Unidad Didáctica 1: Introducción

Unidad Didáctica 2: Operaciones básicas

Unidad Didáctica 3: Órdenes

Unidad Didáctica 4: Guiones (Shell scripts)

Unidad Didáctica 5: Configuración

Unidad Didáctica 6: Instalación del Sistema Operativo

5.2. Desarrollo de las unidades didácticas

Tema 0. Sumario de la Asignatura

0.1. Motivación

0.2. Asignatura Sistemas Informáticos en ITIG

0.2.1. Objetivos

0.2.2. Temario

0.2.3. Evaluación

0.2.4. Bibliografía básica recomendada

0.2.5. Otras fuentes

0.2.6. Enlaces

0.2.7. Tutorías

Tema 1. Conceptos básicos.

1.1. Tratamiento de la Información. Informática

1.2. Personajes de la historia de la Informática

1.3. Definiciones básicas. Conceptos

1.4. Tipos de los computadores

1.5. El método científico

Unidad Didáctica II. La máquina y los datos

Tema 2. Visión global de la máquina

2.1. Diagrama de bloques de un computador

2.2. Soporte Físico:

2.2.1. El procesador

2.2.2. La memoria

2.2.3. Periféricos

2.2.4. Sistemas de Almacenamiento

2.3. Soporte Lógico:

2.3.1. Sistema Operativo

2.3.2. Programando la máquina

2.3.3. Aplicaciones

2.4. Flujo de datos en el interior

2.5. Redes

Tema 3. Sistemas de Numeración.

3.1. Sistema binario, octal y hexadecimal. Bits y bytes.

3.2. Cambio de base

3.3. Operaciones aritméticas sencillas en binario

3.4. Ejercicios

Tema 4. Codificación de la Información.

4.1. ¿Qué es un código?. Sistemas de codificación

4.2. Codificación de números: Enteros y Reales

4.3. Códigos alfanuméricos

4.4. Codificación de datos en el ordenador: sonidos, imágenes, vídeos

4.5. Compresión de la información

4.6. Protección de la información

4.7. Codificación física de los datos

4.8. Ejercicios

Unidad Didáctica III:

Tema 5. Soporte Físico

5.1. El procesador. Estructura física y lógica

5.2. *La memoria. Características, tipos y clasificación*

5.3. Periféricos. Entrada, salida y E/S

5.4. Soportes de Almacenamiento. Magnéticos, ópticos

Tema 6. Soporte lógico

6.1. Sistema Operativo. Módulos. Tipos

6.2. *Lenguajes de programación. Tipos. Filosofías*

6.3. *Ingeniería del Software*

Tema 7. Teleinformática

7.1. Redes de transmisión

7.2. Internet

Team 8. Historia de la Informática

- 8.1. Evolución histórica de la informática
- 8.1. Evolución de los lenguajes de programación

PRÁCTICA:

1.2. Introducción al editor VI

- 2.1. Operaciones básicas con ficheros y directorios
- 2.2. Estructura de directorios y atributos de los ficheros.

- 3.1. La línea de órdenes: órdenes, opciones y metacaracteres
- 3.2. Órdenes para el manejo de ficheros y directorios
- 3.3. Control y monitorización de procesos

- 4.1. Modos de ejecución. Procesos padre e hijo
- 4.2. Variables locales, globales, de entorno y argumentos posicionales
- 4.3. Sentencias condicionales y ejecución selectiva
- 4.4. Bucles for, while y until. Break y continue
- 4.5. Expresiones aritméticas
- 4.6. Funciones

Unidad Didáctica 5: Configuración

- 5.1. Ficheros de configuración del entorno y del sistema

Unidad Didáctica 6: Instalación del Sistema Operativo

6. Metodología y estrategias de aprendizaje

6.1. Metodología docente

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

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

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

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

Para caminar en este sentido, el modelo educativo que se va a seguir en la asignatura de Sistemas Informáticos tiene la clase magistral como un elemento más, pero no exclusivo, en la transmisión de conocimiento. Este tipo de enseñanza se va a unificar con otros procesos como van a ser la realización de diferentes tipos de trabajos tanto en la parte de teoría de la asignatura como en la parte práctica, que en su conjunto buscan una enseñanza colaborativa, un proceso de trabajo continuo y una interacción profesor-estudiante más próxima. Concretamente las actividades que se proponen son las siguientes:

- *Clases de teoría con apoyo de material audiovisual.* En estas clases se presentarán los contenidos básicos de los temas incluidos en las Unidades Didácticas I y II. Las clases llevarán control de asistencia y comenzarán con un resumen de los contenidos que se pretenden transmitir en la clase, así como con un breve comentario a los conceptos vistos en clases anteriores y que sirven de enlace a los que se pretenden desarrollar. El desarrollo de la clase se llevará a cabo con medios audiovisuales, textos, transparencias, conexiones a la red, componentes físicos (hardware) como apoyo a las explicaciones, ... que permitan un adecuado nivel de motivación e interés en los estudiantes. Se debe intentar motivar a los estudiantes a intervenir en cualquier momento en las clases para hacer éstas más dinámicas y facilitar el

aprendizaje. Es importante intentar terminar la exposición con las conclusiones más relevantes del tema tratado.

Las transparencias que se utilizarán en clase son un subconjunto de las que se facilitan a los estudiantes en la página web y en la fotocopidora. Estas transparencias son una guía para el estudio, pero no son sustitutas de la bibliografía recomendada. Todos los temas comparten una misma estructura que se compone de los siguientes ítems:

- *Portada*: Con el título del tema y fecha de última modificación.
- *Esquema*: Con el índice del tema y bibliografía recomendada que deben consultar para ampliar / preparar el tema.
- *Desarrollo del tema*: Con los apartados en los que se divide el tema.
- *Aportaciones principales del tema*: A forma de resumen del tema, incluye las aportaciones y conclusiones más importantes del tema.
- *Ejercicios*: Conjunto de enunciados con cuestiones y ejercicios sobre el tema, se incluyen también ejercicios resueltos.
- *Lecturas complementarias*: Lecturas opcionales para profundizar en el tema presentado.
- *Referencias*: Lista de todas las referencias que se citan en el desarrollo del tema.
- *Trabajos en grupo*: Entre los métodos de aprendizaje empleados, la elaboración de trabajos se considera un elemento interesante para fomentar el "saber hacer junto con otros".

Los trabajos en equipo constituyen una herramienta básica para realizar un cambio respecto a la mentalidad de los estudiantes acerca del proceso de aprendizaje. La tendencia clásica en la Universidad está actualmente basada en un tipo de estudiante "pasivo". Consideramos fundamental que la actitud a fomentar desde el profesorado debe ser la del estudiante "activo", que decide su propio proceso de aprendizaje.

En esta asignatura se realizarán trabajos en equipo, que denominaremos trabajos de complemento. Los trabajos de complemento, con una guía de realización claramente definida, y una planificación rigurosa en el tiempo, le enseñarán a desenvolverse en el mundo empresarial, proporcionándole habilidades no sólo técnicas, sino también sociales, como la organización, dirección, comunicación, coordinación, tolerancia, expresión oral, sentido de la responsabilidad, capacidad para el debate, etc.

La elaboración de trabajos dirigirá al estudiante hacia la lectura y comentario de artículos y bibliografía relacionada, acerca de un apartado concreto de la materia, motivando su interés por la asignatura. En otros casos se puede plantear la elaboración de un informe sobre un tema concreto que implique la búsqueda de bibliografía. De esta manera se despierta el interés por la

investigación, a la vez que permite un conocimiento más profundo de la materia, o de aspectos avanzados de la misma.

Una vez realizado el trabajo, los componentes del grupo deberán exponerlo en clase, durante un tiempo prefijado. Transcurrida la exposición, se iniciará un debate en clase entre todos los estudiantes sobre distintos aspectos relacionados con el trabajo, bajo la supervisión del profesor.

Los trabajos se realizarán en grupos de 4 componentes. Cada grupo deberá realizar 4 trabajos, con la planificación que el profesor establezca.

Los tipos de los trabajos serán:

- **Tipo 1:** Ejercicios de sistemas de numeración y codificación de la información. Formato electrónico.
- **Tipo 2:** Trabajo de documentación. Manejo de revistas electrónicas de investigación para localización de artículos sobre un tema propuesto. Formato electrónico.
- **Tipo 3:** Trabajo de investigación. Elaboración de un trabajo, con reglas preestablecidas de formato, distribución y contenido sobre algún punto de la Unidad Didáctica III. Utilizando tanto bibliografía en papel como electrónica. Formato electrónico y papel.
- **Tipo 4:** Trabajo de síntesis. Elaboración de un póster sobre algún punto de la Unidad Didáctica III. Formato electrónico y papel (Dimensiones A1)

Una vez entregado cada trabajo, y en el horario establecido al respecto, se procederá a la defensa del mismo, como se explicará posteriormente en los métodos de evaluación.

Los contenidos concretos de los trabajos se darán a conocer en las primeras semanas del curso, pero en todo caso versarán sobre los contenidos del programa de la asignatura.

- *Corrección de trabajos.* Finalizada la exposición de cada trabajo, cada estudiante deberá enviar por correo electrónico una hoja de calificaciones, puntuando distintos aspectos del trabajo expuesto. La valoración del profesor de la corrección realizada por el estudiante, contribuirá a la nota final.
- *Talleres de prácticas.* Las clases prácticas presenciales estarán dedicadas a la comprensión, manejo e instalación del sistema operativo GNU/Linux. El taller se organiza de la siguiente forma:
 - Sesiones de trabajo de dos horas.
 - La primera parte de cada sesión de prácticas estará dedicada a una exposición por parte del profesor sobre uno o varios de los temas

programados. Esta exposición se hará preferentemente mediante ejercicios demostrativos.

- El resto de la sesión lo dedicará el estudiante a la solución de un problema planteado por el profesor con el fin de evaluar su capacidad y aprendizaje sobre el tema expuesto.
 - Finalizadas todas las sesiones de prácticas, cada estudiante deberá enviar por correo electrónico, informes asociados a cada una de las sesiones prácticas realizadas.
- *Práctica obligatoria.* Hay una práctica final obligatoria en la que se plantea la realización de guiones que engloban conceptos vistos en las distintas sesiones de prácticas. El estudiante entregará de cada problema planteado la solución en forma de guión (shell script) y un informe escrito que describa y comente su propuesta.
 - *Tutorías presenciales.* El alumnado tiene a su disposición seis horas de tutorías a la semana en las que puede consultar cualquier duda relacionada con los contenidos, organización y planificación de la asignatura. Las tutorías pueden ser individuales o grupales para resolver problemas relacionados con las actividades a realizar en grupo.
 - *Tutorías obligatorias.* Previamente a la elaboración de los trabajos tipo 3 y 4 se realizará, al menos, una sesión de tutoría obligatoria para la presentación al profesor de la bibliografía seleccionada para la elaboración de ambos trabajos.
 - *Tutorías Telemáticas.* Se podrá utilizar el correo electrónico como medio de comunicación, para resolución de dudas y comunicación entre profesor y estudiantes, el profesor responderá dentro de sus horas de tutorías.
 - *Página web.* Se convierte en el vehículo de comunicación y registro de información de la asignatura.

6.2. Estrategias de aprendizaje

Se detalla las actuaciones concretas a realizar para la aplicación y aprovechamiento de la metodología propuesta. Se estructura en las siguientes fases:

6.2.1. Recopilación de la documentación de la asignatura

- Los estudiantes obligatoriamente deberán tener una cuenta de correo electrónico, para su comunicación con el profesor de la asignatura.
- Los estudiantes dispondrán en la página web de la asignatura de toda la información y normativa relacionada con la asignatura: temario, criterios de evaluación, bibliografía, apuntes, enunciados de prácticas, trabajos propuestos, reglas para la elaboración de los trabajos, enlaces de interés, lecturas complementarias, avisos, distribución de grupos, fechas de entrega y defensa de los diferentes trabajos ... La página también incluirá los trabajos finalizados según se vayan entregando.
- El profesor mantendrá actualizada la información de esta página para que se convierta en un vehículo de comunicación con los estudiantes.

- Los estudiantes deben conocer y manejar con fluidez toda la información y la documentación que se incluye en la página de la asignatura.
- Asiduamente, el estudiante debe consultar la página para estar actualizado.
- El profesor responderá a los correos electrónicos dentro de sus horas de tutorías.

6.2.2. Planificación de las clases teóricas. Unidades Didácticas I y II

- Las clases teóricas se dedicarán a la presentación de contenidos y a la discusión sobre las dudas que surjan durante las exposiciones.
- Las clases se desarrollarán según el siguiente esquema:
 1. El profesor presenta los objetivos a conseguir y el contexto en el que se va a desarrollar la clase.
 2. Se presentan los conceptos.
 3. Se aclaran las dudas cuando éstas surjan. El profesor puede incentivar el debate con preguntas para hacer la clase más participativa.
 4. El profesor termina con un resumen de los principales conceptos tratados en la clase y puede introducir la siguiente clase.
- El estudiante debe haber realizado una lectura previa de los contenidos que se van a tratar en la clase, máxime cuando no todos los contenidos se desarrollan en las clases magistrales.
- Una vez terminada la clase magistral, se debe estudiar de forma autónoma su contenido y en caso de no entender algo intentar aclararlo utilizando la bibliografía recomendada o cualquiera de los materiales adicionales recomendados en la asignatura. En caso de necesitarlo, resolver las dudas asistiendo a tutorías
- Las clases magistrales se desarrollarán durante las 6 primeras semanas del cuatrimestre.

6.2.3. Planificación de las clases teóricas. Unidad Didáctica III

- Durante 5 semanas se suspenderán las clases teóricas, para que los estudiantes realicen los distintos trabajos encomendados a cada grupo.
- Dentro de ese periodo se fijará las fechas de entrega de los trabajos tipo 1 y 2, así como su correspondiente defensa.
- En las 4 últimas semanas del cuatrimestre se procederá, en las clases teóricas, a las exposiciones que los distintos grupos harán ante el resto de la clase de los trabajos tipo 3 y 4 realizados.
- Los estudiantes deben leer previamente, los trabajos que se expongan cada día, para poder emitir la hoja de calificación correspondiente.
- Entregados los trabajos tipo 4 se procederá a realizar una exposición por los pasillos de la Escuela.

6.2.4. Planificación de las clases prácticas

- El profesor publicará en la página de la asignatura los enunciados, con indicación de plazos de entrega, con suficiente antelación.

- En el caso de los talleres, los estudiantes deberán leer y comprender bien el enunciado para realizar su propuesta de solución.
- En el caso de la práctica obligatoria los estudiantes deben leer y entender bien todos los puntos antes de ponerse a llevarla a cabo.
- Durante los talleres el profesor resolverá las dudas que origine el problema dando argumentos a cada una de las decisiones o alternativas que se planteen.
- Los estudiantes deben cumplir el calendario de entrega de prácticas.
- Los informes elaborados sobre los talleres y la práctica obligatoria deberán enviarse por correo electrónico al profesor.

6.2.4. Evaluación

- Se plantea una forma de evaluación continua: control de asistencia a clase, participación en clase y defensa de cada uno de los trabajos.
 - Defensa Trabajo Tipo 1: Prueba escrita individual. Realización de ejercicios.
 - Defensa Trabajo Tipo 2: Entrevista en grupo con el profesor en su despacho.
 - Defensa Trabajo Tipo 3: Exposición en clase ante el resto de los compañeros.
 - Defensa Trabajo 4: Explicación en clase ante el resto de los compañeros y exposición de los póster en los pasillos del Centro.
- Para aquellos estudiantes que no superen la evaluación continua, o no quieran, o no puedan acogerse a ella, existirá una prueba final.
- Cada estudiante evaluará el trabajo expuesto por sus compañeros, tanto de forma en su forma escrita como en la exposición oral, mediante la hoja de calificación correspondiente. La valoración del profesor de esta corrección formará parte de la nota final.

7. Planificación del tiempo y del esfuerzo

La asignatura de Sistemas Informáticos tiene en el plan de estudios actual 6 créditos LRU asignados, 1,5 créditos de teoría y 4,5 créditos prácticos. Los 4,5 créditos prácticos se han dividido en 3 créditos para los talleres de prácticas y 1,5 créditos asociados a las clases de teoría. Para hacer su estudio en ECTS se ha tomado tomando la equivalencia de crédito LRU a crédito ECTS, con implicación de cambio de metodología docente.

Al hacer el estudio de carga de trabajo máxima, se tiene que:

- 6 ECTS → 150-180 horas de trabajo
- 1,5cr + 1,5 cr LRU → 30 horas presenciales + (30*1,5) horas de asimilación = 75 horas
- 3cr LRU → 30 horas presenciales + (30*2) horas de asimilación = 90 horas
- 75 + 90 = 165 horas de trabajo

- 150 < 165 < 180

Dos son los escenarios más probables: el resultante de tomar el crédito ECTS como 30 horas de trabajo y el resultante de tomarlo como 25 horas de trabajo.

Por temas de organización y calendario, se ha optado por partir del escenario que asigna 25 horas de trabajo al ECTS.

Para poder elaborar las guías, cuyo destino final sería la realización de experiencias piloto, dentro de los planes de estudio vigentes, se debe tener en cuenta la definición del crédito del R.D. 1497/1987 y sus posteriores modificaciones. Así:

- La unidad de valoración de las enseñanzas se corresponde a diez horas de enseñanza teórica, práctica o de sus equivalencias entre las que podrán incluirse actividades académicas dirigidas, que habrán de preverse en el correspondiente plan docente junto con los mecanismos y medios objetivos de comprobación de los resultados académicos de las mismas. Todo ello sin perjuicio del cumplimiento del régimen de dedicación del profesorado, de conformidad con el Real Decreto 898/1985, de 30 de abril sobre régimen del profesorado universitario.
- En ningún caso, salvo que se trate de enseñanzas en Universidades a distancia, el porcentaje del crédito correspondiente a las actividades académicas dirigidas será superior al **30%**.

De esta forma, si 6 créditos LRU equivalen a 60 horas, se debe asegurar el 70% como mínimo de presencialidad, esto es, 42 horas entre horas de teoría y de práctica.

En la Tabla 1 se presenta la organización del esfuerzo del estudiante para cubrir los 6 ECTS asociados a la asignatura. Por su parte en la Tabla 2 se distribuyen las horas presenciales entre los diferentes temas y el examen.

			A	B	C	D	E
	Técnica	Actividad	Horas equivalentes de clase	Factor de trabajo del estudiante [†]	Horas de trabajo personal del estudiante	Horas totales (A+C)	ECTS (D ÷ 25)
Teoría	Seguimiento de la parte teórica de las unidades didácticas	Asimila contenidos. Se plantea dudas que planteará a los profesores en las tutorías	12	1,5	18	30	1,2
Práctica	Seguimiento de la parte práctica de las unidades didácticas	Experimenta, practica, configura, instala	24	2	48	72	2,88
Trabajos de complemento	Ejercicios de afianzamiento de lo aprendido. Trabajos de búsqueda, investigación y síntesis. Computarán para la	Realiza, resuelve problemas, investiga, sintetiza, busca información, lectura	8	2,5	20	28	1,12

[†] Número de horas dedicadas por el estudiante al trabajo personal (organización de apuntes, estudio, documentación...) por cada hora de clase.

	superación de la asignatura.	compresiva...					
Práctica obligatoria	Práctica individual de obligada realización	Realiza una serie de guiones agrupando conceptos	-	-	10	10	0,4
Examen	Controles de evaluación continua y/o examen final de la asignatura	Exámenes de evaluación de conceptos, realización de guiones ...	3			3	0,12
Otras actividades	Tutorías personalizadas y grupales	Recibe orientación personalizada	-	-	3	3	0,12
	Búsquedas en la red, participación en foros especializados...	Busca elementos para completar los contenidos	-	-	4	4	0,16
TOTAL			47		103	150	6

Tabla 10. Escenario ECTS = 25 horas de trabajo

NÚMERO DE HORAS PRESENCIALES		
Actividad	Horas Teoría	Horas Prácticas
TEORÍA		
Unidad Didáctica I: Introducción		
Tema 0	1	
Tema 1	1	
Unidad Didáctica II: La máquina y los datos		
Tema 2	3	
Tema 3	2	
Tema 4	5	
Unidad Didáctica III: Aprendiendo a aprender		
Tema 5 – 6- 7- 8	8	
PRACTICA		
Unidad Didáctica 1: Introducción		2
Unidad Didáctica 2: Operaciones Básicas		2
Unidad Didáctica 3: Órdenes		4
Unidad Didáctica 4: Shell scripts		12
Unidad Didáctica 5: Configuración		2
Unidad Didáctica 6: Instalación del Sistema Operativo		2
Examen	3	
TOTAL: 47	23	24

Tabla 11. Reparto de horas presenciales

8. Bibliografía

8.1. Bibliografía básica

Alcalde, Eduardo, García Miguel (1997), "Informática Básica", Ed. Mc Graw Hill.

Beekmann, George - (2005) "Introducción a la Informática" - 6ª Edición, Ed. Pearson Prentice Hall. 664 Páginas

Martín Martín-Pozuelo, José Mª (2005), "Hardware Microinformático: Viaje a las Profundidades del PC", Ed. Ra-Ma. 632 Páginas.

Miguel Anasagasti, Pedro de (2004), "Fundamentos de los Computadores", Ed. Paraninfo, 672 Páginas.

Pareja, C.-Andeyro, A. - Ojeda Aciego, M. (1994), "Introducción a la Informática", Ed. Complutense. (Disponible en pdf: <http://dalila.sip.ucm.es/~cpareja/intro-inf/>).

Prieto-Lloris-Torres (2001), "Introducción a la Informática", Ed. McGraw-Hill.

Prieto Espinosa, A. y Prieto Campos, B. (2005), "Conceptos de Informática" Serie Schaum, Ed. McGraw-Hill, 533 Páginas.

Sánchez Vidales, M.A. (2001), "Introducción a la Informática: Hardware, Software y Teleinformática", Publicaciones Universidad Pontificia de Salamanca.

Afzal, Amir - "Introducción a UNIX - Un enfoque práctico", Ed. Prentice-Hall, 1999

Carretero, Jesús et al "Sistemas Operativos-Una visión aplicada" Ed. McGrawHill, 2001.

Mansoor Sarwar, Syed - Koretsky, Robert - Ageel Sarwar, Syed: "El Libro de Linux", Ed. Pearson Addison Wesley, 2003

Ray, John, "Guía esencial LINUX" Ed. Prentice Hall, 2002, 284 pág.

Rollón Barrios, Daniel J. - Casado Estrada, J.A.: "Linux: Guía Rápida", Ed. Paraninfo

8.2. Bibliografía complementaria

Bartee, T. C.: "Fundamentos de computadores digitales", Ed. McGraw-Hill, 1990.

García Peñalvo, F.J., et al "Programación en C", Dpto. Informática y Automática. Universidad de Salamanca, 2003.

Hamacher, V. C., Vranesic, Z. G., Zaky, S. C.: "Organización de computadoras", Ed. McGraw-Hill, 1987.

Lipschutz, Seymour: "Estructuras de datos", Ed. McGraw-Hill, 1987.

- Mansoor Sarwar, Syed - Koretsky, Robert - Ageel Sarwar, Syed. "El Libro de Unix", Ed. Prentice Hall, 2002
- Stallings, W. "Comunicaciones y Redes de computadores" (7ª Edición) Ed. Pearson Prentice Hall, 2004.
- Stallings, W., "Sistemas Operativos" (5ª edición), Prentice Hall, 2005.
- Tanenbaum, A.S. - "Redes de Ordenadores" (4ª Edición), Ed. Pearson Prentice Hall, 2003.

8.3. Otros recursos

Pohl, K. (1997) Requirements Engineering: An Overview. En M. Dekker (Ed.), *Encyclopedia of Computer Science and Technology*, 36. Disponible en <ftp://sunsite.informatik.rwth-aachen.de/pub/CREWS/CREWS-96-02.pdf>. [Última vez visitado, 22-9-2005].

8.4. Enlaces de interés

- **Association for Computing Machinery (ACM)**
 - <http://www.acm.org>.
 - Fundada en 1947 fue la primera sociedad científica y de educación del mundo. El portal de información que presenta es impresionante, tanto en cuanto a enlaces de interés, grupos de trabajo, documentos electrónicos, conferencias como por su biblioteca digital conteniendo revistas y actas de congresos (<http://portal.acm.org>).
- **Consejo Superior de Informática**
 - <http://www.csi.map.es/>.
 - Web del Consejo Superior de Informática donde, entre otras cosas, se puede obtener la documentación y herramientas sobre Métrica 3 (<http://www.csi.map.es/csi/metrica3/index.html>).
- **European Software Institute (ESI)**
 - <http://www.esi.es>.
 - El Instituto del Software Europeo tiene su sede en Bilbao (España). Dispone tanto de documentación privada para los miembros del instituto como documentación pública con los análisis de proyectos, necesidades de empresas y software europeo.
- **Institute of Electrical and Electronics Engineers (IEEE)**
 - <http://www.ieee.org>.
 - Otra prestigiosa organización compuesta por diversas sociedades, donde la que más relación tiene con los temas abordados en el presente curso es la IEEE Computer Society (<http://computer.org>).
 - A semejanza de ACM, ofrece información sobre conferencias, estándares, educación y mantiene otra biblioteca digital con revistas y actas de congresos.

- **The World Wide Web Consortium**
 - <http://www.w3.org/>.
 - Sitio central donde se recogen las especificaciones de los diversos lenguajes relacionados con la Web (HTML, XML, RDF, SOAP...).
- **Página del profesor Rafael Barzanallana - Universidad de Murcia**
 - <http://www.um.es/docencia/barzana>
 - Sitio con unos buenos apuntes y enlaces sobre la asignatura.

8.5. Enlaces Linux

- Fundamentales:
 - <http://www.linux.org>, <http://www.linuxhq.com>, <http://www.kernel.org>, <http://www.li.org>, <http://www.tux.org>
- Sitios dedicados a docs (inglés y español):
 - <http://www.linuxdoc.org>, <http://www.linuxnewbie.org>, <http://sunsite.dk/linux-newbie>, <http://www.firstlinux.com>, <http://www.xmission.com/~howardm>, <http://www.debian.org>, <http://www.debianhelp.org>, <http://www.hispalinux.es>, <http://www.linux-es.com>, <http://www.laespinal.org>, <http://www.linuxparatodos.com>
- Sitios noticias, revistas, ayudas en línea:
 - <http://www.barrapunto.com>, <http://www.linuxjournal.com>, <http://www.linuxgazette.com>, <http://www.linux-mag.com>
- Sitios almacén de proyectos, paquetes software:
 - <http://sourceforge.net>, <http://savannah.gnu.org>, <http://rpmfind.net>, <http://packages.debian.org>
- Distribuciones Linux:
 - **Debian GNU/Linux:** Es un proyecto totalmente no-comercial; posiblemente el más puro con los ideales que iniciaron el movimiento del software libre.
 - A favor: 100% libre, bien probada, bastante actualizada, instalación de software sencillísima usando apt-get.
 - En contra: No tiene tantos paquetes como otras distribuciones.
 - Sistema de paquetes: DEB
 - Descarga gratuita: Si
 - <http://www.debian.org>
 - **Gentoo.** Gentoo Linux es una distribución basada en código fuente, la única en esta relación.
 - A favor: Fácil instalación de paquetes de software individuales, altamente actualizada, el sentimiento de crear tu propia distribución atendiendo a las necesidades del usuario.
 - En contra: Instalación larga y tediosa, ocasionalmente inestable y con riesgos de romperse, no aconsejada para servidores con funciones críticas.
 - Sistema de paquetes: SRC
 - Descarga gratuita: Si
 - <http://www.gentoo.org>

- **Knoppix.** Su arranque automático, gran cantidad de software, su sistema de descompresión al vuelo y la posibilidad de instalarla en el disco duro han convertido a knoppix en una herramienta indispensable.
 - A favor: Excelente autodetección de hardware, funciona directamente del CD sin instalación en el disco duro, puede ser usada como herramienta de recuperación.
 - En contra: Bajo rendimiento y velocidad si se utiliza directamente del CD.
 - Sistema de paquetes: DEB
 - Descarga gratuita: Si
 - <http://www.knoppix.org>
- **Linspire** (antiguo Lindows.com). Querían desarrollar un SO basado en Linux capaz de utilizar aplicaciones Linux, y herramientas Windows.
 - A favor: Diseñado para principiantes, instalación rápida y fácil, instalación de software con un click.
 - En contra: Cara, cota anual para acceder al directorio de software, prácticas comerciales preocupantes.
 - Sistema de paquetes: DEB
 - Descarga gratuita: No
 - <http://www.linspire.com>
- **Lycoris.** El objetivo principal era crear una instalación fácil de usar para realizar la transición entre Windows y Linux lo más cómodamente posible.
 - A favor: Amigable para principiantes, diseñada para parecerse a Windows.
 - En contra: Contiene paquetes obsoletos, requiere una licencia para uso comercial.
 - Sistema de paquetes: RPM
 - Descarga gratuita: Si
 - <http://www.lycoris.org>
- **Mandriva:** Antigua Mandrake Linux. Distribución popular entre los nuevos usuarios de Linux y aquellos hogares que buscan un sistema operativo alternativo.
 - A favor: Amigable para el usuario, herramientas de configuración gráfica, enorme soporte de la comunidad, posibilidad de cambiar el tamaño de particiones NTFS y muy actualizada.
 - En contra: Actualmente está integrándose con conectiva, por lo que es posible que haya cambios estructurales en la distribución, para adaptarse a todos los mercados. Puede convertirse en una gran distribución si este paso lo lleva a cabo correctamente.
 - Sistema de paquetes: RPM
 - Descarga Gratuita: Si
 - <http://www.mandriva.com>
- **Red Hat – Fedora:** Red Hat Linux se ha convertido en la distribución Linux dominante en servidores en todo el mundo. Siendo el proyecto Fedora para usuarios finales.
 - A favor: Muy usada, excelente soporte de la comunidad y muchas innovaciones.
 - En contra: Menor soporte para la versión gratuita (fedora), soporte multimedia pobre.

- Sistema de paquetes: RPM
- Descarga Gratuita: Si
- <http://www.redhat.com>
- <http://fedora.redhat.com>
- **Slackware.** Solo se recomienda a aquellos usuarios nuevos que deseen invertir tiempo aprendiendo acerca de Linux.
 - A favor: Alta estabilidad y ausencia de fallos, sigue fielmente los principios de UNIX.
 - En contra: Toda la configuración se realiza mediante la edición de ficheros de texto, autodetección de hardware limitada.
 - Sistema de paquetes: TGZ
 - Descarga gratuita: Si
 - <http://www.slackware.org>
 - <http://slax.linux-live.org/?lang=es>
- **SuSE - Novell Linux Desktop.** Siguen la política de no permitir descargar el software hasta tiempo después de que salgan a la venta las versiones comerciales.
 - A favor: Atención profesional en cada detalle, herramienta de configuración de fácil uso (YaST), posibilidad de integrar con el directorio activo de Novell.
 - En contra: Solo disponible en algunas partes del mundo en las tiendas de software o mediante instalación FTP, incluye componentes propietarios, que no permiten su redistribución.
 - Sistema de paquetes: RPM
 - Descarga gratuita: SuSE no proporciona imágenes ISO para descarga, no obstante la versión Profesional de su distribución es accesible para la instalación FTP normalmente 1 o 2 meses más tarde de la versión oficial. La instalación mediante FTP no es difícil, pero requiere una buena conexión.
 - <http://www.suse.de>
- **Xandros.** Xandros Desktop es sin lugar a dudas la distribución más sencilla de usar del mercado, y altamente recomendable para los nuevos usuarios de Linux.
 - A favor: Diseñada para principiantes, funciona al instante, excelente administrados de archivos y otras utilidades.
 - En contra: Incluye software propietario, lo que no permite su redistribución, no hay descarga gratuita.
 - Sistema de paquetes: DEB
 - Descarga gratuita: No
 - <http://www.xandros.org>
- **Ubuntu:** Es una distribución de Linux de tipo escritorio, basada en Debian.
 - A favor: 100% libre, web y recursos de la comunidad excelentes, instalación de software sencillísima usando apt-get.
 - En contra: La versión estable no está actualizada.
 - Sistema de paquetes: DEB
 - Descarga gratuita: Si
 - <http://www.ubuntu.com>

9. Evaluación de los procesos y resultados de aprendizaje

9.1. Sistema de evaluación

En la evaluación de esta asignatura se tendrá en cuenta la parte de prácticas, los trabajos realizados, la implicación en clase y la corrección de trabajos ajenos. Cada una de las partes pesará en la nota final de acuerdo a los siguientes porcentajes:

Trabajos complementarios:	30%
Calificación de trabajos ajenos:	10%
Participación en clase:	10%
Nota de prácticas:	50%

Cada una de las partes "se guardan" para todas las convocatorias pertenecientes a el curso académico (febrero, septiembre y, en su caso, enero del año siguiente).

En el caso de que un estudiante no supere la asignatura en una de las convocatorias oficiales (febrero, septiembre o enero), constarán como suspenso únicamente aquellos estudiantes que se presenten al examen práctico de la asignatura correspondiente a esa convocatoria y no aprueben la asignatura.

De acuerdo con los porcentajes indicados al principio de este apartado, la **nota de teoría** de la asignatura se calculará de acuerdo con la siguiente fórmula:

Nota final teoría = *Parte Trabajos + Parte Corrector + Parte Participación en clase*

Parte Trabajos = $(\text{Nota media trabajos} * 0,80 + \text{Nota exposición} * 0,2) * 0,6$

Parte Corrector = $\text{Suma}((\text{Nota Corrección grupo} * 0,66) + (\text{Nota Corrección personal} * 0,44)) / \text{N}^\circ \text{ total de trabajos expuestos en clase por todos los grupos} * 0,2$

Parte participación en clase = *hasta 2 puntos, asignados por el profesor en función de la asistencia a clase y la participación activa en la misma.*

Para poder aplicar la fórmula indicada para el cálculo de la nota de teoría **será necesario realizar todos los trabajos asignados, así como calificar un mínimo equivalente al 80% de los trabajos presentados en clase.**

En caso de no poder aplicarse la fórmula para el cálculo de la nota de teoría, el estudiante podrá realizar un **examen escrito** sobre el temario teórico, con ejercicios prácticos de teoría.

En cuanto a la **nota de prácticas**

- El **50 %** será de una **prueba práctica** con ordenador, en la que se pide demostrar los conocimientos adquiridos durante las horas prácticas.

- El otro **50 %** se obtendrá de la calificación de los **informes** y la **práctica obligatoria** que hay que presentar a lo largo del cuatrimestre.
- La nota final será el promedio de las notas obtenidas en ambos apartados, aunque para poder sumar hay que tener un mínimo de 4 sobre 10 en cada parte.

La calificación final de la asignatura

La nota final será el promedio de los dos bloques (teoría y práctica), teniendo en cuenta que no se hará media si no se obtiene un mínimo de 4 (sobre 10) en cualquiera de los 2 bloques.

9.2. Criterios de evaluación

La calificación se hará conforme a las siguientes pautas:

Matrícula de honor

- La nota final es superior o igual a 9,75 puntos
- El conocimiento y la comprensión de la materia se extienden más allá del trabajo cubierto por el programa.
- La comprensión conceptual es sobresaliente.
- Los problemas relacionados con la asignatura son resueltos con soltura y se es capaz de razonar alternativas de solución estableciendo comparativas entre ellas.
- La participación en las diferentes actividades, incluyendo su actuación como corrector y las clases, sobresale por su corrección y satisfacción.

Sobresaliente

- La nota final es superior o igual a 8,5 puntos
- El conocimiento y la comprensión de la materia son muy satisfactorios.
- La comprensión conceptual es sobresaliente.
- Los problemas relacionados con la asignatura son resueltos con soltura y se es capaz de razonar alternativas de solución.
- La participación en las diferentes actividades, incluyendo su actuación como corrector y las clases, ha sido muy correcta y muy satisfactoria.

Notable

- La nota final es superior o igual a 6,75 puntos
- El conocimiento y la comprensión de la materia son satisfactorios.
- La comprensión conceptual es notable.
- Los problemas relacionados con la asignatura son resueltos con soltura.
- La participación en las diferentes actividades, incluyendo su actuación como corrector y las clases, ha sido correcta y bastante satisfactoria.

Aprobado

- La nota final es superior o igual a 5 puntos

- El conocimiento y la comprensión de la materia son básicos.
- La comprensión conceptual es suficiente.
- Los problemas relacionados con la asignatura son resueltos con éxito razonable.
- La participación en las diferentes actividades, incluyendo su actuación como corrector y las clases, ha sido correcta, pero no siempre satisfactoria.

Suspenseo

- La nota final es inferior a 5 puntos
- El conocimiento y la comprensión de la materia no son aceptables.
- La comprensión conceptual es insuficiente.
- Los problemas relacionados con la asignatura no se resuelven de forma adecuada.
- La participación en las diferentes actividades, incluyendo su actuación como corrector y las clases, ha sido escasa y deficiente.

10. Análisis de coherencia de la guía docente

Este análisis permite condensar las diferentes relaciones existentes entre los objetivos y competencias a desarrollar, con el plan de trabajo del estudiante y el sistema de evaluación empleado. El análisis se ha estructurado en tres tablas separadas en función del tipo de objetivo y competencia analizados en cada caso: instrumental, interpersonal y sistémico.

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS		BLOQUES DE CONTENIDOS			PLAN DE TRABAJO DE LOS ESTUDIANTES	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Interpersonales		Unidad I	Unidad II	Unidad III		
OIP1	CIPTC1			2-4	5-8	1-6	Enseñanza presencial (Talleres de prácticas) Enseñanza no presencial (Realización de trabajos) Procedimientos Actividades en grupo Criterios Grado de destreza en trabajos en grupo
OIP2	CIPTC2 CIPTC3			2-4	5-8	1-6	Enseñanza presencial (Talleres de prácticas /Defensa de trabajos) Enseñanza no presencial (Realización de trabajos) Procedimientos Actividades en grupo Criterios Grado de destreza en trabajos en grupo
OIP3	CIPTR1-CIPTR4			2-4	5-8	1-6	Enseñanza presencial (Talleres de prácticas /Defensa de

						trabajos / Corrección trabajos) Enseñanza no presencial (Realización de trabajos)	Actividades en grupo Criterios Grado de destreza en trabajos en grupo
OIP4	CPTC4			5-8		Enseñanza presencial (Corrección trabajos) Enseñanza no presencial (Lectura otros trabajos)	Procedimiento Hoja de calificación Criterios Grado de destreza en la corrección

Tabla 12. Relación entre objetivos y competencias interpersonales

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS		BLOQUES DE CONTENIDOS			PLAN DE TRABAJO DE LOS ESTUDIANTES	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Instrumentales	Unidad I	Unidad II	Unidad III	Prácticas		
OI1	CIC1,CIC2	0,1	2,3,4	5-8	1-6	Enseñanza Presencial (Clases magistrales / Talleres de prácticas) Enseñanza no presencial (Trabajos en grupo) Tutorías	Procedimientos Talleres prácticos Defensa trabajos Criterios Grado de comprensión de los conceptos sobre Sistemas Informáticos
OI2 OI3	CIC4-CIC21	1	2,3,4			Enseñanza Presencial (Clases magistrales / Realización de ejercicios) Enseñanza no presencial (Trabajo en grupo Tipo 1) Tutorías	Procedimientos Defensa individual del trabajo realizado en grupo Criterios Grado de comprensión INDIVIDUAL de los sistemas de numeración y métodos de codificación
OI4 OI5 OI6 OI7 OI8 OI9	CIC10 -CIC21			5-8		Enseñanza presencial (Defensa de trabajos realizados) Enseñanza no presencial (Trabajos en grupo / Revisión bibliográfica / Consulta recursos en Internet) Tutorías	Procedimientos Tutorías, Defensa oral de los trabajos, Exposición pública de los trabajos, Corrección otros trabajos Criterios Detectar el grado de comprensión de la materia y la implicación en el grupo
OI10 OI11	CIC22 -CIC26				1-6	Enseñanza presencial (Talleres de prácticas) Enseñanza no presencial (Realización de informes y	Procedimientos Examen, Informes prácticas, Práctica obligatoria Criterios

						práctica obligatoria) Tutorías	Capacidad de manejo y administración del Sistema Operativo
OI12	CIC3			5-8	1-6	Enseñanza presencial (Defensa de trabajos, Talleres de prácticas) Enseñanza no presencial (Trabajos en grupo, Práctica obligatoria, Informes, Revisión bibliográfica, Consulta recursos en Internet) Tutorías	Procedimientos Trabajos, Examen, Informes, Práctica obligatoria, Criterios Ajuste de los documentos generados a las reglas establecidas.
Elementos transversales	CIM1-CIM5 CIT1-CIT7 CIL1-CIL2	0,1	2,3,4	5-8	1-6	Enseñanza presencial (Clases magistrales, Talleres de prácticas, Defensa de trabajos) Enseñanza no presencial (Trabajos en grupo, Realización de la práctica obligatoria, Realización de ejercicios, Revisión bibliográfica, Consulta recursos en Internet) Tutorías	Procedimientos Trabajos, Correcciones, Examen, Talleres, Práctica obligatoria Criterios Grado de destreza en la competencias transversales

Tabla 13 Relación entre objetivos y competencias instrumentales

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS		BLOQUES DE CONTENIDOS			PLAN DE TRABAJO DE LOS ESTUDIANTES	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Sistémicas		Unidad I	Unidad II	Unidad III		
OS1	CS1	0,1	2-4	5-8	1-6	Enseñanza presencial (Clases magistrales, Talleres de prácticas, Defensa de trabajos) Enseñanza no presencial (Trabajos en grupo, Realización de la práctica obligatoria, Revisión bibliográfica, Consulta recursos Internet) Tutorías	Procedimientos Trabajos en grupo Examen Talleres Práctica obligatoria Criterios Nivel de actuación en destrezas transferibles
OS2 OS3 OS5	CS2 CS5 CS6	0,1	2-4	5-8	1-6	Enseñanza presencial (Clases magistrales, Talleres de prácticas, Defensa de trabajos, Corrección trabajos)	Procedimientos Trabajos en grupo Examen

OS6						Enseñanza no presencial (Trabajos en grupo, Realización de la práctica obligatoria, Revisión bibliográfica, Consulta recursos Internet) Tutorías	Talleres Práctica obligatoria Hojas de calificación Criterios Nivel de precisión y eficiencia relacionados con la resolución de temas planteados
OS4	CS3 CS4	1	2-4	5-8	1-6	Enseñanza presencial (Clases magistrales, Talleres de prácticas, Defensa de trabajos) Enseñanza no presencial (Trabajos en grupo, Realización de la práctica obligatoria, Revisión bibliográfica, Consulta recursos Internet) Tutorías	Procedimientos Prácticas Trabajos en grupo Criterios Grado de análisis y evaluación de los procedimientos relacionados con la resolución de problemas

Tabla 14. Relación entre objetivos y competencias sistémicas

Guía Docente de Redes de Ordenadores

Versión 2.0 – 03/06/2006

Sergio Bravo Martín

Departamento de Informática y Automática

Escuela Politécnica Superior de Zamora – Universidad de Salamanca

Avda. Requejo, 33 – 49022 Zamora, España

Tfno. +34 980 545000. Ext. 3635

Fax. +34 980 545001

ser@usal.es

1. Contexto

1.1. Perfil de los créditos y su adecuación al perfil profesional y académico de la titulación

Para un ingeniero técnico en informática de gestión es imprescindible tener unos conocimientos robustos de cómo diseñar y construir una red de ordenadores así como aplicaciones que funcionen sobre ellas, para lo cual son necesarios conceptos referentes a la arquitectura, interconexión de redes y programación en red.

Las redes de ordenadores han estado presentes desde mediados de los años 70 pero es en estos últimos años es cuando hemos asistido a una espectacular expansión gracias a la red de redes, Internet. La proliferación de las redes y de todos los dispositivos de interconexión, el desarrollo de los servicios telemáticos avanzados, la implantación de redes privadas virtuales, la generalización en el uso de redes de área local y de las redes inalámbricas, así como la necesidad de gestionar y garantizar la seguridad de las comunicaciones, son algunos de los factores han venido a complicar el panorama actual del sector de las telecomunicaciones.

Las *Redes de Ordenadores*, posee profundas raíces en la Ingeniería de Telecomunicaciones, no obstante se considera con obligatoriedad el que los futuros ingenieros técnicos en informática de gestión conozcan las bases del diseño y construcción de redes a fin de poder comprender, diseñar y gestionar la innumerable cantidad de aplicaciones destinadas a este entorno de trabajo y explotación.

Las Redes de Ordenadores como asignatura dentro del Plan de Estudios actual, consta de 7,5 créditos LRU (4,5 teóricos y 3 prácticos). Las Redes de Ordenadores modernas son sistemas de elevada complejidad que requieren de un estudio prolongado en el tiempo y la práctica de ciertos conceptos fundamentales que guiarán su entendimiento. Para dicho estudio tomamos como base fundamental el modelo OSI de la ISO. Además, no se puede obviar la importancia que ha tenido y tiene Internet en la sociedad actual. Esto nos obliga a incluir un capítulo adicional destinado a la red de redes, arquitectura y protocolos TCP/IP. Otra consecuencia del alto grado de aceptación que han tenido las telecomunicaciones en la sociedad de la información es el aumento de la seguridad en redes.

Uno de los principales objetivos didácticos de esta asignatura es introducir al alumno a la complejidad del funcionamiento de las redes de computadoras. Por ello es de

obligado cumplimiento compaginar todos estos conceptos teóricos nuevos con unas sesiones de prácticas a medida que cubran los aspectos fundamentales del diseño, arquitectura y funcionamiento, así como la creación de software para las Redes de Ordenadores por medio de interfaces de programación de aplicaciones como *sockets*. Otro capítulo importante dentro de la formación práctica del alumno es la identificación del software disponible para la gestión de redes.

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

1.2. Ubicación y relaciones en el plan de estudios

La asignatura de Redes de Ordenadores se imparte en el primer cuatrimestre del tercer curso de la Ingeniería Técnica en Informática de Gestión, cuyo Plan de Estudios actual data de 2002. Se recomienda que el alumno haya cursado las asignaturas de primer y segundo curso. No obstante posee una relación más estrecha con ciertas asignaturas de este mismo Plan de Estudios e incluso con otras de la titulación de segundo ciclo Ingeniería en Informática, cuyo Plan de Estudios data de 1998.

Las dependencias e interrelaciones entre estas asignaturas se muestran en la Figura 1. En el establecimiento de estas dependencias se ha tenido en cuenta el factor tiempo, que claramente establece el orden lógico en el que se van a cursar las asignaturas.

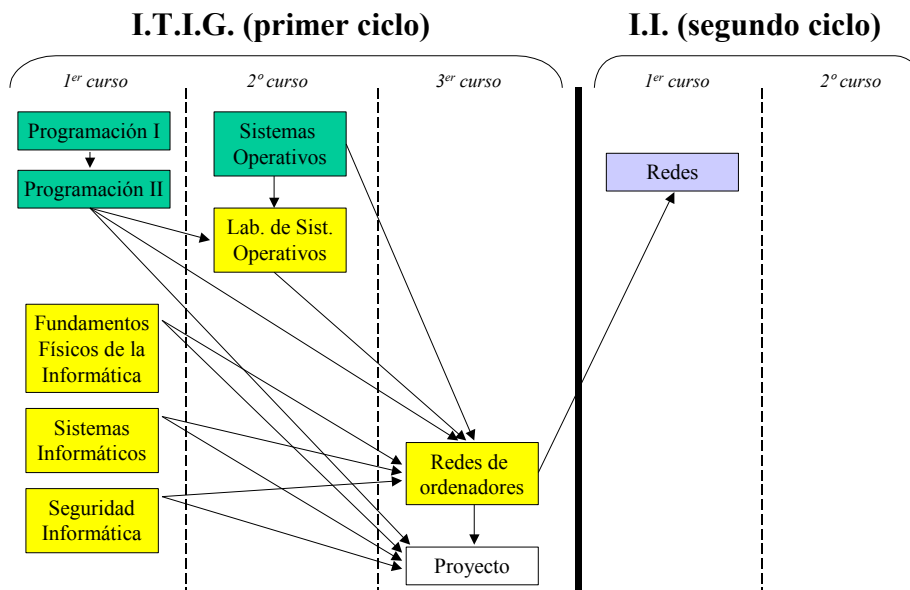


Figura 1. Relaciones de Redes de Ordenadores con otras asignaturas

2. Objetivos

2.1. Objetivos instrumentales generales

- OI1: Aplicar los conocimientos adquiridos mediante la resolución de problemas y de cinco bloques de prácticas.
- OI2: Utilizar con fluidez las herramientas y comandos para la gestión y monitorización de las Redes de Ordenadores.

- OI3: Adquirir y emplear un buen lenguaje formal, tanto oral como escrito, siendo riguroso en las explicaciones y exposiciones.
- OI4: Conocer, entender y utilizar la terminología utilizada en Redes de Ordenadores de una manera fluida y formal.
- OI5: Adquirir un buen manejo de la bibliografía recomendada en la asignatura, de forma que se potencie la autosuficiencia a la hora de completar la formación.
- OI6: Comprender el ámbito de las Redes de Ordenadores dentro de la Ingeniería Informática y dentro de los perfiles profesionales.
- OI7: Tomar conciencia de las implicaciones del trabajo individual y en equipo.
- OI8: Adquirir una visión inicial del campo de las Redes de Ordenadores, abordando los aspectos arquitectónicos de las comunicaciones desde una triple perspectiva: principios básicos de los sistemas abiertos, problemas de diseño en cada nivel y una discusión comprensiva de los estándares relacionados.
- OI9: Adquirir una buena destreza para la resolución de problemas teórico-prácticos sobre encaminamiento, control de flujo y direccionamiento (entre otros).
- OI10: Identificar los diferentes tipos de redes atendiendo a diversos criterios de clasificación como la disposición geográfica (redes de área local y redes de área extensa).
- OI11: Entender y utilizar los términos fundamentales de los modelos de referencia como elementos básicos del estudio de las redes de computadoras modernas.
- OI12: Conocer y comprender los términos referentes a los servicios y protocolos de los niveles arquitectónicos en que se asientan las Redes de Ordenadores.
- OI13: Mostrar un dominio práctico de conceptos fundamentales entre los que destacan la interconexión física de computadores en la creación de redes de área local y redes de área extensa.
- OI14: Introducir al alumno en los conceptos físicos relacionados con el envío de datos a través de diferentes medios de transmisión (digitales y analógicos).
- OI15: Adquirir un dominio y destreza en la creación de software para las Redes de Ordenadores.
- OI16: Adquirir la capacidad de crear documentaciones técnicas completas, correctas y legibles.
- OI17: Aplicar los conocimientos adquiridos en programación con *sockets* para la realización de una aplicación en red.

2.2. *Objetivos interpersonales generales*

- OIP1: Destrezas para la participación responsable: capacidad de coordinación, asistencia, contribuciones al grupo...
- OIP2: Capacidad de trabajar en equipo adquiriendo y mejorando las habilidades sociales y la inteligencia emocional.

OIP3: Comprometerse de forma ética con el trabajo, con el resto de los integrantes del grupo y consigo mismo.

2.3. *Objetivos sistémicos generales*

- OS1: Capacidad de integrar los conocimientos y destrezas prácticas de las diferentes asignaturas del plan de estudio para resolver situaciones reales relacionadas con la Ingeniería Técnica en Informática de Gestión, así como con otras disciplinas relacionadas.
- OS2: Reforzar el hábito de plantearse interrogantes. Ante un problema preguntarse por el número de soluciones, la relación entre ellas, cómo afectaría a las condiciones iniciales alguna modificación...
- OS3: Capacidad de aplicar y relacionar, de forma autónoma, los contenidos de las Redes de Ordenadores de forma interdisciplinar.
- OS4: Adquirir una comprensión del método científico, a través de las diversas actividades realizadas en la asignatura de Redes de Ordenadores, y asimilar su importancia como manera de pensar y actuar en la labor de científico e ingeniero, fomentando la capacidad de abstracción y el espíritu crítico.
- OS5: Desarrollar la madurez necesaria en el proceso de abstracción para abordar problemas reales y plantear modelos y soluciones de forma razonada y correcta.
- OS6: Reforzar el hábito de desarrollar diferentes alternativas, cuestionando las características, riegos y viabilidad de cada una, para cada problema planteado.

3. **Competencias**

3.1. *Competencias instrumentales*

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

3.1.1. Habilidades cognitivas

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

Generales

- CIC1: Conocer y entender las relaciones que mantiene la asignatura con el resto de asignaturas de la titulación a partir de la introducción explícita de ejemplos y comentarios, aprovechando puntos relevantes del temario.
- CIC2: Conocer y comprender algunas de las capacidades, aptitudes y conocimientos que la asignatura aporta para el desarrollo de los diversos perfiles profesionales.
- CIC3: Ser capaz de crear documentaciones técnicas completas, correctas y legibles.

Unidad Didáctica I: Conceptos básicos

- CIC4: Conocer la evolución que han ido experimentando las redes de ordenadores desde sus comienzos hasta nuestros días.
- CIC5: Entender el concepto de red de ordenadores.

- CIC6: Conocer las aplicaciones y objetivos de las redes de ordenadores y entenderlas como razones del crecimiento y evolución de las mismas.
- CIC7: Entender el modelo de comunicación como base fundamental para la comunicación entre dos interlocutores y en una red de ordenadores.
- CIC8: Conocer la estructura de la red y los elementos que forman parte de ella.
- CIC9: Ser capaz de identificar distintos tipos de redes atendiendo a diferentes criterios de clasificación.
- CIC10: Entender la necesidad de una normalización en el ámbito de las redes de ordenadores así como los organismos oficiales que se encargan actualmente de ello.
- CIC11: Conocer y comprender el modelo de OSI de la ISO como modelo arquitectónico de referencia para las redes de ordenadores. Identificar las funcionalidades y servicios de cada uno de los niveles que lo forman.
- CIC12: Conocer la terminología propuesta por OSI para el entendimiento de las funcionalidades y servicios de cada unos de los niveles.

Unidad Didáctica II: Arquitectura de redes

- CIC13: Entender la importancia y el alcance del nivel físico y de enlace dentro de la arquitectura de redes. Identificar sus funcionalidades y servicios.
- CIC14: Conocer y comprender las bases de la transmisión analógica y digital, así como las posibles perturbaciones a que puede verse sometida.
- CIC15: Identificar las diferentes técnicas de codificación y comprender los modos de transmisión en banda base y banda ancha.
- CIC16: Entender conceptos relativos a la velocidad y multiplexación del canal de comunicación.
- CIC17: Razonar acerca de cómo influye la utilización de las distintas técnicas de conmutación en la transmisión de información en una red de ordenadores.
- CIC18: Identificar el enlace de datos dentro del modelo de comunicación propuesto.
- CIC19: Entender las diferencias entre los servicios. Identificar al usuario y al proveedor del servicio dentro del modelo de referencia propuesto.
- CIC20: Conocer las fase en el proceso de intercambio de información.
- CIC21: Razonar sobre la necesidad de una sincronización, un control de errores y un control del flujo a nivel del circuitos de datos. Conocer y entender las distintas técnicas para cada una de las funcionalidades anteriores.
- CIC22: Razonar sobre los problemas de comunicación en canales de acceso múltiple y necesidad de una coordinación de la comunicación. Conocer y entender las distintas técnicas para tal efecto.
- CIC23: Identificar las diferencias clave entre el diseño de protocolos de comunicaciones para las redes de área local y de área extensa.
- CIC24: Razonar acerca de las ventajas que presentan las redes de área local frente a otras redes de ordenadores.

- CIC25: Conocer y comprender las características fundamentales de las topologías de redes de área local. Identificar las diferencias y similitudes entre ambos tipos de redes.
- CIC26: Conocer y comprender los esfuerzos de normalización sobre las redes de área local propuestos por el comité 802.x de la IEEE. Comprender la subdivisión del nivel de enlace en dos subniveles con el fin de proporcionar independencia sobre el medio de comunicación.
- CIC27: Identificar las normas 802.x de la IEEE referentes a redes de área local.
- CIC28: Conocer y comprender las características principales referentes al nivel físico y subniveles MAC y LLC del enlace de datos para las normas 802.3 (*Ethernet*), 802.4 (*TokenBus*), 802.5 (*TokenRing*) y FDDI.
- CIC29: Conocer las nociones sobre el cableado estructurado propuesto para redes de área local. Ser capaz de afrontar y resolver problemas de cableado de edificios y redes campus.
- CIC30: Identificar las características fundamentales de un Sistema Operativo en red. Razonar acerca de los ejemplos que conocemos actualmente.

Unidad Didáctica III: Interconexión de redes

- CIC31: Entender la necesidad del nivel de red como base fundamental para la interconexión de redes.
- CIC32: Identificar las funcionalidades y servicios proporcionados al nivel de transporte.
- CIC33: Entender y comprender las dos filosofías propuestas para la organización interna de la subred de comunicaciones. Razonar sobre el funcionamiento de dichas filosofías bajo varios escenarios.
- CIC34: Conocer y comprender los protocolos y direccionamiento de la subred propuestos por OSI.
- CIC35: Comprender la necesidad e importancia de un algoritmo de encaminamiento, sus características más importantes y métrica utilizada.
- CIC36: Identificar y comprender los algoritmos de encaminamiento propuestos.
- CIC37: Conocer y comprender las técnicas de que dispone el nivel de red para el control de la congestión.
- CIC38: Identificar las diferencias que podemos encontrarnos al intentar interconectar redes de ordenadores.
- CIC39: Conocer y comprender las características y funcionalidades de los diferentes elementos propuestos para la interconexión de redes. Razonar entre las diferencias de utilizar unos u otros elementos de interconexión.
- CIC40: Ser capaz de resolver problemas de interconexión de redes.
- CIC41: Identificar las funcionalidades y servicios proporcionados por el nivel de transporte.
- CIC42: Conocer y comprender los protocolos del nivel de transporte propuestos por el modelo de referencia OSI.

CIC43: Comprender el mecanismo de direccionamiento propuesto por el nivel de transporte para identificar a los extremos de la comunicación. Razonar acerca de la necesidad de multiplexación ofrecida por el nivel de transporte.

Unidad Didáctica IV: Internet

CIC44: Capacidad para analizar la evolución que ha ido experimentando la red Internet desde sus orígenes hasta nuestros días.

CIC45: Identificar las semejanzas y similitudes entre el modelo OSI y la red Internet.

CIC46: Conocer y comprender la encapsulación y demultiplexión realizada en la red Internet.

CIC47: Conocer y comprender las funcionalidades y servicios proporcionados por el nivel de red en Internet. Razonar las semejanzas con el nivel de red en OSI.

CIC48: Conocer el protocolo IP de Internet como protocolo de nivel de red e identificar como realiza las funciones básicas de dicho nivel.

CIC49: Entender el encaminamiento y direccionamiento a nivel de red que se lleva a cabo en Internet.

CIC50: Identificar la problemática surgida como consecuencia de la expansión del mundo Internet y comprender las soluciones propuestas, especialmente el concepto de subredes. Razonar la utilización de diferentes escenarios con la filosofía de subredes.

CIC51: Entender la necesidad de un protocolo para el control de mensajes en Internet y el reporte de errores.

CIC52: Conocer y comprender los protocolos a nivel de transporte que propone la red Internet. Analizar las semejanzas y similitudes con los protocolos propuestos por el modelo OSI.

CIC53: Comprender la noción de número de puerto para la identificación del punto final de la comunicación extremo a extremo.

CIC54: Comprender el modelo cliente-servidor como base fundamental para la construcción de aplicaciones en Internet.

CIC55: Conocer la estructura del nivel de aplicación y los protocolos elementales de Internet.

Unidad Didáctica V: Niveles superiores

CIC56: Conocer y comprender las funciones del nivel de sesión.

CIC57: Razonar sobre la necesidad y utilidad de estas funcionalidades en aplicaciones del mundo real.

CIC58: Conocer y comprender las funciones del nivel de presentación.

CIC59: Conocer el funcionamiento de conceptos del nivel de presentación en OSI como la representación de los datos. Analizar la necesidad de una sintaxis abstracta y una sintaxis de transferencia.

CIC60: Identificar la estructura del nivel de aplicación en OSI.

CIC61: Razonar sobre la necesidad de una normalización de protocolos básicos y más utilizados en el nivel de aplicación.

CIC62: Conocer la terminología OSI como base para la construcción de aplicaciones de usuario.

3.1.2. Capacidades metodológicas

CIM1: Ser capaz de tomar decisiones de manera razonada y consecuente con los conocimientos adquiridos.

CIM2: Tener capacidad de análisis y síntesis en la resolución de problemas del ámbito de las Redes de Ordenadores.

CIM3: Ser capaz de manejar bibliografía relacionada con las Redes de Ordenadores.

CIM4: Capacidad para aplicar los conceptos referentes a los servicios, funcionalidades y protocolos en la creación y diseño de redes de ordenadores modernas.

CIM5: Capacidad para documentar formalmente una solución completa a un problema de diseño de redes abordando las diferentes capas sobre las que ésta se asienta.

CIM6: Ser capaz de identificar y modelar un sistema de información que exija determinadas funcionalidades desde el punto de vista de las redes de ordenadores.

3.1.3. Destrezas tecnológicas

CIT1: Habilidades básicas de navegación por la Web y uso del resto de servicios de red para la obtención y manejo de la información relacionada con la asignatura y prácticas propuestas.

CIT2: Manejar las operaciones básicas y de administración de diferentes sistemas operativos para la instalación y ejecución de las diferentes herramientas de gestión y monitorización de la red utilizadas en la asignatura.

CIT3: Manejar con fluidez diferentes herramientas y comandos de gestión y monitorización de la red.

CIT4: Manejo avanzado de un sistema de procesamiento de textos para la realización de los informes de las prácticas propuestas en la asignatura.

CIT5: Conocimientos fluidos de programación en Lenguaje C bajo los sistemas operativos Windows y Unix.

3.1.4. Destrezas lingüísticas

CIL1: Adquirir y utilizar con fluidez un buen lenguaje científico, tanto oral como escrito, siendo riguroso en las explicaciones de cualquier concepto dentro de la asignatura.

CIL2: Conocer y utilizar la terminología usual de las Redes de Ordenadores, tanto en español como en inglés.

CIL3: Ser capaz de identificar y utilizar correctamente la mayor cantidad posible de acrónimos que disponemos en el mundo de las Redes de Ordenadores de la forma más adecuado posible.

3.2. Competencias interpersonales

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

3.2.1. Competencias para tareas colaborativas

CIPTC1: Ser capaz de realizar de trabajar en equipo para resolver los problemas propuestos en las prácticas cuatro y cinco propuestas en la asignatura. En primer lugar para la construcción de una aplicación en red y la segunda para la exposición en público de un trabajo relacionado con la asignatura.

CIPTC2: Ser capaz de presentar en público una solución a un problema planteado de cableado estructurado. Mantener un debate con el resto de la clase sobre la solución planteada, para así buscar colaborativamente la mejor solución al problema.

CIPTC3: Ser capaz de realizar y defender frente al profesor la práctica cuatro orientada a la construcción de una aplicación en red bajo la interfaz de programación de aplicaciones de *sockets* en los sistemas operativos Windows y Unix.

3.2.2. Compromiso con el trabajo

CIPTR1: Se ha de definir un plan de trabajo en el que el volumen de trabajo de todos los miembros del equipo sea similar.

CIPTR2: Una vez finalizado el trabajo, todos los miembros del grupo deben conocer en profundidad todo el desarrollo realizado.

CIPTR3: Se debe cumplir el plazo de entrega de los trabajos.

CIPTR4: Se debe adquirir un compromiso ético entre todos los componentes del grupo.

3.3. Competencias sistémicas

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

CS1: Capacidad de aplicar los conocimientos, métodos y herramientas vistos en la asignatura de Redes de Ordenadores a situaciones y problemas concretos del área de la Ingeniería Informática y de otras disciplinas relacionadas.

CS2: Capacidad de aprender y aplicar, de forma autónoma e interdisciplinar, nuevos conceptos y métodos relacionados con cada asignatura.

CS3: Motivación por la calidad y por la creatividad.

CS4: Capacidad de adoptar el proceso marcado por el método científico y de ingeniería en el planteamiento y realización de trabajos diversos, tanto a nivel académico como profesional.

CS5: Capacidad de asimilación y adaptación a la evolución del estado del arte en el ámbito de las Redes de Ordenadores y de la Ingeniería Informática como profesión.

4. Prerrequisitos

4.1. Competencias y contenidos mínimos

- Conocer los elementos básicos de un lenguaje de programación de 3ª generación.
- Conocer las principales formas de representar los algoritmos.
- Conocer el concepto de tipo abstracto de datos.
- Conocer los principios fundamentales del diseño de estructuras de datos.
- Conocer los principales mecanismos de comunicación entre procesos.
- Conocer las principales arquitecturas de sistemas informáticos.

4.2. Plan de trabajo para la consecución de los prerrequisitos

Los prerrequisitos para el estudio y entendimiento de la asignatura Redes de Ordenadores se cubren fundamentalmente en las asignaturas de Programación (I y II), Fundamentos Físicos de la Informática, Sistemas Informáticos de primer curso, y en las asignaturas de Sistemas Operativos y Laboratorio de Sistemas Operativos del segundo curso.

5. Temario

5.1. Unidades didácticas

Unidad Didáctica I: Conceptos básicos

Tema 1. Introducción a las Redes de Ordenadores

Tema 2. Normalización

Unidad Didáctica II: Arquitectura de Redes

Tema 3. Nivel Físico

Tema 4. Nivel de Enlace

Tema 5. Redes de área local

Unidad Didáctica III: Interconexión de redes

Tema 6. Nivel de Red

Tema 7. Interconexión de redes

Tema 8. Nivel de Transporte

Unidad Didáctica IV: Internet

Tema 9. Arquitectura y protocolos TCP/IP

Unidad Didáctica V: Niveles superiores

Tema 10. Niveles de Sesión y Presentación

Tema 11. Nivel de Aplicación

5.2. Desarrollo de las unidades didácticas

Unidad Didáctica I: Conceptos básicos

Tema 1. Introducción a las Redes de Ordenadores

- 1.1. Evolución histórica
- 1.2. ¿Qué es una red de ordenadores?
- 1.3. Aplicaciones
- 1.4. Objetivos
- 1.5. Ventajas e inconvenientes
- 1.6. Un modelo para las comunicaciones
- 1.7. Estructura de la red
- 1.8. Elementos de un sistema teleinformático
- 1.9. Criterios de clasificación y tipos de redes
- 1.10. Arquitectura de las redes

Tema 2. Normalización

- 2.1. Necesidad y vías para la normalización
- 2.2. Organismos internacionales de normalización
- 2.3. El modelo de referencia OSI de la ISO
- 2.4. Descripción de los niveles
- 2.5. Terminología OSI
- 2.6. Arquitectura de Internet

Unidad Didáctica II: Arquitectura de Redes

Tema 3. Nivel Físico

- 3.1. Introducción
- 3.2. Señales en el dominio del tiempo y de la frecuencia
- 3.3. Perturbaciones en la transmisión
- 3.4. Datos y señales
- 3.5. Transmisión analógica y digital
- 3.6. Capacidad de un canal
- 3.7. Medios de transmisión guiados y no guiados
- 3.8. Técnicas de codificación:
 - 3.8.1. Datos digitales – señales digitales
 - 3.8.2. Datos digitales – señales analógicas
 - 3.8.3. Datos analógicos – señales digitales
 - 3.8.4. Datos analógicos – señales analógicas
- 3.9. Transmisión en banda base y banda ancha
- 3.10. Conceptos relativos a la velocidad

- 3.11. Multiplexación
- 3.12. Sincronización
- 3.13. Explotación de un circuito de datos
- 3.14. Conmutación

Tema 4. Nivel de Enlace

- 4.1. Introducción
- 4.2. Los enlaces de datos
- 4.3. Servicios suministrados a la capa de red
- 4.4. Protocolos elementales de nivel de enlace
- 4.5. Clases de protocolos
- 4.6. Ejemplos de protocolos

Tema 5. Redes de área local

- 5.1. Introducción
- 5.2. Características
- 5.3. Topologías
- 5.4. El modelo OSI y las LAN
- 5.5. Las normas 802.X
- 5.6. FDDI
- 5.7. Cableado estructurado
- 5.8. Sistemas operativos de red

Unidad Didáctica III: Interconexión de redes

Tema 6. El nivel de red

- 6.1. Introducción
- 6.2. Técnicas de conmutación
- 6.3. Servicios suministrados a la capa de transporte
- 6.4. Primitivas de servicio
- 6.5. Protocolos de nivel de red en OSI
- 6.6. Direccionamiento de la subred
- 6.7. Organización interna de la capa de red
- 6.8. Encaminamiento
- 6.9. Congestión
- 6.10. Ejemplos: X.25, IP, etc.

Tema 7. Interconexión de redes

- 7.1. Introducción
- 7.2. Elementos de interconexión

7.3. Repetidores

7.4. Puentes

7.5. Pasarelas

Tema 8. Nivel de Transporte

8.1. Introducción

8.2. Diseño del nivel de transporte

8.3. Servicios suministrados al nivel de sesión

8.4. Calidad del servicio

8.5. Primitivas del servicio

8.6. Protocolo de transporte

8.7. Elementos del protocolo de transporte

8.8. Ejemplos de la capa de transporte

Unidad Didáctica IV: Internet

Tema 9. Internet

9.1. Evolución histórica

9.2. Características de Internet

9.3. Niveles Internet vs OSI

9.4. Direccionamiento en Internet

9.5. Nivel de red

9.6. Nivel de transporte

9.7. Números de puerto y modelo cliente-servidor

9.8. Correspondencia de direcciones

9.9. Nombres de dominio

9.10. Ejemplo práctico: el comando *ping*

9.11. ¿Cómo se accede a Internet?

9.12. La red Sideral/Internet

9.13. Los servicios de Internet

9.14. Programación con sockets

Unidad Didáctica V: Niveles superiores

Tema 10. Niveles de Sesión y Presentación

10.1. Introducción a los niveles superiores de OSI

10.2. El Nivel de Sesión

10.2.1. Introducción

10.2.2. Funciones del Nivel de Sesión

10.2.3. Gestión de sesiones

- 10.2.4. Gestión del diálogo
- 10.2.5. Sincronización
- 10.2.6. Gestión de actividades
- 10.2.7. Informe de excepciones
- 10.2.8. Primitivas del servicio
- 10.3. El Nivel de Presentación
 - 10.3.1. Introducción
 - 10.3.2. Normas y conceptos
 - 10.3.3. Representación de los datos
 - 10.3.4. Gestión de contextos
 - 10.3.5. ASN.1
 - 10.3.6. Seguridad
 - 10.3.7. Primitivas del servicio

Tema 11. Nivel de Aplicación

- 11.1. Introducción
- 11.2. Estructura del Nivel de Aplicación
- 11.3. Conceptos OSI
- 11.4. Elementos del servicio de aplicación
- 11.5. Correo electrónico
- 11.5. Servicio Directorio

6. Metodología y estrategias de aprendizaje

6.1. Metodología docente

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

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

- Convertirse en el responsable de su aprendizaje, debiendo gestionar y controlar la forma en que se produce.
- Desarrollar un espíritu crítico y una actitud abierta ante los cambios científico-técnicos de su especialidad, con curiosidad intelectual y rigor científico.
- Fomentar actitudes y adquirir técnicas para trabajar en equipo con eficacia.
- Basar en principios deontológicos su futura actividad profesional.

- Valorar el sentido humanístico de la ciencia y de la técnica como resultado del esfuerzo de generaciones anteriores, que debe servir para resolver los problemas del hombre actual y futuro.
- Estimular el perfeccionamiento profesional y la formación continua a lo largo de la vida.

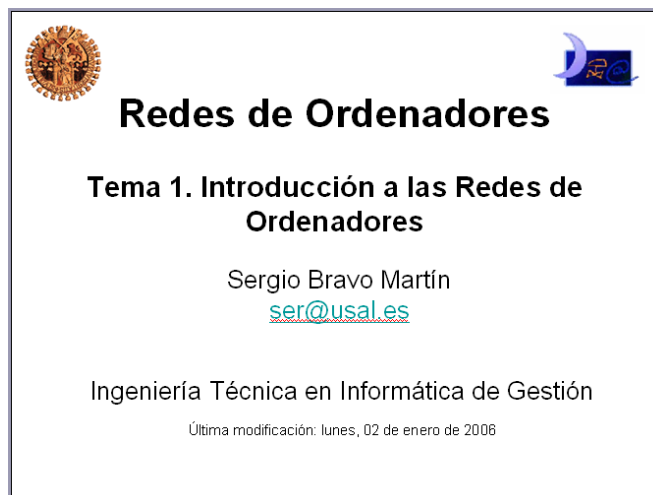
Siendo congruentes con lo anterior, y con carácter general, el Ingeniero en Informática debe estar capacitado para aprender a conocer, hacer, convivir y ser, en su ámbito personal, profesional y social, de acuerdo con lo recogido en el informe de la UNESCO sobre las perspectivas de la educación en el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

En el sentido de lo expuesto anteriormente, el modelo educativo que se va a seguir en la asignatura de Redes de Ordenadores de la titulación de Ingeniero Técnico en Informática de Gestión tiene un elemento importante en la clase magistral, pero ya no exclusivo, en la transmisión de conocimiento. Esta enseñanza se complementa con otros mecanismos entre los que podemos destacar las prácticas basadas en enseñanza colaborativa y estructuradas por bloques. Concretamente las actividades que se proponen son las siguientes:

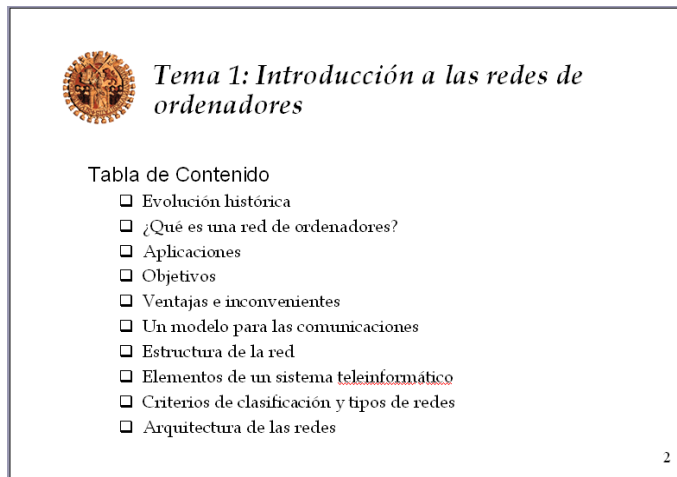
- *Clases de teoría con apoyo de material audiovisual.* En estas clases se presentarán los contenidos básicos de un cierto tema. Las clases comenzarán con una breve introducción de los contenidos que se pretenden transmitir en la clase, así como con un breve comentario a los conceptos vistos en clases anteriores y que sirven de enlace a los que se pretenden desarrollar. El desarrollo de la clase se llevará a cabo con medios audiovisuales, textos, transparencias... que permitan un adecuado nivel de motivación e interés en los alumnos. Se debe intentar motivar a los alumnos a intervenir en cualquier momento en las clases para hacer éstas más dinámicas y facilitar el aprendizaje. Es importante intentar terminar la exposición con las conclusiones más relevantes del tema tratado y un breve resumen de lo visto. Además se sugieren los capítulos de la bibliografía recomendada donde pueden consultar para ampliar/preparar más a fondo el tema en concreto.

Las transparencias utilizadas para las clases son una guía para el estudio, pero no son sustitutas de la bibliografía recomendada. Todos los temas comparten una misma estructura que se compone de los siguientes ítems:

- *Portada:* Con el título del tema y fecha de última modificación.



- *Esquema*: Con la tabla de contenido del tema.



- *Desarrollo del tema*: Con los apartados en los que se divide el tema.
- *Lecturas complementarias*: Lecturas opcionales para profundizar en el tema presentado.
- *Bloque de prácticas*. Las clases prácticas estarán dedicadas a poner de manifiesto y sobre escenarios reales gran parte de los conceptos teóricos expuestos en las clases teóricas. Por tal motivo, las prácticas están divididas en cinco bloques que se corresponden prácticamente con las cinco unidades didácticas de la asignatura. Cada uno de estos bloques consta de una presentación (disponible a través del Sitio Web de la asignatura) en donde se exponen los contenidos prácticos y los ejercicios propuestos. La duración de los bloques de prácticas se estructura por sesiones, siendo cada sesión de 2 horas. La distribución sesiones para cada uno de los bloques es la siguiente:
 - *Bloque I. Introducción a las redes de ordenadores*: una sesión.
 - *Bloque II. El enlace de datos, LAN y cableado estructurado*: tres sesiones.
 - *Bloque III. Comandos y herramientas de diagnóstico*: una sesión.
 - *Bloque IV. Programación de Aplicaciones bajo el modelo cliente-servidor: sockets*: cuatro sesiones.
 - *Bloque V. Nivel de aplicación*: dos sesiones.

Al finalizar cada bloque el alumno deberá cumplimentar un informe que deberá ser entregado al profesor en el plazo fijado al comienzo del curso. El formato del informe responde a una plantilla disponible en el Sitio Web de la asignatura y puede reutilizarse para su edición y entrega. Esta plantilla es de obligado cumplimiento y consta obligatoriamente de los apartados:

- Resumen de la práctica: nombre, alumnos, fecha, ...
- Tabla de Contenido
- Lista de Figuras
- Ejercicios
- Glosario

- *Entrega de ejercicios.* Se pondrá una fecha tope para la entrega de los informes correspondientes a cada uno de los bloques. Los informes de los bloques I, II y III son individuales, mientras que los bloques IV y V se realizarán en grupos de dos alumnos.
- *Tutorías.* El alumnado tiene a su disposición horas de tutorías en las que puede consultar cualquier duda relacionada con los contenidos, organización y planificación de la asignatura. Las tutorías pueden ser individualizadas, pero se admite tutorías grupales para resolver problemas relacionados con las actividades que se pueden realizar en grupos.
- *Sitio web de la asignatura.* Se convierte en el vehículo de comunicación y registro de información de la asignatura especialmente dedicado a la parte práctica de la misma. Se trata de un Sitio web diseñado para garantizar la transparencia de la asignatura y siguiendo las normativas de accesibilidad del nivel Triple-A del W3C-WAI.

6.2. Estrategias de aprendizaje

Se detallan las actuaciones concretas a realizar para la aplicación y aprovechamiento de la metodología propuesta. Se estructura en las siguientes fases:

6.2.1. Recopilación de la documentación de la asignatura

- Los alumnos dispondrán en el Sitio web de la asignatura de toda la información y normativa relacionada con la asignatura: temario, criterios de evaluación, bibliografía, apuntes, enunciados de prácticas, trabajos, enlaces de interés, lecturas complementarias, avisos...
- El profesor mantiene actualizada la información de este sitio web para que se convierta en un vehículo de comunicación con los alumnos.
- Los alumnos deben conocer y manejar con fluidez toda la información y la documentación que se integra en el sitio web de la asignatura.
- Asiduamente, el alumno debe consultar el sitio web para estar actualizado.

6.2.2. Planificación de las clases teóricas

- Las clases teóricas se dedicarán a la presentación de contenidos y a la discusión sobre las dudas que surjan durante las exposiciones.
- Las clases se desarrollarán según el siguiente esquema:
 1. El profesor presenta los objetivos a conseguir y el contexto en el que se va a desarrollar la clase.
 2. Se presentan los conceptos.
 3. Se aclaran las dudas cuando éstas surjan. El profesor puede incentivar el debate con preguntas para hacer la clase más participativa.
 4. El profesor termina con un resumen de los principales conceptos tratados en la clase y puede introducir la siguiente clase.
- El alumno debe haber realizado una lectura previa de los contenidos que se van a tratar en la clase, máxime cuando no todos los contenidos se desarrollan en las clases magistrales.

- Una vez terminada la clase magistral, se debe estudiar de forma autónoma su contenido y en caso de no entender algo intentar aclararlo utilizando la bibliografía recomendada o cualquiera de los materiales adicionales recomendados en la asignatura. También se dispone de las horas de tutorías a las que el alumno podrá acudir a solucionar dudas.

6.2.3. Planificación de las clases prácticas

- El profesor publicará en la página de la asignatura los contenidos del bloque y los enunciados de los ejercicios a entregar, con indicación de plazos de entrega, con suficiente antelación.
- En el caso de los trabajos en grupo el profesor decidirá las personas que formarán parte del mismo.
- El grupo o individuo, independiente del bloque de prácticas en cuestión, debe leer y entender bien todos los puntos antes de ponerse a llevarla a cabo.
- Los alumnos deben cumplir el calendario de entrega de prácticas.
- Las prácticas en grupo se defenderán individualmente.

6.2.4. Evaluación

- La calificación total de la asignatura se obtendrá a partir de la suma de la calificación en los respectivos bloques de prácticas y del resultado de una prueba final de acuerdo a la ponderación existente para los créditos prácticos y teóricos de la asignatura.

7. Planificación del tiempo y del esfuerzo

La asignatura de Ingeniería del Software tiene en el plan de estudios actual 7,5 créditos LRU asignados, 4,5 créditos de teoría y 3 créditos de práctica. Para hacer su estudio en ECTS se ha tomado tomando la equivalencia de crédito LRU a crédito ECTS, con implicación de cambio de metodología docente.

Al hacer el estudio de carga de trabajo máxima, se tiene que:

- 7,5 ECTS -> 187,5-225 horas de trabajo
- 4,5cr LRU -> 45 horas presenciales + (45*1,5) horas de asimilación = 112,5 horas
- 3cr LRU -> 30 horas presenciales + (30*2) horas de asimilación = 90 horas
- $112,5 + 90 = 202,5$ horas de trabajo
- $187,5 < 202,5 < 225$

Dos son los escenarios más probables: el resultante de tomar el crédito ECTS como 30 horas de trabajo y el resultante de tomarlo como 25 horas de trabajo.

Por temas de organización y calendario, se ha optado por partir del escenario que asigna 25 horas de trabajo al ECTS.

Para poder elaborar las guías, cuyo destino final sería la realización de experiencias piloto, dentro de los planes de estudio vigentes, se debe tener en cuenta la definición del crédito del R.D. 1497/1987 y sus posteriores modificaciones. Así:

- La unidad de valoración de las enseñanzas se corresponde a diez horas de enseñanza teórica, práctica o de sus equivalencias entre las que podrán

incluirse actividades académicas dirigidas, que habrán de preverse en el correspondiente plan docente junto con los mecanismos y medios objetivos de comprobación de los resultados académicos de las mismas. Todo ello sin perjuicio del cumplimiento del régimen de dedicación del profesorado, de conformidad con el Real Decreto 898/1985, de 30 de abril sobre régimen del profesorado universitario.

- En ningún caso, salvo que se trate de enseñanzas en Universidades a distancia, el porcentaje del crédito correspondiente a las actividades académicas dirigidas será superior al **30%**.

De esta forma, si 7,5 créditos LRU equivalen a 75 horas, se debe asegurar el 70% como mínimo de presencialidad, esto es, 52,5 horas entre horas de teoría y de práctica.

En la Tabla 1 se presenta la organización del esfuerzo del alumno para cubrir los 6 ECTS asociados a la asignatura. Por su parte en la Tabla 2 se distribuyen las horas presenciales entre los diferentes temas y el examen.

			A	B	C	D	E
	Técnica	Actividad	Horas equivalentes de clase	Factor de trabajo del alumno †	Horas de trabajo personal del alumno	Horas totales (A+C)	ECTS (D ÷ 25)
Teoría	Seguimiento de la parte teórica de las unidades didácticas	Asimila contenidos. Se plantea dudas que planteará a los profesores en las tutorías	37,0	2,0	74,0	111,0	4,4
Ejercicios de apoyo	Ejercicios de repaso y afianzamiento de lo aprendido. Pueden computar o no para la superación de la asignatura, dependiendo de la tipología de éstos	Realiza, resuelve problemas, tests...	-	-	4,0	4,0	0,2
Práctica Bloque I	Seguimiento del Bloque I. Práctica indiv.	Topología de subredes y	1,0	1,5	1,5	2,5	0,1
Práctica Bloque II	Seguimiento del Bloque II. Práctica indiv.	Simulaciones con los protocolos de nivel de enlace y ejercicio de cableado estructurado	4,0	2,0	8,0	12,0	0,5
Práctica Bloque III	Seguimiento del Bloque IV. Práctica indiv.	Ejercicios con los comandos y herramientas de red	1,5	2,0	3,0	4,5	0,2
Práctica Bloque IV	Seguimiento del Bloque V. Práctica en grupo	Aplicación en sockets bajo Win32 y Unix	6,0	5,0	30,0	36,0	1,4
Práctica Bloque V	Práctica en grupo	Informe y exposición	3,0	2,0	6,0	9,0	0,4
Examen	Controles de evaluación continua y/o examen final de la asignatura	Exámenes tipo test, supuestos prácticos...	3,0	-	-	3,0	0,1
Otras actividades	Tutorías personalizadas y grupales	Recibe orientación personalizada	-	-	3,5	3,5	0,1
	Búsquedas en la red, participación en foros especializados...	Busca elementos para completar los contenidos	-	-	2,0	2,0	0,1
TOTAL			55,5		132,0	187,5	7,5

Tabla 1. Escenario ECTS = 25 horas de trabajo

† Número de horas dedicadas por el alumno al trabajo personal (organización de apuntes, estudio, documentación...) por cada hora de clase.

Actividad	Horas Teoría	Horas Prácticas
Unidad Didáctica I: Conceptos básicos		
Tema 1: Introducción a las Redes de Ordenadores	2	1
Tema 2: Normalización	1	
Unidad Didáctica II: Arquitectura de Redes		
Tema 3: Nivel Físico	4	1
Tema 4: Nivel de Enlace	5	1
Tema 5: Redes de área local	4	2
Unidad Didáctica III: Interconexión de Redes		
Tema 6: Nivel de Red	5	0,5
Tema 7: Interconexión de redes	3	1
Tema 8: Nivel de Transporte	2	
Unidad Didáctica IV: Internet		
Tema 9: Arquitectura y protocolos TCP/IP	8	6
Unidad Didáctica V: Niveles superiores		
Tema 10: Niveles de Sesión y Presentación	2	
Tema 11: Nivel de Aplicación	1	3
Examen	3	
TOTAL: 52,5	40	15,5

Tabla 2. Reparto de horas presenciales

8. Bibliografía

8.1. Bibliografía básica

TANENBAUM, A.S. : Redes de Ordenadores, 4ed (Pearson Prentice Hall, 2003)

STALLINGS W.: Comunicaciones y Redes (Pearson Prentice Hall, 2004)

JAMES F. KUROSE, KEITH W. ROSS: Computer Networking: A Top-Down Approach Featuring the Internet (Pearson, 2005)

8.2. Bibliografía complementaria

COMER, D.E. Internetworking with TCP/IP: Principles, Protocols, and Architecture, Vol. I, 3rd. ed. Englewood Cliffs, New Jersey: Prentice Hall; 1995.

COMER, D.E. y STEVENS D.L. Internetworking with TCP/IP: Desing, implementation, and Internals, Vol. II, 2nd ed. Englewood Cliffs, New Jersey: Prentice Hall; 1994.

COMER, D.E. y STEVENS D.L. Internetworking with TCP/IP: Client-Server Programming and Applications, Vol. III, 2nd ed. Englewood Cliffs, New Jersey: Prentice Hall; 1996.

MAGAÑA E., IZME E., PRIETO M.: Comunicación y Redes de Computadores: Problemas y ejercicios resueltos (Pearson Prentice Hall, 2003)

8.3. Otros recursos

Tanenbaum, AS. : Redes de Ordenadores, 4ed (Pearson Prentice Hall, 2003)
<http://authors.phptr.com/tanenbaumcn4/> [Última vez visitado, 22/02/2018].

James F. Kurose, eith W. Ross: Computer Networking: A Top-Down Approach Featuring the Internet (pearson, 2005) <http://www.awl.com/kurose-ross/> [Última vez visitado, 22/02/2018].

8.4. Enlaces de interés

- **RedIRIS (Red académica y de investigación nacional)**
 - <http://www.rediris.es>
 - En el año 1988, el Plan Nacional de Investigación y Desarrollo puso en marcha un programa horizontal especial – IRIS- para la Interconexión de los Recursos Inform´taicos de las Universidades y centros de investigación, y desde su inicio hasta finales de 1993 la gestión del Programa IRIS corrió a cargo de Fundesco. A partir de 1991, cuando se considera finalizada una etapa de promoción y lanzamiento, IRIS se transforma en lo que es actualmente RedIRIS: la red académica y de investigación nacional que sigue siendo patrocinada por el Plan Nacional de I+D y que desde enero de 1994 hasta 2003 ha sido gestionada por el Consejo Superior de Investigaciones Científicas. A partir de enero de 2004 RedIRIS se integra como un departamento con autonomía e identidad propias en el seno de la Entidad Pública empresarial Red.es adscrita al Ministerio de Industria, Turismo y Comercio.
- **IEEE (Instituto de Ingenieros Eléctricos y Electrónicos)**
 - <http://www.ieee.org>.
 - La información más detallada, concisa y veraz que se puede encontrar acerca de las normas 802.x sobre redes de área local.
- **Grupo de Traducción de RFC's al español**
 - <http://www.rfc-es.org/>.
 - Este sitio está dedicado a la traducción al español de la documentación estándar sobre Internet conocida como RFC (Request For Comments).
- **The World Wide Web Consortium**
 - <http://www.w3.org/>.
 - Sitio central donde se recogen las especificaciones de los diversos lenguajes relacionados con la Web (HTML, XML, RDF, SOAP...).

8.5. Herramientas de gestión y monitorización de la red

- **VisualRoute**
 - <http://www.visualware.com>
- **Direccionamiento IP en Subredes**
 - <http://www.boson.com>
- **Ethereal**
 - <http://www.ethereal.com/>

- **IP Tools**
 - <http://avellano.usal.es/~sergio>

9. Evaluación de los procesos y resultados de aprendizaje

9.1. Sistema de evaluación

La evaluación de esta asignatura se divide en dos partes diferenciadas:

- **Parte de Teoría** (60% de la nota final).
 - Un examen final que consta de dos partes que hay que aprobar independientemente:
 - Un test.
 - Un conjunto de supuestos teórico/prácticos.
 - La parte de teoría se guardará durante el curso académico actual.
- **Parte Práctica** (40% de la nota final).
 - Se trata de los informes respectivos de los cinco bloques de prácticas. Siendo los bloques IV y V en grupos de dos alumnos.
 - Sólo requerirá defensa la práctica del bloque IV.
 - Todo grupo tiene derecho a dos defensas, siempre que la primera se haya hecho antes del mes de febrero.
 - La parte práctica se guardará durante el curso académico actual.

La nota final de la asignatura se calculará conforme a la siguiente fórmula:

Si (Teoría $\geq 5,0$) y (Práctica $\geq 5,0$)

Nota Final = (Teoría*0,6) + (Práctica*0,4)

Si no



Fin si

9.2. Criterios de evaluación

La calificación se hará conforme a las siguientes pautas:

Matrícula de honor

- La nota final es superior o igual a 9,75 puntos
- El conocimiento y la comprensión de la materia se extienden más allá del trabajo cubierto por el programa.
- La comprensión conceptual es sobresaliente.
- Los problemas relacionados con la asignatura son resueltos con soltura y se es capaz de razonar alternativas de solución estableciendo comparativas entre ellas.
- La participación en las diferentes actividades, incluyendo las clases, sobresale por su corrección y satisfacción.

Sobresaliente

- La nota final es superior o igual a 8,5 puntos
- El conocimiento y la comprensión de la materia son muy satisfactorios.
- La comprensión conceptual es sobresaliente.
- Los problemas relacionados con la asignatura son resueltos con soltura y se es capaz de razonar alternativas de solución.
- La participación en las diferentes actividades, incluyendo las clases, ha sido muy correcta y muy satisfactoria.

Notable

- La nota final es superior o igual a 6,75 puntos
- El conocimiento y la comprensión de la materia son satisfactorios.
- La comprensión conceptual es notable.
- Los problemas relacionados con la asignatura son resueltos con soltura.
- La participación en las diferentes actividades, incluyendo las clases, ha sido correcta y bastante satisfactoria.

Aprobado

- La nota final es superior o igual a 5 puntos
- El conocimiento y la comprensión de la materia son básicos.
- La comprensión conceptual es suficiente.
- Los problemas relacionados con la asignatura son resueltos con éxito razonable.
- La participación en las diferentes actividades, incluyendo las clases, ha sido correcta y pero no siempre satisfactoria.

Suspenso

- La nota final es inferior a 5 puntos
- El conocimiento y la comprensión de la materia no son aceptables.
- La comprensión conceptual es insuficiente.
- Los problemas relacionados con la asignatura no se resuelven de forma adecuada.
- La participación en las diferentes actividades, incluyendo las clases, ha sido escasa y deficiente.

10. Análisis de coherencia de la guía docente

Este análisis permite condensar las diferentes relaciones existentes entre los objetivos y competencias a desarrollar con el plan de trabajo del alumnado y el sistema de evaluación empleado. El análisis se ha estructurado en tres tablas separadas en función del tipo de objetivo y competencia analizados en cada caso: instrumental, interpersonal y sistémico.

	Instrumentales	Unidad I	Unidad II	Unidad III	Unidad IV	Unidad V	EVALUACIÓN	
O11 O13 O14 O111 O116 O17	CIC1-3	1, 2	3, 4, 5	6, 7, 8	9	10, 11	Enseñanza presencial: (Clases magistrales, sesiones de prácticas, defensa y exposición de prácticas) Enseñanza no presencial: (Trabajo individual y en grupo, realización de las prácticas, revisión bibliográfica, consulta de recursos en Internet) Tutorías	Procedimientos: Examen, Ejercicios, Trabajos e Informes de prácticas Criterios: Grado de comprensión y aplicación de los conceptos de las Redes de Ordenadores
O18 O110 O114	CIC4-12	1, 2	3, 4				Enseñanza presencial: (Clases magistrales, Bloque de prácticas I y II) Enseñanza no presencial: (Trabajo individual, consulta de recursos en Internet) Tutorías	Procedimientos: Examen, Trabajos e Informes de prácticas Criterios: Grado de comprensión y aplicación de los conceptos y herramientas de las Redes de Ordenadores
O18 O113	CIC13-30	1, 2	3, 4, 5				Enseñanza presencial: (Clases magistrales, Bloque de prácticas I y II) Enseñanza no presencial: (Trabajo individual, revisión bibliográfica, consulta de recursos en Internet) Tutorías	Procedimientos: Examen, Ejercicios, Trabajos e Informes de prácticas Criterios: Grado de comprensión y aplicación de los conceptos y herramientas de las Redes de Ordenadores
O18 O19 O12 O113	CIC31-43	1, 2	3, 4, 5	6, 7, 8	9		Enseñanza presencial: (Clases magistrales, Bloque de prácticas III) Enseñanza no presencial: (Trabajo individual, revisión bibliográfica, consulta de recursos en Internet) Tutorías	Procedimientos: Examen, Ejercicios, Trabajos e Informes de prácticas Criterios: Grado de comprensión y aplicación de los conceptos y herramientas de las Redes de Ordenadores
O15 O18 O115 O117	CIC44-55	1, 2	3, 4, 5	6, 7, 8	9		Enseñanza presencial: (Clases magistrales, Bloque de prácticas IV) Enseñanza no presencial: (Trabajo en grupo, revisión bibliográfica, consulta de recursos en Internet) Tutorías	Procedimientos: Examen, Ejercicios, Trabajos e Informes de prácticas, Defensa en grupo Criterios: Grado de comprensión y aplicación de los conceptos y herramientas de las Redes de Ordenadores
O15 O18 O112	CIC55 CIC56-62	1, 2	3, 4, 5	6, 7, 8	9	10, 11	Enseñanza presencial: (Clases magistrales, Bloque de prácticas V) Enseñanza no presencial: (Trabajo en grupo, revisión bibliográfica, consulta de recursos en Internet)	Procedimientos: Examen, Trabajos e Informes de prácticas, Exposición en público Criterios: Grado de comprensión y aplicación de los conceptos y herramientas de las Redes de

							Tutorías	Ordenadores
--	--	--	--	--	--	--	-----------------	-------------

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS		BLOQUES DE CONTENIDOS					PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Interpersonales		Unidad I	Unidad II	Unidad III	Unidad IV	Unidad V		
OIP1	CIPTC1-2			3, 4, 5	6, 7, 8	9		Enseñanza presencial (Sesiones de prácticas/Exposición y Defensa de prácticas)	Procedimientos Actividades en grupo Criterios Grado de destreza de trabajos en grupo
OIP2	CIPTC1-3			3, 4, 5	6, 7, 8	9		Enseñanza presencial (Sesiones de prácticas/Exposición y Defensa de prácticas)	Procedimientos Actividades en grupo Criterios Grado de destreza de trabajos en grupo
OIP3	CIPTR1-CIPTR4	1, 2	3, 4, 5	6, 7, 8	9	10, 11		Enseñanza presencial (Sesiones de prácticas/Exposición y Defensa de prácticas)	Procedimientos Actividades en grupo Criterios Grado de destreza de trabajos en grupo

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS		BLOQUES DE CONTENIDOS					PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Sistémicas		Unidad I	Unidad II	Unidad III	Unidad IV	Unidad V		
OS1	CS1	1, 2	3, 4, 5	6, 7, 8	9	10, 11	Enseñanza presencial (Clases magistrales/Sesiones de prácticas/Defensa y Exposición de prácticas) Enseñanza no presencial (Trabajo en grupo/Realización de prácticas/Revisión bibliográfica/Consulta de recursos en Internet) Tutorías	Procedimientos Ejercicios Examen Prácticas Trabajos y prácticas de otras asignaturas Proyecto fin de carrera Criterios Nivel de actuación en destrezas transferibles	
OS2 OS3 OS5	CS2 CS5	1, 2	3, 4, 5	6, 7, 8	9	10, 11	Enseñanza presencial (Clases magistrales/Sesiones de prácticas/Defensa y Exposición de prácticas)	Procedimientos Prácticas Trabajo en grupo	

							Enseñanza no presencial (Trabajo en grupo/Realización de prácticas/Revisión bibliográfica/Consulta de recursos en Internet) Tutorías	Ejercicios Criterios Nivel de precisión y eficiencia relacionados con la resolución de problemas nuevos
OS4 OS6	CS3 CS4	1, 2	3, 4, 5	6, 7, 8	9	10, 11	Enseñanza presencial (Clases magistrales/Sesiones de prácticas/Defensa y Exposición de prácticas) Enseñanza no presencial (Trabajo en grupo/Realización de prácticas/Revisión bibliográfica/Consulta de recursos en Internet) Tutorías	Procedimientos Prácticas Trabajo en grupo Ejercicios Criterios Grado de análisis y evaluación de los procedimientos relacionados con la resolución de problemas

Guía Docente de Sistemas Operativos Ingeniería Técnica en Informática de Gestión

Versión 2.0 – 02-09-2006

Susana Álvarez Rosado

Departamento de Informática y Automática

Escuela Politécnica Superior de Zamora – Universidad de Salamanca

Avda Requejo, 33, 49022, Zamora, España

Tfno. +34 980545000. Ext. 3635

Fax. +34 980 545001

sar@usal.es

1. Contexto

1.1. Perfil de los créditos y su adecuación al perfil profesional y académico de la titulación

La finalidad de Ingeniería Técnica en Informática de Gestión es formar profesionales capacitados para crear todo tipo de programas y aplicaciones informáticas que se ajusten a las necesidades sociales y de las empresas. Para ello se adentran en el conocimiento del ordenador y todos sus componentes, tanto a nivel hardware como a nivel software.

Los *Sistemas Operativos* como asignatura dentro del Plan de Estudios actual, consta de 6 créditos LRU (todos teóricos). Su cometido es presentar las funciones que tiene un sistema operativo dentro de un sistema informático. Además describe los problemas, principios, algoritmos y alternativas de diseño de los sistemas operativos de manera general.

Para dar una visión más concreta del mundo actual el enfoque que se da a la asignatura de *Sistemas Operativos* permite aplicar los conceptos estudiados (funcionalidades, algoritmos, estructuras, etc...) a dos sistemas operativos “genéricos”: los sistemas Windows y los sistemas tipo UNIX. Se realiza un estudio comparativo de las distintas alternativas estructurales y funcionales adoptadas por estos sistemas.

El estudio de los componentes fundamentales de un sistema operativo y de su funcionamiento, son básicos tanto para poder conocer la naturaleza de los diferentes problemas que puedan plantearse en un sistema informático, como para la búsqueda de soluciones en cualquier campo de la informática. De esta forma los conocimientos adquiridos en la asignatura *Sistemas Operativos* son claves para poder comprender y solucionar diferentes problemáticas de las disciplinas cursadas por un Ingeniero Informático.

Los conceptos de *Sistemas Operativos* darán un apoyo básico y fundamental a la asignatura “*Laboratorio de Sistemas Operativos*” que se cursa a continuación, y a la que compete realizar labores prácticas relacionadas con los sistemas operativos.

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

1.2. Ubicación y relaciones en el plan de estudios

La asignatura de *Sistemas Operativos* se imparte en el primer cuatrimestre del segundo curso de la Ingeniería Técnica en Informática de Gestión, cuyo Plan de Estudios actual data de 2002. Esta asignatura se relaciona directamente con otras de este mismo Plan de Estudios, pero además con otras de la titulación de segundo ciclo Ingeniería en Informática, cuyo Plan de Estudio data de 1998.

Las dependencias e interrelaciones entre estas asignaturas se muestran en la Figura 1. En el establecimiento de estas dependencias se ha tenido en cuenta el factor tiempo, que claramente establece el orden lógico en el que se van a cursar las asignaturas.

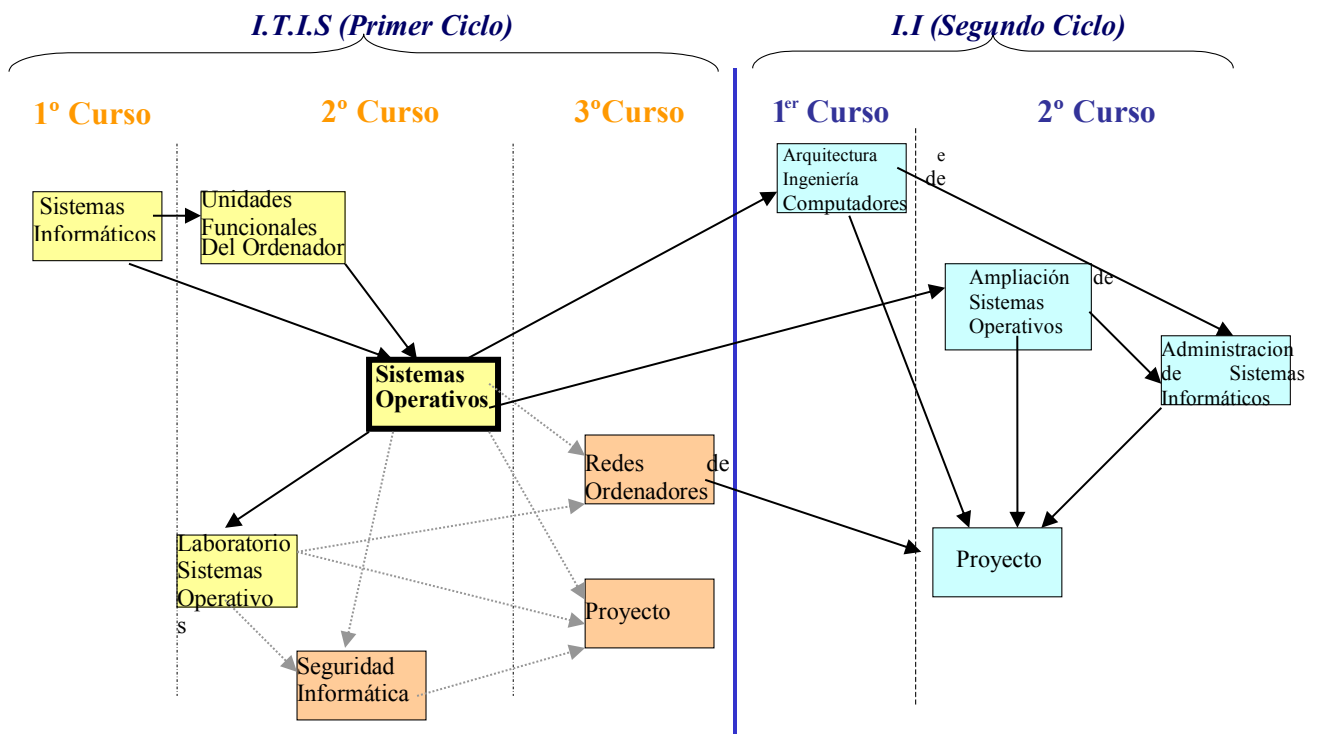


Figura 1. Relaciones de Sistemas Operativos con otras asignaturas

2. Objetivos

2.1. Objetivos instrumentales generales

- OI1: Aplicar los conocimientos adquiridos mediante la resolución de cuestiones teórico-prácticas expuestas en clase.
- OI2: Justificar y dar a conocer la función de un sistema operativo dentro de un sistema informático.
- OI3: Desarrollar la capacidad de evaluar las distintas alternativas de diseño empleadas en un sistema operativo.
- OI4: Conocer y comprender los conceptos, las abstracciones, y principios de diseño utilizados en sistemas operativos.

- OI5: Conocer y utilizar la terminología utilizada en Sistemas Operativos.
- OI6: Adquirir un buen manejo de la bibliografía recomendada en la asignatura, de forma que se potencia la autosuficiencia a la hora de completar la formación.
- OI7: Comprender el ámbito de los Sistemas Operativos dentro de la Ingeniería Informática y dentro de los perfiles profesionales actuales y futuros.
- OI8: Conocer la importancia del Sistemas Operativos y su conexión con el hardware.
- OI9: Adquirir una visión inicial del campo de los Sistemas Operativos (conceptos básicos y fundamentales).
- OI10: Estar capacitado para detectar deficiencias de un Sistema Operativo y poder dar soluciones de mejora a esas deficiencias.
- OI11: Conocer los principios de diseño de sistemas operativos desde varios puntos de vista relacionados con su estructura y funcionalidad.
- OI12: Conocer los principios de diseño de los sistemas operativos de las familias Windows y Unix. Ser capaces de diferenciarlos.
- OI13: Conocer, detectar y solucionar problemas de sincronización y concurrencia.
- OI14: Conocer y aplicar los principios de diseño de la gestión de memoria de un Sistema Operativo.
- OI15: Conocer y aplicar los principios de diseño de la gestión de procesos de un Sistema Operativo.
- OI16: Conocer y aplicar los principios de diseño de la gestión de entrada y salida de un Sistema Operativo.

2.2. Objetivos interpersonales generales

- OIP1: Participación en clase en la resolución de cuestiones teórico-prácticas.

2.3. Objetivos sistémicos generales

- OS1: Capacidad de integrar los conocimientos y destrezas prácticas de las diferentes asignaturas del plan de estudio para resolver situaciones reales relacionadas con la Ingeniería Informática, así como con otras disciplinas relacionadas.
- OS2: Reforzar el hábito de plantearse interrogantes. Ante un problema preguntarse por el número de soluciones, la relación entre ellas, cómo afectaría a las condiciones iniciales alguna modificación...
- OS3: Capacidad de aplicar y relacionar, de forma autónoma, los contenidos de Sistemas Operativos de forma interdisciplinar.
- OS4: Desarrollar la madurez necesaria para abordar problemas reales de los Sistemas Operativos y plantear modelos y soluciones de forma razonada y correcta.

OS5: Reforzar el hábito de desarrollar diferentes alternativas, cuestionando las características, riesgos y viabilidad de cada una, para cada problema planteado.

3. Competencias

3.1. Competencias instrumentales

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

3.1.1. Habilidades cognitivas

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

Generales

- CIC1: Conocer y entender las relaciones que mantiene la asignatura con el resto de asignaturas de la titulación a partir de la introducción explícita de ejemplos y comentarios, aprovechando puntos relevantes del temario.
- CIC2: Conocer y comprender algunas de las capacidades, aptitudes y conocimientos que la asignatura aporta para el desarrollo de los diversos perfiles profesionales.
- CIC3: Ser capaz de entender el funcionamiento de cualquiera de los módulos que componen un Sistema Operativo. Poder diseñar los módulos.

Unidad Didáctica I: Conceptos básicos

- CIC4: Conocer los elementos básicos de un ordenador.
- CIC5: Entender el proceso de ejecución de instrucciones por parte del procesador.
- CIC6: Entender el concepto de multiprocesamiento a grandes rasgos.
- CIC7: Entender la necesidad de algún mecanismo de protección del sistema y conocer los diferentes mecanismos que existen.
- CIC8: Conocer las diferentes técnicas para llevar a cabo las operaciones de Entrada y Salida, analizando sus ventajas e inconvenientes.
- CIC9: Conocer el concepto de sistema operativo, estructura, componentes y funciones.
- CIC10: Entender la evolución temporal y tecnológica de los Sistemas Operativos.
- CIC11: Aplicar los conceptos de esta unidad didáctica a dos familias de Sistemas Operativos conocidos: Unix y Windows.

Unidad Didáctica II: Gestión de Procesos

- CIC11: Entender el concepto de proceso y la necesidad de planificación.
- CIC12: Conocer los diferentes estados de un proceso, así como las posibles transiciones entre ellos y sus causas.
- CIC13: Reconocer las diferentes estructuras que tiene el Sistema Operativo para la gestión de procesos junto con su finalidad.

- CIC14: Conocer los diferentes servicios que ofrece el Sistema Operativo para la gestión de procesos.
- CIC15: Conocer y diferenciar los conceptos de cambio de proceso y cambio de contexto.
- CIC16: Entender el concepto de hilo diferenciándolo del concepto de proceso.
- CIC17: Hacer un estudio genérico de la gestión de procesos en las familias de sistemas: Unix y Windows.
- CIC18: Conocer los tres tipos de planificación de procesos que se llevan a cabo en un sistema. Diferenciarlas y comprender su frecuencia de actuación.
- CIC19: Conocer diferentes algoritmos de planificación de procesos y ser capaces de analizar sus ventajas e inconvenientes.
- CIC20: Conocer clases de análisis de rendimiento de los algoritmos de planificación de procesos.
- CIC21: Estudiar los algoritmos de planificación de procesos utilizados por las familias de sistemas : Unix y Windows.

Unidad Didáctica III: Sincronización y Concurrency

- CIC22: Entender el concepto de concurrencia y los problemas que acarrea.
- CIC23: Entender el concepto de exclusión mutua y sus requisitos.
- CIC24: Conocer las soluciones software a la exclusión mutua. Analizar progresivamente los algoritmos estudiando las ventajas e inconvenientes de cada uno de ellos.
- CIC25: Conocer las soluciones hardware a la exclusión mutua.
- CIC26: Comprender el concepto de semáforo y sus operaciones. Aplicar los conocimientos adquiridos para solucionar el problema de la exclusión mutua.
- CIC27: Entender el concepto de monitor y sus operaciones.
- CIC27: Conocer el funcionamiento del paso de mensajes con sus diferentes técnicas. Aplicarlo a la exclusión mutua.
- CIC28: Estudiar problemas clásicos de concurrencia y posibles soluciones.
- CIC29: Entender el concepto de interbloqueo y conocer las condiciones necesarias para que se de.
- CIC30: Conocer la prevención de interbloqueos y sus técnicas.
- CIC31: Conocer la evitación de interbloqueos y sus técnicas. Conocer el algoritmo del banquero para la evitación de interbloqueos.
- CIC32: Conocer la detección de interbloqueos y sus técnicas.
- CIC33: Entender el concepto de recuperación de interbloqueos y sus técnicas.
- CIC34: Estudiar problemas clásicos de interbloqueos.

Unidad Didáctica IV: Gestión de Memoria

- CIC35: Conocer los conceptos fundamentales de la gestión de memoria.
- CIC36: Aprender los diferentes tipos de asignación contigua de memoria.
- CIC37: Entender los conceptos de paginación y segmentación, su funcionamiento y su implementación.
- CIC38: Conocer el concepto de memoria virtual y su necesidad.
- CIC39: Entender la paginación bajo demanda y su implementación en el sistema. Estudiar los fallos de página y sus implicaciones.
- CIC40: Conocer el reemplazo de páginas y sus técnicas. Conocer los diferentes algoritmos.
- CIC41: Aprender el concepto de conjunto residente y entender su gestión.
- CIC42: Estudiar la gestión de memoria aplicada a las familias de sistemas: Unix y Windows.

Unidad Didáctica V: Gestión de la Entrada/ Salida

- CIC43: Estudiar el hardware de entrada y salida
- CIC44: Aprender el diseño del software de entrada y salida.
- CIC45: Conocer la estructura y funcionamiento del disco magnético.
- CIC46: Conocer los tipos de organización de ficheros en un sistema operativo.
- CIC47: Aprender los diferentes tipos de gestión del área de datos, estudiando sus ventajas y sus inconvenientes.
- CIC48: Estudiar el sistema de ficheros de Unix y su funcionamiento.
- CIC49: Estudiar el sistema de ficheros Ms-dox y su funcionamiento.
- CIC50: Estudiar otros conceptos relacionados con los sistemas de ficheros: montaje, cuotas, herramientas... etc.

3.1.2. Capacidades metodológicas

- CIM1: Ser capaz de tomar decisiones de manera razonada.
- CIM2: Tener capacidad de análisis y síntesis.
- CIM3: Ser capaz de manejar bibliografía relacionada con los Sistemas Operativos.
- CIM4: Ser capaz de analizar un problema dado desde los diferentes puntos de vista estructurales del sistema operativo, pudiendo plantear diferentes soluciones a dicho problema.
- CIM5: Ser capaz de aplicar correctamente y manejar con comodidad los diferentes algoritmos aprendidos, siendo capaces de analizar los resultados y obtener mejoras de manera razonada.

3.1.3. Destrezas tecnológicas

CIT1: Habilidades básicas de navegación por la Web y uso del resto de servicios de red para la obtención y manejo de la información relacionada con la asignatura.

CIT2: Manejar las operaciones básicas y de administración de diferentes sistemas operativos para observar y comprobar conceptos aprendidos en clase.

3.1.4. Destrezas lingüísticas

CIL1: Adquirir y utilizar con fluidez un buen lenguaje científico, tanto oral como escrito, siendo riguroso en las explicaciones de cualquier proceso dentro de la asignatura.

CIL2: Conocer y utilizar la terminología usual de los Sistemas Operativos, tanto en español como en inglés.

3.2. Competencias interpersonales

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

3.2.1. Competencias para tareas colaborativas

CIPTC1: Ser capaz de realizar de trabajar en equipo para resolver las cuestiones teórico-prácticas propuestas en las clases de la asignatura.

CIPTC2: Ser capaz de presentar en público una solución a un problema planteado y mantener un debate con el resto de la clase sobre la solución planteada, para así buscar colaborativamente la mejor solución al problema.

3.2.2. Compromiso con el trabajo

CIPTR1: Se debe colaborar para la resolución de cuestiones teórico-prácticas, pudiendo así analizar las deficiencias. Con esto se descubren los aspectos a profundizar para el bien de todos los compañeros.

3.3. Competencias sistémicas

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

CS1: Capacidad de aplicar los conocimientos y algoritmos vistos en la asignatura de Sistemas Operativos a situaciones y problemas concretos del área de la Ingeniería Informática y de otras disciplinas relacionadas.

CS2: Capacidad de aprender y aplicar, de forma autónoma e interdisciplinar, nuevos conceptos y métodos relacionados con cada asignatura.

CS3: Capacidad de asimilación y adaptación a la evolución de los Sistemas Operativos y de la Ingeniería Informática como profesión.

4. Prerrequisitos

4.1. Competencias y contenidos mínimos

- Conocer los principios fundamentales del diseño de sistemas operativos.
- Conocer los elementos básicos de un sistema operativo y su funcionamiento.

- Conocer los algoritmos más importantes de sistemas operativos sus ventajas e inconvenientes.
- Analizar de manera razonada los diferentes problemas que se puedan dar proponiendo diferentes soluciones.

4.2. Plan de trabajo para la consecución de los prerrequisitos

Los prerrequisitos para el estudio y entendimiento de la asignatura Sistemas Operativos se cubren fundamentalmente en la asignatura Sistemas Informáticos del primer curso, y en la asignatura de Unidades Funcionales del Computador del segundo curso.

5. Temario

5.1. Unidades didácticas

Unidad Didáctica I: Conceptos básicos

Tema 1. Introducción a los sistemas informáticos

Tema 2. Introducción a los Sistemas Operativos

Unidad Didáctica II: Gestión de Procesos

Tema 3. Descripción y control de procesos

Tema 4. Planificación del procesador

Unidad Didáctica III: Gestión de la Concurrencia

Tema 5. Concurrencia: exclusión mutua

Tema 6. Concurrencia: interbloqueo e inanición

Unidad Didáctica IV: Gestión de Memoria

Tema 7. Gestión de la memoria principal

Tema 8. Gestión de la memoria virtual

Unidad Didáctica V: Gestión de Entrada y Salida

Tema 9. Gestión de la memoria secundaria

Tema 10. Sistema de ficheros

5.2. Desarrollo de las unidades didácticas

Unidad Didáctica I: Conceptos básicos

Tema 1. Introducción a los sistemas informáticos

- 1.1. Elementos básicos de un ordenador.
- 1.2. Visión a más alto nivel.
- 1.3. La memoria principal.
- 1.4. Los puertos de entrada/salida.
- 1.5. El procesador.
- 1.6. Ejecución de instrucciones.
- 1.7. Jerarquía de memorias. Memorias caché.
- 1.8. Multiprocesado.

- 1.9. Interrupciones.
- 1.10. Mecanismos de protección del sistema.
- 1.11. Modo usuario/supervisor.
- 1.12. Registros límite.
- 1.13. Temporizadores.
- 1.14. Técnicas para llevar a cabo las operaciones de E/S.
- 1.15. E/S programada.
- 1.16. E/S dirigida por interrupciones.
- 1.17. Acceso directo a memoria (DMA).
- 1.18. Memoria mapeada.

Tema 2. Introducción a los Sistemas Operativos

- 2.1. Funciones y objetivos de los sistemas operativos.
- 2.2. Evolución de los sistemas operativos.
- 2.3. Otros tipos de sistemas operativos.
 - 2.3.1. Máquinas con varios procesadores.
 - 2.3.2. Sistemas distribuidos.
 - 2.3.3. Sistemas en tiempo real.
- 2.4. Componentes de un sistema operativo.
 - 2.4.1. Gestión de procesos.
 - 2.4.2. Gestión de memoria.
 - 2.4.3. Seguridad y protección de la información.
 - 2.4.4. Planificación y gestión de recursos.
- 2.5. Estructura de un sistema operativo.
- 2.6. Sistemas de ejemplo.
 - 2.6.1. Unix.
 - 2.6.2. Windows

Unidad Didáctica II: Gestión de Procesos

Tema 3. Descripción y control de Procesos

- 3.1. Estados de un proceso.
 - 3.1.1. Creación y finalización de procesos.
 - 3.1.2. Un modelo de procesos con dos estados.
 - 3.1.3. Modelo de cinco estados.
 - 3.1.4. Procesos suspendidos.

- 3.1.5. Modelo de siete estados.
- 3.2. Implementación de procesos.
 - 3.2.1. Estructuras de control del sistema operativo.
 - 3.2.2. Bloque de control de procesos.
- 3.3. Control de procesos.
 - 3.3.1. Modos de ejecución.
 - 3.3.2. Servicios ofrecidos por el núcleo del S.O.
 - 3.3.3. Creación y finalización.
 - 3.3.4. Cambio de proceso.
 - 3.3.5. Cambio de contexto.
 - 3.3.6. Ejecución del sistema operativo.
- 3.4. Procesos e hilos.
 - 3.4.1. Varios hilos en un solo proceso.
 - 3.4.2. Otras estructuras.
- 3.5. Sistemas de ejemplo.
 - 3.5.1. Unix.
 - 3.5.2. Windows.

Tema 4. Planificación del Procesador

- 4.1. Tipos de planificación.
 - 4.1.1. Planificación a largo plazo.
 - 4.1.2. Planificación a medio plazo.
 - 4.1.3. Planificación a corto plazo.
- 4.2. Criterios de planificación.
- 4.3. Prioridades.
- 4.4. Algoritmos de planificación
 - 4.4.1. Primero en llegar, primero en ser servido.
 - 4.4.2. Turno rotatorio.
 - 4.4.3. Primero el proceso más corto.
 - 4.4.4. Primero el de menor tiempo restante.
 - 4.4.5. Primero el de mayor tasa de respuesta.
 - 4.4.6. Realimentación.
- 4.5. Comparativa de rendimientos.
 - 4.5.1. Análisis de colas.
 - 4.5.2. Modelos de simulación.

- 4.6. Planificación por reparto equitativo.
- 4.7. Planificación en multiprocesadores y en tiempo real.
 - 4.7.1. Planificación en multiprocesadores.
 - 4.7.2. Planificación en tiempo real.
- 4.8. Ejemplos de planificación.
 - 4.8.1. Unix.
 - 4.8.2. Windows.

Unidad Didáctica III: Sincronización y Concurrencia

Tema 5. Concurrencia: Exclusión mutua y Sincronización

- 5.1. Principios generales de la concurrencia.
 - 5.1.1. Definiciones.
 - 5.1.2. Interacción entre procesos.
 - 5.1.3. Requisitos para la exclusión mutua.
- 5.2. Exclusión mutua: soluciones por software.
 - 5.2.1. Algoritmo de Dekker.
 - 5.2.2. Algoritmo de Peterson.
- 5.3. Exclusión mutua: soluciones por hardware.
 - 5.3.1. Inhabilitación de interrupciones.
 - 5.3.2. Instrucciones especiales de la máquina.
- 5.4. Semáforos
- 5.5. Monitores.
- 5.6. Paso de mensajes.
 - 5.6.1. Cuestiones de diseño.
 - 5.6.2. Exclusión mutua.
- 5.7. Problemas clásicos.
 - 5.7.1. Productor y consumidor.
 - 5.7.2. La barbería.
 - 5.7.3. Lectores y escritores.
- 5.8. Sistemas de ejemplo.

Tema 6. Concurrencia: Interbloqueo e inanición

- 6.1. Caracterización de interbloqueos.
 - 6.1.1. Condiciones necesarias.
 - 6.1.2. Grafo de asignación de recursos.

- 6.2. Prevención de interbloqueos.
- 6.3. Evitación de interbloqueos.
 - 6.3.1. Varias instancias por tipo de recurso.
 - 6.3.2. Una instancia por tipo de recurso.
- 6.4. Detección de interbloqueos.
 - 6.4.1. Varias instancias por tipo de recurso.
 - 6.4.2. Una instancia por tipo de recurso.
- 6.5. Recuperación del estado de interbloqueo.
 - 6.5.1. Finalización de procesos.
 - 6.5.2. Expropiación de recursos.
- 6.6. Problemas clásicos.
 - 6.6.1. Atasco de tráfico.
 - 6.6.2. La cena de los filósofos.
- 6.7. Sistemas de ejemplo.
 - 6.7.1. Unix.
 - 6.7.2. Windows.

Unidad Didáctica IV: Gestión de Memoria

Tema 7. Gestión de la memoria principal

- 7.1. Conceptos fundamentales.
 - 7.1.1. Montaje (enlazado) de programas.
 - 7.1.2. Carga de programas.
 - 7.1.3. Montaje (enlazado) dinámico de programas.
 - 7.1.4. Carga dinámica.
 - 7.1.5. Intercambio (swapping).
 - 7.1.6. Superposiciones (overlays).
- 7.2. Asignación contigua de memoria.
 - 7.2.1. Particiones de tamaño estático.
 - 7.2.2. Particiones de tamaño dinámico.
 - 7.2.3. Reubicación y protección.
- 7.3. Paginación.
 - 7.3.1. Hardware.
 - 7.3.2. Realización de la tabla de páginas.
 - 7.3.3. Compartición de código.
 - 7.3.4. Protección.

7.4. Segmentación.

- 7.4.1. Hardware.
- 7.4.2. Realización de la tabla de segmentos.
- 7.4.3. Compartición de código.
- 7.4.4. Protección.
- 7.4.5. Fragmentación.
- 7.4.6. Paginación y segmentación combinadas.

Tema 8. Gestión de la memoria virtual

8.1. Memoria virtual.

8.2. Paginación bajo demanda.

- 8.2.1. Tabla de páginas y bit de validez.
- 8.2.2. Fallo de página.
- 8.2.3. Rendimiento de la paginación bajo demanda.
- 8.2.4. Objetivos del sistema operativo.
- 8.2.5. Carga de páginas. Prepaginación.

8.3. Reemplazo de páginas.

- 8.3.1. El bit de modificación.
- 8.3.2. Políticas de reemplazo de páginas.
- 8.3.3. FIFO. Primera en entrar, primera en salir.
- 8.3.4. Óptimo.
- 8.3.5. LRU. Menos recientemente usada.
- 8.3.6. Algoritmo de segunda oportunidad.
- 8.3.7. Algoritmo del reloj.
- 8.3.8. Comparación de rendimientos.

8.4. Gestión del conjunto residente.

- 8.4.1. Asignación de memoria principal.
- 8.4.2. Alcance del reemplazo de páginas.

8.5. Asignación variable y alcance global.

- 8.5.1. Estrategia de frecuencia de fallos de página.
- 8.5.2. Conjunto de trabajo muestreado a tiempo variable.

8.6. Control de carga.

8.7. Gestión de memoria principal en el i386.

8.8. Ejemplos de gestión de memoria.

- 8.8.1. Unix.

8.8.2. Windows.

Unidad Didáctica IV: Gestión de la Entrada – Salida

Tema 9. Gestión de la E/S. Memoria secundaria

- 9.1. Hardware de entrada/salida.
- 9.2. Tipos de organización de la entrada/salida.
 - 9.2.1. E/S programada.
 - 9.2.2. Mediante interrupciones.
 - 9.2.3. Mediante acceso directo a memoria (DMA).
- 9.3. Aspectos de diseño del software de E/S.
 - 9.3.1. Tratamiento de interrupciones.
 - 9.3.2. Manejadores de dispositivos (drivers).
 - 9.3.3. Software independiente del dispositivo.
 - 9.3.4. Software de usuario.
- 9.4. El disco magnético.
 - 9.4.1. Estructura física de los discos magnéticos.
 - 9.4.2. Parámetros de rendimiento de un disco magnético.
 - 9.4.3. Planificación de discos magnéticos.
 - 9.4.3.1. FCFS, SSTF, SCAN, C-SCAN, LOOK, C-LOOK.
 - 9.4.3.2. Comparación de rendimientos.
 - 9.4.4. Gestión de discos magnéticos.
- 9.5. Caché de disco.
 - 9.5.1. El búffer caché.
- 9.6. Área de swap o intercambio.
 - 9.6.1. Windows.
- 9.7. Particiones y arranque.
 - 9.7.1. Ordenadores compatibles IBM PC.

Tema 10. Sistemas de Ficheros

- 10.1. Organización de ficheros.
 - 10.1.1. Direccionamiento.
 - 10.1.2. Entrada de directorio.
- 10.2. Gestión del área de datos.
 - 10.2.1. Asignación contigua de bloques.
 - 10.2.2. Asignación de bloques enlazados.
 - 10.2.3. Asignación indizada.

- 10.2.4. Gestión del espacio libre.
- 10.3. Sistema de ficheros UNIX.
- 10.4. Sistema de ficheros MS-DOS (FAT).
- 10.5. Posibilidades opcionales.
 - 10.5.1. Montaje de unidades.
 - 10.5.2. Cuotas de disco.
 - 10.5.3. Ficheros especiales.
 - 10.5.4. Sistema de ficheros virtual.
 - 10.5.5. Herramientas auxiliares.

6. Metodología y estrategias de aprendizaje

6.1. Metodología docente

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

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

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

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

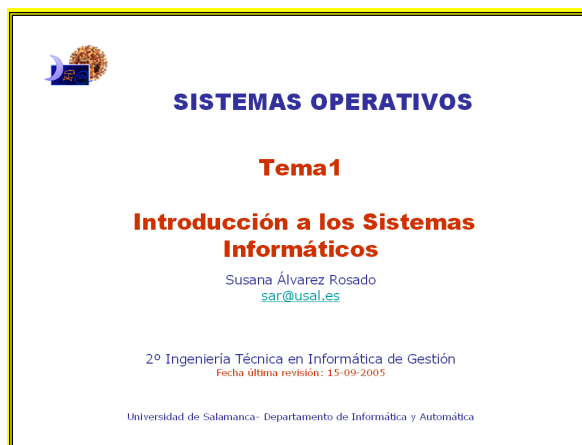
Para caminar en este sentido, el modelo educativo que se va a seguir en la asignatura de *Sistemas Operativos* tiene en la clase magistral un elemento importante, pero ya no exclusivo, en la transmisión de conocimiento. Este tipo de enseñanza se va a complementar con otros procesos entre los que cabe destacar la resolución de cuestiones

teórico–prácticas basadas en enseñanza colaborativa y organizadas dentro de las propias clases magistrales. Concretamente las actividades que se proponen son las siguientes:

- *Clases de teoría con apoyo de material audiovisual.* En estas clases se presentarán los contenidos básicos de un cierto tema. Las clases comenzarán con una breve introducción de los contenidos que se pretenden transmitir en la clase, así como con un breve comentario a los conceptos vistos en clases anteriores y que sirven de enlace a los que se pretenden desarrollar. El desarrollo de la clase se llevará a cabo con medios audiovisuales, textos, transparencias... que permitan un adecuado nivel de motivación e interés en los alumnos. Se debe intentar motivar a los alumnos a intervenir en cualquier momento en las clases para hacer éstas más dinámicas y facilitar el aprendizaje. Es importante intentar terminar la exposición con las conclusiones más relevantes del tema tratado.

Las transparencias que se utilizarán en clase son un subconjunto de las que se facilitan a los alumnos. Estas transparencias son una guía para el estudio, pero no son sustitutas de la bibliografía recomendada. Todos los temas comparten una misma estructura que se compone de los siguientes ítems:


- *Portada:* Con el título del tema y fecha de última modificación.



- *Resumen:* Con el resumen del tema, unos descriptores en forma de palabras clave y los capítulos de la bibliografía recomendada que deben consultar para ampliar/preparar el tema.
- *Esquema:* Con el índice del tema.

Sistemas Operativos- 2º ITIG

TEMA1 :Introducción a los Sistemas Informáticos



Índice

1. Elementos básicos de un ordenador.
 - 1.1 Visión a más alto nivel.
 - 1.2 La memoria principal.
 - 1.3 Los puertos de entrada/salida.
2. El procesador.
3. Ejecución de instrucciones.
4. Jerarquía de memorias. Memorias caché.
5. Multiprocesado.
6. Interrupciones.
7. Mecanismos de protección del sistema.
 - 7.1 Modo usuario/supervisor.
 - 7.2 Registros límite.
 - 7.3 Temporizadores.
8. Técnicas para llevar a cabo las operaciones de E/S.
 - 8.1 E/S programada.
 - 8.2 E/S dirigida por interrupciones.
 - 8.3 Acceso directo a memoria (DMA).
 - 8.4 Memoria mapeada.

Susana Álvarez Rosado- Departamento de Informática y Automática- Universidad de Salamanca

- *Desarrollo del tema:* Con los apartados en los que se divide el tema.
- *Aportaciones principales del tema:* A forma de resumen del tema, incluye las aportaciones y conclusiones más importantes del tema.
- *Ejercicios:* Conjunto de enunciados con cuestiones y ejercicios sobre el tema, se incluyen también ejercicios resueltos.
- *Lecturas complementarias:* Lecturas opcionales para profundizar en el tema presentado.
- *Referencias:* Lista de todas las referencias que se citan en el desarrollo del tema.
- *Clases teórico-prácticas.* Las clases prácticas presenciales estarán dedicadas a la resolución de cuestiones teórico-prácticas relativas a los Sistemas Operativos. Estas clases suceden a las clases teóricas de cada tema, y pretenden que el alumno afiance los conocimientos teóricos aprendidos en dichas clases. Las cuestiones se resolverán de manera colaborativa entre profesor y alumnos.
- *Tutorías.* El alumnado tiene a su disposición horas de tutorías en las que puede consultar cualquier duda relacionada con los contenidos, organización y planificación de la asignatura.
- *Página web.* Se convierte en el vehículo de comunicación y registro de información de la asignatura.

6.2. Estrategias de aprendizaje

Se detalla las actuaciones concretas a realizar para la aplicación y aprovechamiento de la metodología propuesta. Se estructura en las siguientes fases:

6.2.1. Recopilación de la documentación de la asignatura

- Los alumnos dispondrán en la página web de la asignatura información y normativa relacionada con la asignatura: temario, criterios de evaluación, bibliografía, enlaces de interés, lecturas complementarias, avisos...
- El profesor mantiene actualizada la información de esta página para que se convierta en un vehículo de comunicación con los alumnos.

- Los alumnos deben conocer y manejar con fluidez toda la información y la documentación que se integra en la página de la asignatura.
- Asiduamente, el alumno debe consultar la página para estar actualizado.

6.2.2. Planificación de las clases teóricas

- Las clases teóricas se dedicarán a la presentación de contenidos y a la discusión sobre las dudas que surjan durante las exposiciones.
- Las clases se desarrollarán según el siguiente esquema:
 1. El profesor presenta los objetivos a conseguir y el contexto en el que se va a desarrollar la clase.
 2. Se presentan los conceptos.
 3. Se aclaran las dudas cuando éstas surjan. El profesor puede incentivar el debate con preguntas para hacer la clase más participativa.
 4. El profesor termina con un resumen de los principales conceptos tratados en la clase y puede introducir la siguiente clase.
- El alumno debe haber realizado una lectura previa de los contenidos que se van a tratar en la clase, máxime cuando no todos los contenidos se desarrollan en las clases magistrales.
- Una vez terminada la clase magistral, se debe estudiar de forma autónoma su contenido y en caso de no entender algo intentar aclararlo utilizando la bibliografía recomendada o cualquiera de los materiales adicionales recomendados en la asignatura.
- El profesor podrá apoyar al alumno con material complementario al expuesto en clase y propuesto en la bibliografía.

6.2.3. Planificación de las clases teórico-prácticas

- El profesor propondrá cuestiones teórico-prácticas para su resolución.
- La resolución se llevará a cabo entre profesor y alumnos de manera colaborativa.
- El profesor podrá aportar adicionalmente información relativa a dichas cuestiones.
- El profesor podrá proponer cuestiones similares para el desarrollo individual por parte de cada alumno.

6.2.4. Evaluación

- Existirá una prueba final con cuestiones teórico prácticas, en la cual se valorará la capacidad del alumno para entender y aplicar los conceptos aprendidos en clase.

7. Planificación del tiempo y del esfuerzo

La asignatura de Sistemas Operativos tiene en el plan de estudios actual 6 créditos LRU asignados, todos ellos de teoría. Para hacer su estudio en ECTS se ha tomado tomando la equivalencia de crédito LRU a crédito ECTS, con implicación de cambio de metodología docente.

Al hacer el estudio de carga de trabajo máxima, se tiene que:

- 6 ECTS -> 150-180 horas de trabajo
- 6cr LRU -> 60 horas presenciales + (60*1,5) horas de asimilación = 150 horas de trabajo

Dos son los escenarios más probables: el resultante de tomar el crédito ECTS como 30 horas de trabajo y el resultante de tomarlo como 25 horas de trabajo.

Por temas de organización y calendario, se ha optado por partir del escenario que asigna 25 horas de trabajo al ECTS.

Para poder elaborar las guías, cuyo destino final sería la realización de experiencias piloto, dentro de los planes de estudio vigentes, se debe tener en cuenta la definición del crédito del R.D. 1497/1987 y sus posteriores modificaciones. Así:

- La unidad de valoración de las enseñanzas se corresponde a diez horas de enseñanza teórica, práctica o de sus equivalencias entre las que podrán incluirse actividades académicas dirigidas, que habrán de preverse en el correspondiente plan docente junto con los mecanismos y medios objetivos de comprobación de los resultados académicos de las mismas. Todo ello sin perjuicio del cumplimiento del régimen de dedicación del profesorado, de conformidad con el Real Decreto 898/1985, de 30 de abril sobre régimen del profesorado universitario.
- En ningún caso, salvo que se trate de enseñanzas en Universidades a distancia, el porcentaje del crédito correspondiente a las actividades académicas dirigidas será superior al **30%**.

De esta forma, si 6 créditos LRU equivalen a 60 horas, se debe asegurar el 70% como mínimo de presencialidad, esto es, 42 horas entre horas de teoría y de práctica.

En la Tabla 1 se presenta la organización del esfuerzo del alumno para cubrir los 6 ECTS asociados a la asignatura. Por su parte en la Tabla 2 se distribuyen las horas presenciales entre los diferentes temas y el examen.

			A	B	C	D	E
	Técnica	Actividad	Horas equivalentes de clase	Factor de trabajo del alumno [†]	Horas de trabajo personal del alumno	Horas totales (A+C)	ECTS (D ÷ 25)
Teoría	Seguimiento de la parte teórica de las unidades didácticas	Asimila contenidos. Se plantea dudas que planteará a los profesores en las tutorías	36	1,5	54	90	3,6
Práctica	Seguimiento de la parte práctica de las unidades didácticas	Experimenta y practica	6	2	12	18	0,72
Ejercicios de apoyo	Ejercicios de repaso y afianzamiento de lo aprendido.	Realiza, resuelve problemas,	-	-	35	35	1,4

[†] Número de horas dedicadas por el alumno al trabajo personal (organización de apuntes, estudio, documentación...) por cada hora de clase.

	Pueden computar o no para la superación de la asignatura, dependiendo de la tipología de éstos	tests...					
Examen	Examen final de la asignatura	Supuestos teórico-prácticos	3			3	0,12
Otras actividades	Tutorías personalizadas y grupales	Recibe orientación personalizada	-	-	2	2	0,08
	Búsquedas en la red, participación en foros especializados...	Busca elementos para completar los contenidos	-	-	2	2	0,08
TOTAL			45		100	150	6

Tabla 1. Escenario ECTS = 25 horas de trabajo

NÚMERO DE HORAS PRESENCIALES		
Actividad	Horas Teoría	Horas Prácticas
Unidad Didáctica I: Conceptos básicos		
Tema 1	2	
Tema 2	2	
Unidad Didáctica II: Gestión de Procesos		
Tema 3	3	1
Tema 4	4	1
Unidad Didáctica III: Gestión de la Concurrencia		
Tema 5	4	1
Tema 6	4	1
Unidad Didáctica IV: Gestión de memoria		
Tema 7	4	1
Tema 8	5	1
Unidad Didáctica IV: Gestión de Entrada y Salida		
Tema 9	3	1
Tema 10	3	1
Examen	3	
TOTAL: 45	37	8

Tabla 2. Reparto de horas presenciales

8. Bibliografía

8.1. Bibliografía básica

Stallings, W. (2005) *Sistemas Operativos: Aspectos internos y principios de diseño*, Prentice Hall.

Tanenbaum, Andrew S., *Sistemas Operativos Modernos*, Prentice-Hall.

Tanenbaum, Andrew S., *Sistemas Operativos, Diseño e Implementación*, Prentice-Hall.

Silberschatz, A., *Sistemas Operativos*, Pearson

Silberschatz, A. et al. *Operating System Concepts*, Addison-Wesley.

Carretero Pérez, J, y otros, (2001) *Sistemas Operativos. Una Visión Aplicada*, McGraw-Hill

Deitel, Harvey M., *Introducción a los Sistemas Operativos*, Addison-Wesley.

Finkel, Raphael A., *Fundamentos de los Sistemas Operativos*, Anaya.

Flynn, Ida, Ann McHoes, *Sistemas Operativos 3ª Edición*, Paraninfo Thomson Learning.

G. Nutt, (2004) *Sistemas Operativos 3ª Edición*, Pearson Education.

Perez F., Carretero Perez J (2003) , *Problemas de sistemas operativos: de la base al diseño*, MacGraw-Hill

Casillas A. , Iglesias I, *Sistemas Operativos: problemas y ejercicios resueltos*, Pearson educacion

8.2. Bibliografía complementaria

Bacon J., Harris T, (2003) *Operating Systems: Concurrent and Distributed Software Design*, Addison Wesley.

8.3. Otros recursos

- Librería de conceptos de Sistemas Operativos:
 - <http://gaia.ecs.csus.edu/~zhangd/oscal/oscal.htm>
 - Contiene animaciones de muchos de los algoritmos que se detallan en clase. El alumno puede simular el comportamiento de los algoritmos introduciendo sus propios datos.
 - Es útil para comprobar algunas de las cuestiones teórico-prácticas resueltas en clase.
- Centro de Recursos de Sistemas Operativos
 - <http://www.nondot.org/sabre/os/articles>
 - Una colección de documentos que cubren muchos de los conceptos de los Sistemas Operativos.
- Universo Gurú de UNIX
 - <http://www.ugu.com/>
 - Fuente de información de UNIX.
- Desarrollo de Windows
 - http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnanchor/html/anch_windowsdev.asp
 - Fuente de información del funcionamiento interno de Windows.

8.4. Enlaces de interés

- Libro de Tanenbaum A.
 - <http://www.cs.vu.nl/~ast/books/mos2/>
- Recursos para estudiantes del libro de Stalling W.
 - <http://williamstallings.com/OS/OS5e.html>
- Recursos para estudiantes del libro de Nutt G.
 - http://wps.aw.com/aw_nutt_opsys_3

9. Evaluación de los procesos y resultados de aprendizaje

9.1. Sistema de evaluación

La evaluación de esta asignatura se fundamenta en una prueba escrita de cuestiones teórico-prácticas.

Si (NotaPrueba \geq 5,0)

Nota Final = NotaPrueba

Si no



Fin

si

9.2. Criterios de evaluación

La calificación se hará conforme a las siguientes pautas:

Matrícula de honor

- La nota final es superior o igual a 9,75 puntos
- El conocimiento y la comprensión de la materia se extienden más allá del trabajo cubierto por el programa.
- La comprensión conceptual es sobresaliente.
- Los problemas relacionados con la asignatura son resueltos con soltura y se es capaz de razonar alternativas de solución estableciendo comparativas entre ellas.
- La participación en las clases, sobresale por su corrección y satisfacción.

Sobresaliente

- La nota final es superior o igual a 8,5 puntos
- El conocimiento y la comprensión de la materia son muy satisfactorios.
- La comprensión conceptual es sobresaliente.
- Los problemas relacionados con la asignatura son resueltos con soltura y se es capaz de razonar alternativas de solución.
- La participación en las clases, ha sido muy correcta y muy satisfactoria.

Notable

- La nota final es superior o igual a 6,75 puntos
- El conocimiento y la comprensión de la materia son satisfactorios.
- La comprensión conceptual es notable.
- Los problemas relacionados con la asignatura son resueltos con soltura.
- La participación en las clases, ha sido correcta y bastante satisfactoria.

Aprobado

- La nota final es superior o igual a 5 puntos
- El conocimiento y la comprensión de la materia son básicos.
- La comprensión conceptual es suficiente.
- Los problemas relacionados con la asignatura son resueltos con éxito razonable.
- La participación en las clases, ha sido correcta y pero no siempre satisfactoria.

Suspenso

- La nota final es inferior a 5 puntos
- El conocimiento y la comprensión de la materia no son aceptables.
- La comprensión conceptual es insuficiente.
- Los problemas relacionados con la asignatura no se resuelven de forma adecuada.
- La participación en las clases, ha sido escasa y deficiente.

10. Análisis de coherencia de la guía docente

Este análisis permite condensar las diferentes relaciones existentes entre los objetivos y competencias a desarrollar con el plan de trabajo del alumnado y el sistema de evaluación empleado. El análisis se ha estructurado en tres tablas separadas en función del tipo de objetivo y competencia analizados en cada caso: instrumental, interpersonal y sistémico.

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS		BLOQUES DE CONTENIDOS					PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Instrumentales		Unidad I	Unidad II	Unidad III	Unidad IV	Unidad V		
OI2	CIC1							Enseñanza presencial (Clases magistrales/ Realización de cuestiones teórico- prácticas)	Procedimientos Ejercicios Examen Criterios
OI5	CIC2	1,2							
OI6									
OI7	CIC4-CIC10								

O18 O19							Enseñanza no presencial (Revisión bibliográfica/ Consulta de recursos en Internet/ Realización de ejercicios) Tutorías	Grado de conocimiento de la necesidad de los Sistemas Operativos
O11 O14 Oi10 O111	CIC3	1,2	3,4	5,6	7,8	9,10	Enseñanza presencial (Clases magistrales/ Realización de cuestiones teórico-prácticas) Enseñanza no presencial (Revisión bibliográfica/ Consulta de recursos en Internet/ Realización de ejercicios) Tutorías	Procedimientos Ejercicios Examen Criterios Grado de análisis y evaluación de los procedimientos relacionados con la resolución de problemas
O112	CIC11 CIC17 CIC21 CIC42 CIC48 CIC49	2	3,4		8	9,10	Enseñanza presencial (Clases magistrales/ Realización de cuestiones teórico-prácticas) Enseñanza no presencial (Revisión bibliográfica/ Consulta de recursos en Internet/ Realización de ejercicios) Tutorías	Procedimientos Ejercicios Examen Criterios Grado de comprensión y aplicación de las particularidades funcionales y estructurales de estos Sistemas Operativos.
O113	CIC22-CIC34			5,6			Enseñanza presencial (Clases magistrales/ Realización de cuestiones teórico-prácticas) Enseñanza no presencial (Revisión bibliográfica/ Consulta de recursos en Internet/ Realización de ejercicios) Tutorías	Procedimientos Ejercicios Examen Criterios Grado de destreza para detectar problemas de concurrencia, analizarlos y plantear posibles soluciones.
O114	CIC35-CIC41				7,8		Enseñanza presencial (Clases magistrales/ Realización de cuestiones teórico-prácticas) Enseñanza no presencial (Revisión bibliográfica/ Consulta de recursos	Procedimientos Ejercicios Examen Criterios Grado de comprensión del funcionamiento de la gestión de memoria. Grado de destreza para

							en Internet/ Realización de ejercicios) Tutorías	el desarrollo y aplicación de los algoritmos.
OI15	CIC11-CIC16 CIC18-CIC20		3,4				Enseñanza presencial (Clases magistrales/ Realización de cuestiones teórico- prácticas) Enseñanza no presencial (Revisión bibliográfica/ Consulta de recursos en Internet/ Realización de ejercicios) Tutorías	Procedimientos Ejercicios Examen Criterios Grado de comprensión del funcionamiento de la gestión de procesos Grado de destreza para el desarrollo y aplicación de los algoritmos.
OI16	CIC43-CIC47 CIC50		4			9,10	Enseñanza presencial (Clases magistrales/ Realización de cuestiones teórico- prácticas) Enseñanza no presencial (Revisión bibliográfica/ Consulta de recursos en Internet/ Realización de ejercicios) Tutorías	Procedimientos Ejercicios Examen Criterios Grado de comprensión del funcionamiento de la gestión de procesos Grado de destreza para el desarrollo y aplicación de los algoritmos.
Elementos transversales	CIM1-CIM5 CIT1-CIT2 CIL1-CIL2	1,2	3,4	5,6	7,8	9,10	Enseñanza presencial (Clases magistrales/ Realización de cuestiones teórico- prácticas) Enseñanza no presencial (Revisión bibliográfica/ Consulta de recursos en Internet/ Realización de ejercicios) Tutorías	Procedimientos Ejercicios Examen Criterios Grado de destreza en la competencias transversales

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS		BLOQUES DE CONTENIDOS					PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Interpersonales		Unidad I	Unidad II	Unidad III	Unidad IV	Unidad V		
OIP1	CIPTC1			3,4	5,6	7,8	9,10	Enseñanza presencial (Realización de	Procedimientos

	CIPTR1						cuestiones teórico-prácticas	Actividades colaborativas Criterios Grado de destreza en la realización de cuestiones teórico-prácticas. Grado de colaboración participativa.
--	--------	--	--	--	--	--	------------------------------	---

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS		BLOQUES DE CONTENIDOS					PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Sistémicas		Unidad I	Unidad II	Unidad III	Unidad IV	Unidad V		
OS1	CS1		1,2	3,4	5,6	7,8	9,10	Enseñanza presencial (Clases magistrales/ Realización de cuestiones teórico- prácticas) Enseñanza no presencial (Revisión bibliográfica/ Consulta de recursos en Internet/ Realización de ejercicios) Tutorías	Procedimientos Ejercicios Examen Trabajos y prácticas de otras asignaturas Proyecto fin de carrera Criterios Nivel de actuación en destrezas transferibles
OS2 OS3 OS4 OS5	CS2 CS3		1,2	3,4	5,6	7,8	9,10	Enseñanza presencial (Clases magistrales/ Realización de cuestiones teórico- prácticas) Enseñanza no presencial (Revisión bibliográfica/ Consulta de recursos en Internet/ Realización de ejercicios) Tutorías	Procedimientos Ejercicios Criterios Nivel de precisión y eficiencia relacionados con la resolución de problemas nuevos

Guía Docente de Ingeniería del Software

(en las Titulaciones de Informática de la Universidad de Valladolid)

Miguel A. Laguna

Departamento de Informática

Universidad de Valladolid

1. Contexto y estructura general de la Ingeniería del Software en la Universidad de Valladolid

La *Ingeniería del Software*, que tradicionalmente ha aparecido dentro de la Ciencia de la Computación, en los últimos años está siendo considerada como una entidad curricular independiente, aunque con profundas raíces en la Ciencia de la Computación y las Matemáticas. De hecho, se ha definido un cuerpo de conocimiento para la Ingeniería del Software (*Software Engineering Body of Knowledge – SWEBOK*), y más recientemente se ha publicado el currículo público (2004) del *Computing Curriculum – Software Engineering* por parte de la acción conjunta entre IEEE-CS y ACM, que se mantiene como uno de los cinco perfiles profesionales que se mantiene en el reciente borrador (2005) del *Computing Curricula*, junto a los perfiles de *Computer Engineering*, *Computer Science*, *Information Systems* e *Information Technology*.

Se muestran a continuación las orientaciones generales de la situación actual de la disciplina en el marco de los planes de estudio de la Universidad de Valladolid.

Examinando los planes de estudio de las Ingenierías Técnicas en Informática, se observa la existencia de varias asignaturas que tienen la denominación específica de *Ingeniería del Software*, reflejadas en la siguiente tabla¹⁶:

Asignatura	Área de Conocimiento	Créditos	Carácter	Curso/Cuat.
Ingeniería Técnica en Informática de Sistemas				
Ingeniería del Software I	CCIA, LSI	6 (3T+3P)	Obligatoria	2º/2º
Ingeniería del Software II	CCIA, LSI	6 (3T+3P)	Obligatoria	3º/1º
Ingeniería Técnica en Informática de Gestión				
Ingeniería del Software I	CCIA, LSI	7,5 (4,5T+3P)	Troncal	3º/1º
Ingeniería del Software II	CCIA, LSI	6 (3T+3P)	Troncal	3º/2º
Ingeniería Informática (segundo ciclo)				
Ingeniería del Software I	CCIA, LSI	9 (6T+3P)	Troncal	4º/1º
Ingeniería del Software II	CCIA, LSI	9 (6T+3P)	Troncal	5º/1º

El descriptor general de las asignaturas de primer ciclo, especialidad de Gestión, señala como contenidos: *Diseño, propiedades y mantenimiento del software de Gestión. Planificación y gestión de proyectos informáticos. Análisis de aplicaciones de Gestión.*

¹⁶ CCIA .- Ciencias de la Computación e Inteligencia Artificial. LSI .- Lenguajes y Sistemas Informáticos.

Paralelamente, el descriptor de las asignaturas de primer ciclo, especialidad de Sistemas, indica: *Planificación y gestión de proyectos informáticos. Análisis de requisitos de aplicaciones informáticas. Diseño y mantenimiento de aplicaciones informáticas.*

Por su parte, el descriptor de las asignaturas de segundo ciclo incluye: *Análisis y definición de requisitos. Diseño, propiedades y mantenimiento del software. Gestión de configuraciones. Planificación y gestión de proyectos informáticos. Análisis de aplicaciones.*

Como puede observarse, salvo la referencia explícita a la *Gestión de configuraciones* y el matiz de *Gestión* en el caso de ITIG, los descriptores son muy similares. Por esta razón se puede y se debe planificar globalmente la disciplina en su conjunto. Por otro lado, los temas específicos de programación no están incluidos en los descriptores de estas asignaturas. En general, la formación en temas como lenguajes de programación, estructuras de datos, algoritmos, pruebas o bases de datos, está asignada a otras asignaturas del vigente plan de estudios.

El trabajo en equipo es una constante en la parte práctica de estas asignaturas, buscando que el alumno se acostumbre a formar parte de un equipo, a semejanza de como, probablemente, deberá desempeñar su vida profesional. Se busca que las asignaturas contribuyan a la formación del alumno más allá de los contenidos técnicos, potenciando su capacidad de comunicación, tanto de forma oral como escrita.

Las asignaturas de primer ciclo corresponden a una formación para estudiantes en Ingeniería Técnica, es decir, estudiantes no graduados, cuya formación debe estar orientada a los conceptos fundamentales, con una fuerte componente práctica. Los alumnos que cursen estas asignaturas deben recibir una formación completa, con independencia de su continuidad en un segundo ciclo.

El titulado superior, además de profundizar en los métodos y técnicas de desarrollo, incluyendo métodos formales, debe recibir una formación suficiente para asumir la planificación y dirección de un proyecto informático en situaciones reales. Se debe tratar aspectos de métricas y gestión de configuración.

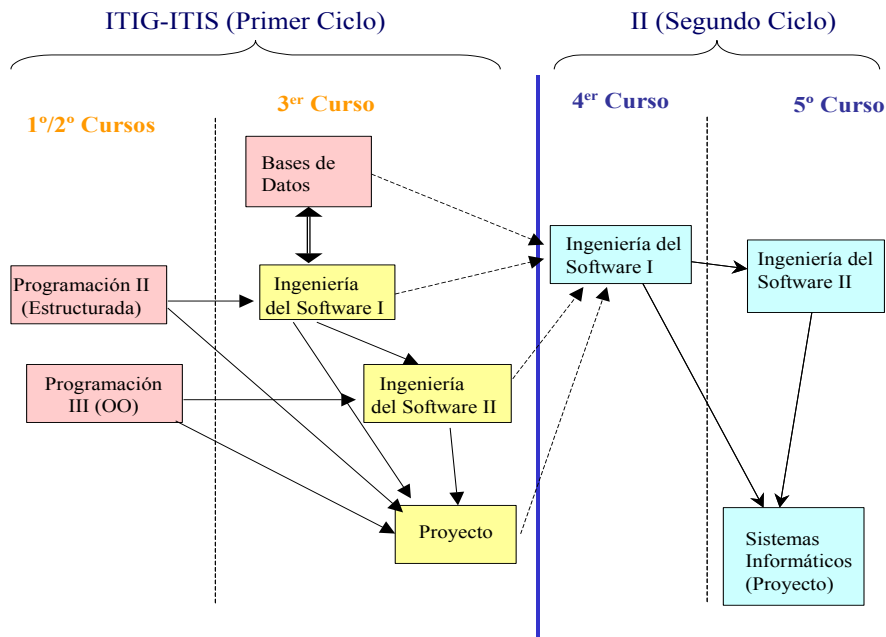


Figura 1. Relaciones de la Ingeniería del Software con otras asignaturas

Las dependencias e interrelaciones entre las asignaturas involucradas en la propuesta se muestran en la figura 1. La inclusión de las asignaturas de *Programación* se debe a que en ellas se tratan algunos de los conocimientos necesarios en Ingeniería del Software, incluyendo, además de temas de programación y pruebas unitarias, los conceptos básicos de diseño. De hecho, en *Programación II* y *III*, se incluyen contenidos sobre módulos, diagramas de estructura modular, manejo de bibliotecas de clases y frameworks (enlazando con API), ideas básicas sobre patrones de diseño, etc.

2. Objetivos y distribución de contenidos

El objetivo global, válido para el conjunto de todas las titulaciones consideradas, es producir profesionales que puedan resolver de forma sistemática la producción de software de calidad, que respondan a las necesidades y exigencias de las organizaciones, además de ser capaces de evaluar las nuevas tecnologías o comprender cómo pueden aplicarse de la mejor forma posible a la práctica del desarrollo del software. Se procurará potenciar en los alumnos las capacidades de:

- “Aprender a aprender” y “aprender a pensar”.
- Desarrollar un espíritu crítico y una actitud abierta ante cambios de todo tipo que afecten a la sociedad en la que vive y en especial a los cambios científico-técnicos de su especialidad.
- Fomentar actitudes y adquirir técnicas para un eficaz trabajo en equipo.
- Desarrollar actitudes de curiosidad intelectual y rigor científico.
- Basar en criterios deontológicos su futuro comportamiento en el ejercicio de la profesión.
- Estimular el perfeccionamiento profesional y la ampliación de estudios.

La distribución de los contenidos seleccionados entre primer y segundo ciclo, y a su vez en cada una de las asignaturas, obedece a los siguientes criterios:

- En el primer ciclo, y concretamente en *Ingeniería del Software I*, se pretende proporcionar al alumno una visión global de la materia que incluya todas las fases del ciclo de vida, desde el punto de vista de la gestión del proyecto (necesariamente a nivel introductorio) y de los métodos de análisis y diseño.
- Al mismo tiempo, para evitar la excesiva proliferación de técnicas, se ha optado por profundizar únicamente en el paradigma estructurado de desarrollo, al tiempo que se presenta como ejemplo la metodología Métrica.
- La asignatura *Ingeniería del Software II* de primer ciclo se dedica íntegramente al paradigma orientado a objetos y profundiza en el Lenguaje Unificado de Modelado (UML).
- En el segundo ciclo, teniendo en cuenta la procedencia de los alumnos (en su inmensa mayoría, titulados técnicos de Gestión o Sistemas, incluso cuando se trata de alumnos procedentes de otras universidades del entorno), se considera que los mismos tienen una base común mínima en Ingeniería del Software y un conocimiento amplio de orientación a objetos.
- Teniendo en cuenta el punto anterior, en el segundo ciclo se ha optado por presentar temporalmente la materia, no organizada por paradigmas sino por fases del ciclo de vida.
- La asignatura *Ingeniería del Software I* de cuarto curso, una vez introducida la materia, profundiza en el análisis y especificación de requisitos siguiendo los paradigmas de orientación a objetos y formal (con especial atención a problemas de tiempo real, concurrencia, etc.).
- Los conceptos avanzados de diseño (arquitecturas software, patrones de diseño), junto con la parte del programa que profundiza en los aspectos de gestión y métricas del software constituyen los contenidos de *Ingeniería del Software II* de quinto curso.

En cuanto a objetivos concretos, éstos se catalogan en tres apartados: *conceptos teóricos*, *aspectos prácticos* y *habilidades generales*. Los objetivos en el apartado de los conceptos teóricos son:

- T1** Actividades que se llevan a cabo en el ciclo de vida de un producto software.
- T2** Problemas, principios, métodos y tecnologías asociadas con la Ingeniería del Software.
- T3** Importancia de los requisitos en el ciclo de vida del software.
- T4** Obtención, documentación y especificación de los requisitos de un sistema software.
- T5** Especificaciones formales de requisitos.
- T6** Fundamentos del diseño de sistemas software.
- T7** Diseño de la interfaz gráfica de usuario.
- T8** Métodos de análisis y diseño orientado a objetos.
- T9** Métodos de análisis y diseño estructurado.
- T10** Conceptos, métodos y técnicas destinados al mantenimiento y evolución de los sistemas software.
- T11** Ingeniería del Software en dominios de aplicación específicos.
- T12** Gestión de proyectos software: definición de objetivos, gestión de recursos, estimación de esfuerzo y coste, planificación y gestión de riesgos.
- T13** Métricas software para el apoyo a la gestión de proyectos software y aseguramiento de la calidad del software.

Los objetivos en el apartado de los aspectos prácticos son:

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

Los objetivos en el apartado de habilidades generales son:

- H1** Mejora de la capacidad de comunicación.
- 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.

En cuanto a la distribución de contenidos, la tabla siguiente refleja las unidades didácticas que se imparten en el primer y segundo ciclo. La primera columna enumera las directrices originales que se han transcrito fielmente en los planes de estudio de la Universidad de Valladolid, junto con los objetivos propuestos. El resto de las columnas se refieren a los bloques o unidades didácticas de cada asignatura y, entre paréntesis y por referencia, se señalan las claves de las unidades de conocimiento de la propuesta *Computing Curricula* 2001 de ACM/IEEE-CS. Por último se citan las horas LRU que se utilizan en la actualidad como referencia.

Contenidos y objetivos	Ingeniería Técnica en Informática (primer ciclo)		Ingeniería Informática (segundo ciclo)	
	Ingeniería del Software I	Ingeniería del Software II	Ingeniería del Software I	Ingeniería del Software II
Conceptos básicos T1,T2	I. Introducción y conceptos básicos (SE4) 6 horas (4 en ITIS)	I. Ingeniería del Software orientada a objetos 4 horas	I. Introducción (SE4) 8 horas	
Análisis de aplicaciones [de Gestión]	II. Análisis de aplicaciones (SE5, IM3)	II. Análisis orientado a objetos (SE5)	II. El Lenguaje Unificado de Modelado (UML)	

Análisis y definición de requisitos (2° ciclo) T3,T4,T5	20 horas (14 en ITIS)	18 horas	(SE5, SE10) 32 horas III. Métodos Formales en Ingeniería del Software (SE5, SE10) 20 horas	
Diseño, propiedades y mantenimiento del software [de Gestión]. T6,T7,T10	III. Diseño de aplicaciones (SE1). 15 horas (10 en ITIS)	III. Diseño orientado a objetos (SE1) Incluye el diseño de la interfaz de usuario (HC1, HC2, SE2, SE9) 6 horas		I. Diseño y evolución del software (SE1) 20 horas II. Métricas del software (SE4, S10) 12 horas
Planificación y gestión de proyectos informáticos. Gestión de configuraciones (2° ciclo) T8,T9,T12,T13	IV. Proceso de desarrollo (SE4, SE6, SE7,SE8) 4 horas (2 en ITIS)	IV. Proceso orientado a objetos (SE4) 2 horas		III. Gestión de Proyectos software (SE8) 28 horas
Clases prácticas y de laboratorio T11, P1..8, H1..4	Ejercicios. Herramientas CASE (SE3, S12) 30 horas	Ejercicios. Herramientas OO (SE3, S12) 30 horas	Herramientas OO y herramientas formales (SE3, S12) 30 horas	Herramientas de métricas (SE3, S12) 30 horas

2.1. Objetivos instrumentales generales

- OI1: Aplicar los conocimientos adquiridos mediante la resolución de problemas y una práctica obligatoria.
- OI2: Utilizar con fluidez herramientas CASE.
- OI3: Adquirir y emplear un buen lenguaje formal, tanto oral como escrito, siendo riguroso en las explicaciones de cualquier proceso.
- OI4: Conocer y utilizar la terminología utilizada en Ingeniería del Software.

- OI5: Adquirir un buen manejo de la bibliografía recomendada en la asignatura, de forma que se potencia la autosuficiencia a la hora de completar la formación.
- OI6: Comprender el ámbito de la Ingeniería del Software dentro de la Ingeniería Informática y dentro de los perfiles profesionales.
- OI7: Tomar conciencia de las implicaciones del trabajo de ingeniero de software, tanto individualmente como formando parte de un equipo
- OI8: Adquirir una visión inicial del campo de la Ingeniería del Software.
- OI9: Tener una perspectiva global del proceso asociado al ciclo de vida de un producto software.
- OI10: Conocer los problemas, principios, métodos y herramientas propios de la Ingeniería del Software.
- OI11: Conocer los elementos básicos de un modelo objeto, soportado en el lenguaje unificado de modelado (UML).
- OI12: Conocer la importancia y la influencia de los requisitos en el éxito de un proceso software.
- OI13: Aplicar las técnicas básicas de ingeniería de requisitos.
- OI14: Conocer y aplicar los principios de un método de análisis y diseño orientado a objetos.
- OI15: Conocer y aplicar los principios de un método de análisis y diseño estructurado.
- OI16: Adquirir la capacidad de crear documentaciones técnicas completas, correctas y legibles.

2.2. *Objetivos interpersonales generales*

- OIP1: Destrezas para la participación responsable: capacidad de coordinación, asistencia, contribuciones al grupo...
- OIP2: Capacidad de trabajar en equipo adquiriendo y mejorando las habilidades sociales y la inteligencia emocional.
- OIP3: Comprometerse de forma ética con el trabajo, con el resto de los integrantes del grupo y consigo mismo.

2.3. *Objetivos sistémicos generales*

- OS1: Capacidad de integrar los conocimientos y destrezas prácticas de las diferentes asignaturas del plan de estudio para resolver situaciones reales relacionadas con la Ingeniería Informática, así como con otras disciplinas relacionadas.
- OS2: Reforzar el hábito de plantearse interrogantes. Ante un problema preguntarse por el número de soluciones, la relación entre ellas, cómo afectaría a las condiciones iniciales alguna modificación...
- OS3: Capacidad de aplicar y relacionar, de forma autónoma, los contenidos de Ingeniería del Software de forma interdisciplinar.
- OS4: Adquirir una comprensión del método científico, a través de las diversas actividades realizadas en la asignatura de Ingeniería del Software, y

asimilar su importancia como manera de pensar y actuar en la labor de científico e ingeniero, fomentando la capacidad de abstracción y el espíritu crítico.

OS5: Desarrollar la madurez necesaria en el proceso de abstracción para abordar problemas reales y plantear modelos y soluciones de forma razonada y correcta.

OS6: Reforzar el hábito de desarrollar diferentes alternativas, cuestionando las características, riegos y viabilidad de cada una, para cada problema planteado.

3. Competencias

3.1. Competencias instrumentales

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

3.1.1. Habilidades cognitivas

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

CIC1: Conocer y entender las relaciones que mantiene la asignatura con el resto de asignaturas de la titulación a partir de la introducción explícita de ejemplos y comentarios, aprovechando puntos relevantes del temario.

CIC2: Conocer y comprender algunas de las capacidades, aptitudes y conocimientos que la asignatura aporta para el desarrollo de los diversos perfiles profesionales.

CIC3: Ser capaz de crear documentaciones técnicas completas, correctas y legibles.

3.1.2. Capacidades metodológicas

CIM1: Ser capaz de tomar decisiones de manera razonada.

CIM2: Tener capacidad de análisis y síntesis.

CIM3: Ser capaz de manejar bibliografía relacionada con la Ingeniería del Software.

CIM4: Ser capaz de documentar la solución a un problema dado en las diferentes fases del ciclo de vida con los diferentes niveles de abstracción y aproximación que esto requiere.

CIM5: Ser capaz de aplicar correctamente y manejar con comodidad los diferentes elementos de los que dispone un lenguaje de modelado concreto, ya sea orientado a objetos u orientado a procesos.

CIM6: Ser capaz de modelar un sistema de información desde las diferentes perspectivas que exige un determinado proceso.

3.1.3. Destrezas tecnológicas

CIT1: Habilidades básicas de navegación por la Web y uso del resto de servicios de red para la obtención y manejo de la información relacionada con la asignatura.

- CIT2: Manejar las operaciones básicas y de administración de diferentes sistemas operativos para la instalación de las diferentes herramientas CASE a utilizar en la asignatura.
- CIT3: Manejar con fluidez diferentes herramientas CASE.
- CIT4: Manejo avanzado de un sistema de procesamiento de textos para la realización de los informes de las prácticas y la documentación de la práctica obligatoria.

3.1.4. Destrezas lingüísticas

- CIL1: Adquirir y utilizar con fluidez un buen lenguaje científico, tanto oral como escrito, siendo riguroso en las explicaciones de cualquier proceso dentro de la asignatura.
- CIL2: Conocer y utilizar la terminología usual de la Ingeniería del Software, tanto en español como en inglés.

3.2. Competencias interpersonales

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

3.2.1. Competencias para tareas colaborativas

- CIPTC1: Ser capaz de realizar de trabajar en equipo para resolver los problemas de modelado enunciados en los talleres prácticos de la asignatura.
- CIPTC2: Ser capaz de presentar en público una solución a un problema planteado y mantener un debate con el resto de la clase sobre la solución planteada, para así buscar colaborativamente la mejor solución al problema.
- CIPTC3: Ser capaz de realizar y defender la práctica obligatoria de la asignatura en equipo.

3.2.2. Compromiso con el trabajo

- CIPTR1: Se ha de definir un plan de trabajo en el que el volumen de trabajo de todos los miembros del equipo sea similar.
- CIPTR2: Una vez finalizado el trabajo, todos los miembros del grupo deben conocer en profundidad todo el desarrollo realizado.
- CIPTR3: Se debe cumplir el plazo de entrega de los trabajos.
- CIPTR4: Se debe adquirir un compromiso ético entre todos los componentes del grupo.

3.3. Competencias sistémicas

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

- CS1: Capacidad de aplicar los conocimientos, métodos y herramientas vistos en la asignatura de Ingeniería del Software a situaciones y problemas concretos del área de la Ingeniería Informática y de otras disciplinas relacionadas.
- CS2: Capacidad de aprender y aplicar, de forma autónoma e interdisciplinar, nuevos conceptos y métodos relacionados con cada asignatura.

- CS3: Motivación por la calidad y por la creatividad.
- CS4: Capacidad de adoptar el proceso marcado por el método científico y de ingeniería en el planteamiento y realización de trabajos diversos, tanto a nivel académico como profesional.
- CS5: Capacidad de asimilación y adaptación a la evolución del estado del arte en el ámbito de la Ingeniería del Software y de la Ingeniería Informática como profesión.

4. Prerrequisitos y co-requisitos

4.1. Competencias y contenidos mínimos

- Conocer los elementos básicos de un lenguaje de programación de 3ª generación.
- Conocer las principales formas de representar los algoritmos.
- Conocer el concepto de tipo abstracto de datos.
- Conocer los principios fundamentales del diseño de estructuras de datos.
- Conocer los principios fundamentales del modelado conceptual de datos mediante el diagrama entidad/relación.
- Conocer los mecanismos de transformación de un modelo conceptual de datos a un modelo lógico de datos.

4.2. Plan de trabajo para la consecución de los prerrequisitos

Los pre- y co-requisitos para el estudio y entendimiento de la asignatura Ingeniería del Software se cubren fundamentalmente en las asignaturas de Programación, Estructuras de Datos y Bases de Datos.

5. Temario

Se presenta a continuación el detalle general de todas las asignaturas actuales de primer y segundo ciclo, aunque más adelante se centrará el estudio en el primer ciclo, dado que presumiblemente será la base de la asignatura correspondiente al grado, dejando las asignaturas de segundo ciclo como componentes del posgrado.

5.1. Primer ciclo/grado

La asignatura de *Ingeniería del Software I* está enfocada a introducir los conceptos generales de la disciplina, definir el ciclo de vida del software y, como aplicación, al estudio de técnicas y métodos estructurados para el análisis y diseño de sistemas software, incidiendo en las peculiaridades y características de los sistemas de gestión. La asignatura se divide en cuatro bloques temáticos o unidades didácticas: *Conceptos básicos, Análisis de aplicaciones, Diseño de aplicaciones, Proceso de desarrollo*. En la siguiente tabla se aprecia el reparto de horas presenciales (entre paréntesis la alternativa de ITI Sistemas). Los objetivos específicos son:

- Comprender el proceso de diseño y desarrollo de los sistemas software desde el análisis de los requisitos de usuario hasta la implantación del sistema.

- Dominar las herramientas y procedimientos de trabajo de las metodologías estructuradas de análisis y diseño.
- Conocer la documentación que ha de generarse en las distintas fases de la concepción y desarrollo del proyecto.
- Ser capaz de elaborar a lo largo del curso la especificación completa de un sistema utilizando las herramientas, métodos y procedimientos mostrados en las clases teóricas.

Ingeniería del Software I		Horas Teor.	Horas Pract.
UD I	Introducción y conceptos básicos.	6 (4)	
Tema 1	Conceptos básicos de la Ingeniería del Software.	2(2)	
Tema 2	Desarrollo del software. El ciclo de vida.	4(2)	
UD II	Análisis de aplicaciones.	20 (14)	10
Tema 3	Análisis y definición de requisitos.	6(4)	3
Tema 4	Especificación de requisitos software.	14(10)	7
UD III	Diseño de aplicaciones.	15 (10)	5
Tema 5	Introducción al diseño.	5(4)	2
Tema 6	Diseño de la interfaz de usuario.	4(2)	
Tema 7	Diseño estructurado.	6(4)	3
UD IV	Proceso de desarrollo	4 (2)	
Tema 8	Las etapas del proceso. Metodología Métrica.	4(2)	
	Laboratorio de Ingeniería del Software I.		15
	Total	45 (30)	30

En la primera parte, introductoria, se intenta que el alumno tome conciencia de la importancia y necesidad de la Ingeniería del Software. El alumno debe comprender que deberá formar parte de un grupo de trabajo cuyo propósito sea el de desarrollar un sistema de información, de alta calidad, fácil de usar y que se ajuste a las necesidades del usuario.

Se introduce el concepto de sistema y de sistema de información del que forman parte personas, software, hardware, procedimientos, datos e información. Se proporciona una visión general de lo que puede ser la organización y dirección de un proyecto informático. Se estudia además el ciclo de vida del software, proporcionando una visión histórica de los ciclos de vida y un repaso a los vigentes en la actualidad. Por último, en esta unidad se proporcionan algunas nociones de dirección y control de proyectos informáticos.

En la unidad didáctica II, dedicada al análisis de requisitos, se proporcionan las técnicas y procesos necesarios para analizar el sistema en detalle. Se detallan las técnicas de elicitación de requisitos basados fundamentalmente en entrevistas, así como los soportes y métodos de análisis de los resultados de la investigación. Se estudian en detalle las técnicas de modelado de datos y tratamientos que se utilizan en el paradigma estructurado, que servirán para diseñar el sistema futuro.

En la parte III, dedicada al diseño se estudia la propuesta de soluciones técnicas que cumplan con las necesidades de la fase anterior. Se verán las técnicas y procedimientos adecuados para realizar

esta tarea. Se aborda el diseño arquitectónico, de datos y procesos así como su validación. Uno de los temas se dedica específicamente al diseño de la interfaz de usuario y otro al diseño estructurado.

Por último la unidad IV está dedicada al estudio de las metodologías de análisis y diseño estructurado actuales, profundizando en las características más sobresalientes de Métrica. Se analiza la documentación que hay que elaborar como ayuda a la comprensión, operatividad y mantenimiento del sistema creado. Al mismo tiempo se pretende completar la visión global de un proyecto informático y todas sus fases, apoyándose en el enfoque que plantea Métrica.

La asignatura de *Ingeniería del Software II* está enfocada al estudio de las técnicas y métodos no formales para el análisis y diseño de sistemas software siguiendo el paradigma de la orientación a objetos. Los objetivos específicos son:

- Dominar los conceptos y principios de la orientación a objetos.
- Comprender los elementos característicos de la ingeniería del software orientada a objetos.
- Conocer de forma detallada las técnicas y herramientas de modelado orientado a objetos.
- Presentar los principales conceptos del lenguaje estándar de modelado UML, sus reglas, sintaxis y semántica.
- Describir sintácticamente y semánticamente las vistas de UML.
- Aprender los principios generales de diseño de sistemas interactivos.
- Ser capaz de elaborar a lo largo del curso el diseño completo de un sistema utilizando las herramientas, métodos y procedimientos mostrados en las clases teóricas.

En la siguiente tabla se describe de forma esquemática la organización temporal de la asignatura *Ingeniería del Software II*. Se trata de una asignatura obligatoria de 6 créditos LRU, repartidos en tres créditos teóricos y tres créditos prácticos.

Las horas de docencia asignados en el plan de estudios como créditos prácticos se reparten al 50% entre ejercicios en el aula y el laboratorio. La asignatura se divide en cuatro bloques temáticos o unidades didácticas: *Ingeniería del Software orientada a objetos*, *Análisis orientado a objetos*, *Diseño orientado a objetos*, *Proceso orientado a objetos*.

Ingeniería del Software II		Horas Teor.	Horas Pract.
UD I	Ingeniería del Software orientada a objetos.	4	
Tema 1	Introducción.	2	
Tema 2	El Lenguaje Unificado de Modelado (UML)	2	
UD II	Análisis orientado a objetos.	18	12
Tema 3	Clases y objetos.	4	
Tema 4	Modelado del dominio. Diagrama de clases.	6	4
Tema 5	Modelado de los requisitos. Casos de uso.	2	2
Tema 6	Modelado del comportamiento. Diagramas de interacción.	3	3
Tema 7	Modelado avanzado del comportamiento.	3	3
UD III	Diseño orientado a objetos	6	3
Tema 8	Diseño orientado a objetos.	3	2
Tema 9	Diseño de la interfaz de usuario.	3	1
UD IV	Proceso orientado a objetos	2	
Tema 10	Métodos OO. El Proceso Unificado.	2	

	Laboratorio de Ingeniería del Software II		15
	Total	30	30

La primera unidad didáctica sirve como introducción al paradigma de desarrollo orientado a objetos. Se justifica el auge de la tecnología de objetos como una manera diferente de enfocar el desarrollo del software, se pueda hacer frente a la continua demanda social de software de mayor calidad, complejidad y sencillez de manejo. En la Orientación a Objetos el software se organiza como una colección de objetos discretos, los cuales contienen datos y comportamiento; manteniéndose este modelo a lo largo de todo el ciclo de vida, lo que permite una transición mucho más suave entre las diferentes fases de éste, a la vez que se facilita la utilización de modelos de ciclo de vida iterativos e incrementales. Por último, se sitúa en su contexto el Lenguaje Unificado de Modelado (UML) como estándar para las técnicas de modelado basadas en la orientación a objetos.

La segunda unidad didáctica se dedica a las actividades relacionadas con el análisis orientado a objetos. Comienza con el estudio del Modelo Objeto subyacente en UML, al tiempo que se profundiza en las características de los diagramas estructurales propuestos en este estándar. Se enfatiza la fase de obtención de requisitos como algo independiente del paradigma posterior de desarrollo, sea orientado a objetos o estructurado. Para la definición de los requisitos funcionales, se utilizan los casos de uso. La unidad didáctica se completa con el estudio de los diagramas de interacción y diagramas de estados y su aplicación a las actividades del análisis orientado a objetos.

La tercera unidad didáctica se centra en el diseño orientado a objetos. En la fase de análisis lo fundamental es determinar qué debe hacerse, sin que sea necesario, ni conveniente, comprometerse sobre la forma en que se hará. Sin embargo, durante el diseño es necesario tomar decisiones sobre la forma en que se resolverá el problema. El diseño del sistema y diseño de los objetos representan las fases de resolución de este problema. Así, aunque los diagramas utilizados son los mismos, se pasa de una representación del mundo real, propia del modelo de análisis, a una orientación propia del mundo de los ordenadores requerida para llevar a cabo una implementación práctica. Se incluye también en esta unidad un tema dedicado al diseño de la interfaz de usuario.

La última unidad didáctica se dedica a presentar las características específicas de los métodos actuales de desarrollo, en particular el Proceso Unificado.

El programa de prácticas para las dos asignaturas de primer ciclo, *Ingeniería del Software I* e *Ingeniería del Software II*, tiene una estructura idéntica diferenciándose únicamente en el tipo de herramienta CASE a utilizar. Está orientado a satisfacer los objetivos prácticos que se ajustan a las características de estas asignaturas, además de promover las habilidades personales en los alumnos. Como elemento fundamental, se propone la realización de un pequeño proyecto guiado por el profesor en las clases de laboratorio. La elaboración del trabajo se efectuará en grupos de dos o tres alumnos de cualquiera de las dos asignaturas.

El caso de estudio será de libre elección y requiere la aprobación previa de los profesores para asegurar que, por su envergadura, pueda concluirse durante el curso. Este sistema ha sido el utilizado en los últimos años académicos en las asignaturas de Ingeniería del Software de Ingeniería Técnica en Informática (tanto de Gestión como de Sistemas) con buenos resultados.

Para la superación de los objetivos de la asignatura, se precisa la calificación de apto en el cómputo global del proyecto de laboratorio, de entrega obligatoria, y una prueba escrita, similar en sus objetivos a la planteada en *Ingeniería del Software I*. Del mismo modo, la calificación final se obtendrá ponderando el proyecto un 20% y el ejercicio escrito el otro 80%.

5.2. Segundo ciclo/Master

La asignatura de *Ingeniería del Software I* está enfocada al estudio sistemático de las técnicas de especificación de requisitos y diseño de sistemas software. La orientación seguida representa una profundización en el paradigma de la orientación a objetos y una introducción a los métodos formales. Los objetivos de la asignatura son:

- Conocer y comprender los principios y técnicas que permiten una aproximación sistemática y cuantificable a la Ingeniería del Software.
- Ser capaz de evaluar los métodos, técnicas y herramientas de la Ingeniería del Software, y su aplicación a problemas concretos.
- Conocer los estándares internacionales en Ingeniería del Software y, especialmente, de Ingeniería de Requisitos.
- Ser capaz de producir una documentación correcta y completa de especificación de requisitos de software de acuerdo con los estándares y prácticas comunes.
- Conocer de forma detallada las técnicas y herramientas de modelado orientado a objetos.
- Dominar la semántica y sintaxis del lenguaje estándar de modelado UML.
- Conocer y aplicar el lenguaje de restricciones propuesto por UML (OCL).
- Conocer las distintas aproximaciones formales al desarrollo del software.
- Discutir el papel de las técnicas de especificación formal en el contexto del desarrollo del software, explicar sus beneficios potenciales, así como sus puntos débiles.
- Aplicar algunas técnicas de especificación formal de amplia aceptación.
- Ser capaz de realizar a lo largo del curso el análisis de un supuesto real, aplicando las técnicas mostradas en la parte teórica de la asignatura.

En la siguiente tabla se describe de forma esquemática la organización temporal de la asignatura *Ingeniería del Software I* de cuarto curso de la titulación de Ingeniero en Informática. Se trata de una asignatura obligatoria de 9 créditos LRU (90 horas), repartidos en seis créditos teóricos y tres créditos prácticos.

Las horas de docencia asignados en el plan de estudios como créditos prácticos se imparten totalmente en el laboratorio. La asignatura se divide en tres unidades didácticas: *Introducción*, *Especificación orientada a objetos*, *Métodos Formales en Ingeniería del Software*.

La primera unidad didáctica sirve como introducción a la Ingeniería del Software, en general, y a la Ingeniería de Requisitos, en particular. Comienza con un repaso a los factores de calidad del software y un estudio de los diferentes modelos de ciclos de vida. Se diferencia el análisis propiamente dicho de la especificación de requisitos software y se discuten las distintas formas de especificación.

La segunda unidad didáctica se dedica a la especificación orientada a objetos. Comienza con una revisión del modelo básico estructural y de comportamiento de UML, que los alumnos en su mayoría ya conocen por su formación previa. Los siguientes temas se dedican a estudiar los aspectos avanzados de UML. El modelado estructural se completa con la exposición de las posibilidades del Lenguaje de Restricciones de Objetos u *Object Constraint Language* (OCL).

Por su parte, el modelado de comportamiento avanzado incluye el tratamiento en profundidad de conceptos tales como eventos, señales, máquinas de estados, procesos, hilos y objetos activos. La aplicación de estas posibilidades avanzadas de UML se concreta en el estudio de sistemas complejos, como son los sistemas distribuidos y de tiempo real. Finalmente, la unidad didáctica se completa con un tema dedicado por completo a la semántica de UML. El metamodelo de UML se estudia con detalle como una forma más rigurosa de enfocar el estudio de UML.

Ingeniería del Software I		Horas Teor.	Horas Pract.
UD I	Introducción	8	
Tema 1	Ingeniería del Software e Ingeniería de Requisitos.	4	
Tema 2	Estándares de desarrollo y de documentación del software.	4	
UD II	Especificación orientada a objetos	36	
Tema 3	Modelo básicos estructural y de comportamiento.	4	
Tema 4	Modelo estructural avanzado: restricciones y OCL.	8	
Tema 5	Modelo avanzado de comportamiento.	4	
Tema 6	Aplicación a sistemas distribuidos.	8	
Tema 7	Aplicación a sistemas de tiempo real.	8	
Tema 8	Semántica de UML. Estructura del metamodelo.	4	
UD III	Métodos Formales en Ingeniería del Software	16	
Tema 9	Métodos formales. Introducción.	2	
Tema 10	Introducción a las especificaciones algebraicas.	4	
Tema 11	Otras técnicas de especificación formal.	4	
Tema 12	Verificación de especificaciones formales.	6	
	Laboratorio de Ingeniería del Software I		30
	Total	60	30

La tercera unidad didáctica se dedica a presentar las características específicas de las técnicas formales de especificación. No se pretende realizar un estudio exhaustivo de estas técnicas, sino únicamente presentar una introducción a este tipo de métodos indicando en qué situaciones o en qué ámbitos de la ingeniería de requisitos está justificado su uso.

En la siguiente tabla se describe de forma esquemática la organización temporal de la asignatura *Ingeniería del Software II* de quinto curso de la titulación de Ingeniero en Informática. Se trata de una asignatura obligatoria de 9 créditos (90 horas), repartidos en seis créditos teóricos y tres créditos prácticos.

Las horas de docencia asignados en el plan de estudios como créditos prácticos se imparten totalmente en el laboratorio. La asignatura se divide en tres unidades didácticas: *Gestión de proyectos software*, *Diseño y evolución del software*, *Métricas del software*.

La asignatura enfoca el software desde dos ópticas complementarias: producto y proceso. El primero tiene que ver con el software como tal, su diseño y sus propiedades. El segundo aspecto aborda el proceso de planificación, desarrollo y garantía de calidad del software. Por razones prácticas se trata en primer lugar este último aspecto, de modo que los alumnos puedan hacer estimaciones o planificación de un pequeño proyecto en el laboratorio, antes de desarrollarlo. El resto de la asignatura tiene como objetivos el estudio del diseño del producto software y la medición del producto obtenido.

Ingeniería del Software II		Horas Teor.	Horas Pract.
UD I	Gestión de proyectos software	24	
Tema 1	Proceso Software. Administración de proyectos.	4	
Tema 2	Evaluación de soluciones y estimación de riesgos.	4	
Tema 3	Planificación de proyectos.	8	
Tema 4	Garantía de calidad del software.	4	
Tema 5	Gestión de la configuración del software	4	
UD II	Diseño y evolución del software	16	
Tema 6	Conceptos generales. Aspectos del diseño.	2	
Tema 7	Patrones de diseño y estilos arquitectónicos.	6	
Tema 8	Lenguajes de descripción arquitectónica.	4	
Tema 9	Evolución del software.	4	
UD III	Métricas del software	16	
Tema 10	Introducción a las métricas del software.	4	
Tema 11	Métricas de producto.	8	
Tema 12	Métricas de proceso y métricas de recursos.	4	
Laboratorio de Ingeniería del Software II			30
Total		60	30

6. Metodología y estrategias de aprendizaje

6.1. Metodología docente

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

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

- Convertirse en el responsable de su aprendizaje, debiendo gestionar y controlar la forma en que se produce.
- Desarrollar un espíritu crítico y una actitud abierta ante los cambios científico-técnicos de su especialidad, con curiosidad intelectual y rigor científico.
- Fomentar actitudes y adquirir técnicas para trabajar en equipo con eficacia.
- Basar en principios deontológicos su futura actividad profesional.

- Valorar el sentido humanístico de la ciencia y de la técnica como resultado del esfuerzo de generaciones anteriores, que debe servir para resolver los problemas del hombre actual y futuro.
- Estimular el perfeccionamiento profesional y la formación continua a lo largo de la vida.

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

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

- *clases de teoría*

Las clases de teoría se llevan a cabo utilizando una variante de la clase magistral, en la cual el profesor se apoya en un retroproyector y en la pizarra para el desarrollo de los temas. La elección de esta forma de impartir clases viene forzada por dos factores fundamentales: clases de unos 100 alumnos por término medio, y abundancia de conceptos que se han de impartir. Los alumnos cuentan de antemano con las transparencias de los temas para que no tengan que tomar apuntes en el sentido clásico del término, y puedan prestar atención a las explicaciones, completando las transparencias con las notas que cada uno crea oportuno. Además, permite que el alumno que lo desee intervenga en cualquier momento para solventar una duda y no simplemente para pedir que se repita una frase.

En las transparencias se recogen únicamente los puntos fundamentales que se van a desarrollar, más esquemas, gráficos o definiciones de términos. Con ello se consigue que la clase no sea una mera lectura de las transparencias y obliga al alumno a prestar atención a las explicaciones y a completar el material con la bibliografía de consulta básica o las lecturas complementarias que se recomiendan en cada tema.

- *clases prácticas en aula*

En las clases dedicadas a la resolución de ejercicios propuestos o laboratorios, se puede actuar con mucha más flexibilidad. En función de la complejidad del problema abordado, las técnicas deben variar:

- En el caso de los ejercicios introductorios se utilizará la pizarra como el mejor instrumento para detallar todos los pasos de un proceso de análisis o diseño.
- Para la resolución de casos de estudio “de juguete”, de una mayor complejidad, una combinación de pizarra y transparencias permite mostrar una solución completa y al mismo tiempo discutir los detalles en la pizarra.
- Para casos de estudio más realistas, tanto la pizarra como las transparencias son insuficientes. En estos casos, la utilización combinada de ordenador y retroproyector permite la utilización en clase de las herramientas CASE o entornos de desarrollo

necesarios. En estos casos, la utilización de la página Web de la asignatura es fundamental para proporcionar al alumno los ficheros de trabajo.

- En las clases de laboratorio, ordenador y retroproyector son también insustituibles a la hora de presentar las herramientas que deben utilizar los alumnos para realizar sus prácticas.

- *clases prácticas en laboratorio (talleres de desarrollo)*

El programa de prácticas para las dos asignaturas de primer ciclo, *Ingeniería del Software I* e *Ingeniería del Software II*, tiene una estructura idéntica diferenciándose únicamente en el tipo de herramienta CASE a utilizar. El formato es el de trabajo en un taller de desarrollo, tutelado por el profesor. Está orientado a satisfacer los objetivos prácticos que se ajustan a las características de estas asignaturas, además de promover las habilidades personales en los alumnos. Los objetivos de este trabajo son:

- Poner en práctica y reforzar los conocimientos sobre procedimientos y herramientas que se imparten a lo largo del curso.
- Mostrar al estudiante alguno de los problemas que se presentan en el desarrollo de proyectos reales.
- Desarrollar la capacidad de comunicación del estudiante y la de trabajo en equipo.

Para el desarrollo de las prácticas se dispondrá de un laboratorio de herramientas CASE. El programa de prácticas de laboratorio de las asignaturas se desarrolla con un esquema similar sobre una asignación de 15 horas presenciales y trabajo adicional del grupo. Estas 15 horas se distribuyen en 3 horas dedicadas a la presentación de las herramientas y 6 bloques de 2 horas semanales a lo largo del cuatrimestre, destinadas a la tutela del proyecto elegido. En cualquier caso, gracias al alto grado de disponibilidad del laboratorio, los bloques de prácticas se pueden ajustar con gran flexibilidad, de modo que sigan el ritmo de las clases teóricas. Las horas especificadas se refieren exclusivamente a estancia tutelada en el laboratorio ya que los alumnos deberán completar dicha actividad con trabajo personal.

Para su desarrollo se dispone de un laboratorio de herramientas CASE. Las herramientas utilizadas durante los últimos años incluyen **Rational ROSE**, **Together** y **Easy CASE**. Además, para facilitar el trabajo del alumno, éste puede descargar una versión para uso personal, de la herramienta **Rational ROSE** o **Together**, versión Windows o Linux, y sus manuales.

- *tutorías.*

El alumnado tiene a su disposición seis horas de tutorías a la semana en las que puede consultar cualquier duda relacionada con los contenidos, organización y planificación de la asignatura. Las tutorías pueden ser individualizadas, pero se admite tutorías grupales para resolver problemas relacionados con las actividades a realizar en grupo. Un alumno individualmente puede entregar ejercicios resueltos por él, cuyos enunciados debe encontrar en la bibliografía (sin que estén resueltos). Éstos serán discutidos con el profesor en horas de tutoría.

- *página web.*

Se convierte en el vehículo de comunicación y registro de información de la asignatura.

6.2. Estrategias de aprendizaje

Se detalla las actuaciones concretas a realizar para la aplicación y aprovechamiento de la metodología propuesta. Se estructura en las siguientes fases:

6.2.1. Recopilación de la documentación de la asignatura

- Los alumnos disponen en la página web de la asignatura de toda la información y normativa relacionada con la asignatura: temario, criterios de evaluación, bibliografía, apuntes, enunciados de prácticas, trabajos, enlaces de interés, lecturas complementarias, avisos...
- El profesor mantiene actualizada la información de esta página para que se convierta en un vehículo de comunicación con los alumnos.
- Los alumnos deben conocer y manejar con fluidez toda la información y la documentación que se integra en la página de la asignatura.
- Asiduamente, el alumno debe consultar la página para estar actualizado.

6.2.2. Planificación de las clases teóricas

- Las clases teóricas se dedicarán a la presentación de contenidos y a la discusión sobre las dudas que surjan durante las exposiciones.
- Las clases se desarrollarán según el siguiente esquema:
 1. El profesor presenta los objetivos a conseguir y el contexto en el que se va a desarrollar la clase.
 2. Se presentan los conceptos.
 3. Se aclaran las dudas cuando éstas surjan. El profesor puede incentivar el debate con preguntas para hacer la clase más participativa.
 4. El profesor termina con un resumen de los principales conceptos tratados en la clase y puede introducir la siguiente clase.
- El alumno debe haber realizado una lectura previa de los contenidos que se van a tratar en la clase, máxime cuando no todos los contenidos se desarrollan en las clases magistrales.
- Una vez terminada la clase magistral, se debe estudiar de forma autónoma su contenido y en caso de no entender algo intentar aclararlo utilizando la bibliografía recomendada o cualquiera de los materiales adicionales recomendados en la asignatura.

6.2.3. Planificación de las clases prácticas

- El profesor publicará en la página de la asignatura los enunciados de los problemas y anunciará con suficiente antelación las fechas para su resolución en el aula.
- En el caso de la práctica obligatoria el grupo de trabajo debe leer entregar la propuesta inicial durante el primer mes y, una vez obtenido el visto bueno del profesor, ajustarse al calendario de entregas (típicamente requisitos, análisis y diseño).

- Las prácticas obligatorias se defenderán en grupo.

6.2.4. Evaluación

- Se plantea una forma de evaluación continua en la parte práctica, complementada con una prueba final

7. Planificación del tiempo y del esfuerzo

Para hacer su estudio en ECTS se ha tomado tomando la equivalencia de crédito LRU a crédito ECTS, con implicación de cambio de metodología docente. Por temas de organización y calendario, se ha optado por partir del escenario que asigna 25 horas de trabajo al ECTS, incluyendo horas presenciales, horas de estudio personal o trabajo en grupo y tutorías.

En las tablas siguientes se presenta la organización del esfuerzo del alumno para cubrir los créditos ECTS asociados a la asignaturas de primer ciclo, respetando la asignación horaria reflejada en el apartado 5, temario.

	Técnica	Actividad	Horas equivalentes de clase	Factor de trabajo del alumno†	Horas de trabajo personal del alumno	Horas totales (A+C)	ECTS (D ÷ 25)
Teoría	Seguimiento de la parte teórica de las unidades didácticas	Asimila contenidos. Se plantea dudas que planteará a los profesores en las tutorías	44	1,5	66	110	4,4
Ejercicios de apoyo	Ejercicios de repaso y afianzamiento de lo aprendido.	Realiza, resuelve problemas, tests...	15	1,5	22,5	37,5	1,5
Práctica obligatoria	Práctica en grupo de obligada realización	Realiza una ERS completada con elementos de diseño	10	2	30	40	1,6
Otras actividades	Tutorías personalizadas y grupales	Recibe orientación personalizada	-	-	1	1	0,04
	Búsquedas en la red, participación en foros especializados...	Busca elementos para completar los contenidos	-	-	1	1	0,04
TOTAL			69		120,5	189,5	7,58

Tabla 1. Ingeniería del Software I

† Número de horas dedicadas por el alumno al trabajo personal (organización de apuntes, estudio, documentación...) por cada hora de clase.

	Técnica	Actividad	Horas equivalentes de clase	Factor de trabajo del alumno†	Horas de trabajo personal del alumno	Horas totales (A+C)	ECTS (D ÷ 25)
Teoría	Seguimiento de la parte teórica de las unidades didácticas	Asimila contenidos. Se plantea dudas que planteará a los profesores en las tutorías	29	1,5	43,5	72,5	2,9
Ejercicios de apoyo	Ejercicios de repaso y afianzamiento de lo aprendido.	Realiza, resuelve problemas, tests...	15	1,5	22,5	37,5	1,5
Práctica obligatoria	Práctica en grupo de obligada realización	Realiza una ERS completada con elementos de diseño	10	2	30	40	1,6
Otras actividades	Tutorías personalizadas y grupales	Recibe orientación personalizada	-	-	1	1	0,04
	Búsquedas en la red, participación en foros especializados...	Busca elementos para completar los contenidos	-	-	1	1	0,04
TOTAL			54		98	152	6,08

Tabla 2. Ingeniería del Software II

8. Bibliografía

8.1. Bibliografía básica

Booch, G., Rumbaugh, J., Jacobson, I. (1999) *El Lenguaje Unificado de Modelado*. Addison Wesley.

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

Larman, C. (2003) *UML y Patrones*. 2ª Edición. Prentice-Hall.

Pfleeger, S. L. (2002) *Ingeniería del Software*. Teoría y Práctica. Prentice Hall.

Piattini, M. G., Calvo-Manzano, J. A., Cervera, J., Fernández, L. (2004) *Análisis y Diseño de Aplicaciones Informáticas de Gestión. Una perspectiva de Ingeniería del Software*. Ra-ma.

Pressman, R. S. (2005) *Ingeniería del Software: Un Enfoque Práctico*. 6ª Edición. McGraw-Hill.

† Número de horas dedicadas por el alumno al trabajo personal (organización de apuntes, estudio, documentación...) por cada hora de clase.

Rumbaugh, J., Jacobson, I., Booch, G. (2000) *El Lenguaje Unificado de Modelado. Manual de Referencia*. Addison-Wesley.

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

Yourdon, E. (1993) *Análisis Estructurado Moderno*. Prentice-Hall Hispanoamericana.

8.2. Bibliografía complementaria

Gamma, E., Helm, R., Johnson, R., Vlissides, J. (2003) *Patrones de Diseño*. Addison-Wesley.

Larman, C. (2005) *Applying UML and Patterns. An Introduction to Object-Oriented Analysis and Design and Iterative Development*. 3rd Edition. Prentice Hall.

Meyer, B. (1999) *Construcción de Software Orientado a Objetos*. 2ª Edición. Prentice Hall.

Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., Lorensen, W. (1998) *Modelado y Diseño Orientados a Objetos. Metodología OMT*. Prentice Hall, 2ª reimpression.

9. Evaluación de los procesos y resultados de aprendizaje

9.1. Sistema de evaluación

La evaluación de esta asignatura se divide en dos partes diferenciadas:

- **Parte de Teoría** (70% de la nota final).
 - Un examen final que consta de dos partes que hay que aprobar independientemente:
 - Un test.
 - Un conjunto de supuestos teórico/prácticos.
- **Parte Práctica** (30% de la nota final).
 - Práctica realizada en grupos de trabajo.
 - Se realizará una defensa de dicho trabajo en forma de evaluación continua. Puede haber una presentación final.
 - La parte práctica se guardará durante el curso académico actual.

9.2. Criterios de evaluación

La calificación se hará conforme a las siguientes pautas:

Matrícula de honor

- La nota final es superior o igual a 9,5 puntos
- El conocimiento y la comprensión de la materia se extienden más allá del trabajo cubierto por el programa.
- La comprensión conceptual es sobresaliente.
- Los problemas relacionados con la asignatura son resueltos con soltura y se es capaz de razonar alternativas de solución estableciendo comparativas entre ellas.

Sobresaliente

- La nota final es superior o igual a 9 puntos
- El conocimiento y la comprensión de la materia son muy satisfactorios.
- La comprensión conceptual es sobresaliente.
- Los problemas relacionados con la asignatura son resueltos con soltura y se es capaz de razonar alternativas de solución.

Notable

- La nota final es superior o igual a 7 puntos
- El conocimiento y la comprensión de la materia son satisfactorios.
- La comprensión conceptual es notable.
- Los problemas relacionados con la asignatura son resueltos con soltura.

Aprobado

- La nota final es superior o igual a 5 puntos
- El conocimiento y la comprensión de la materia son básicos.
- La comprensión conceptual es suficiente.
- Los problemas relacionados con la asignatura son resueltos con éxito razonable.

Suspenso

- La nota final es inferior a 5 puntos
- El conocimiento y la comprensión de la materia no son aceptables.
- La comprensión conceptual es insuficiente.
- Los problemas relacionados con la asignatura no se resuelven de forma adecuada.

Apéndice II: Artículos publicados

- **García Peñalvo, F. J.** “*Diseño Curricular de una Asignatura en el Contexto del Espacio Europeo de Educación Superior*”. En las *Actas de las I Jornadas de Innovación Educativa de la Escuela Politécnica Superior de Zamora. Las Enseñanzas Técnicas ante el Reto del Espacio Europeo de Educación Superior* (Zamora, 20-22 de junio de 2006). J. L. Pérez Iglesias, M^a L. Pérez Delgado, M. P. Rubio Cavero, J. C. Matos Franco, J. Calvo Gallego (Eds.). Versión CD-ROM. ISBN 84-689-9304-2. Junio, 2006.
- **Calvo Gallego, J., García Peñalvo, F. J.** “*Guía Docente de la Asignatura Ingeniería del Software II dentro del Marco del EEES*”. En las *Actas de las I Jornadas de Innovación Educativa de la Escuela Politécnica Superior de Zamora. Las Enseñanzas Técnicas ante el Reto del Espacio Europeo de Educación Superior* (Zamora, 20-22 de junio de 2006). J. L. Pérez Iglesias, M^a L. Pérez Delgado, M. P. Rubio Cavero, J. C. Matos Franco, J. Calvo Gallego (Eds.). Versión CD-ROM. ISBN 84-689-9304-2. Junio, 2006.
- **Pérez, J. L., Matos, J. C., García, F. J.** “*La Asignatura de Hipermedia en el Marco del E.E.E.S.*”. En las *Actas de las I Jornadas de Innovación Educativa de la Escuela Politécnica Superior de Zamora. Las Enseñanzas Técnicas ante el Reto del Espacio Europeo de Educación Superior* (Zamora, 20-22 de junio de 2006). J. L. Pérez Iglesias, M^a L. Pérez Delgado, M. P. Rubio Cavero, J. C. Matos Franco, J. Calvo Gallego (Eds.). Versión CD-ROM. ISBN 84-689-9304-2. Junio, 2006.
- **González, A. B., Rodríguez, M. J., Olmos S.** “*Aprendizaje activo en Ingeniería Técnica Informática, esp. Gestión. Sistemas Informáticos*”. En las *Actas de las I Jornadas de Innovación Educativa de la Escuela Politécnica Superior de Zamora. Las Enseñanzas Técnicas ante el Reto del Espacio Europeo de Educación Superior* (Zamora, 20-22 de junio de 2006). J. L. Pérez Iglesias, M^a L. Pérez Delgado, M. P. Rubio Cavero, J. C. Matos Franco, J. Calvo Gallego (Eds.). Versión CD-ROM. ISBN 84-689-9304-2. Junio, 2006.
- **Martín, D., San Juan, Y., Vizán, R., González, A. B.** “*Aprender Aprobar*”. En las *Actas de las I Jornadas de Innovación Educativa de la Escuela Politécnica Superior de Zamora. Las Enseñanzas Técnicas ante el Reto del Espacio Europeo de Educación Superior* (Zamora, 20-22 de junio de 2006). J. L. Pérez Iglesias, M^a L. Pérez Delgado, M. P. Rubio Cavero, J. C. Matos Franco, J. Calvo Gallego (Eds.). Versión CD-ROM. ISBN 84-689-9304-2. Junio, 2006.
- **García Peñalvo, F. J., Alonso Romero, L., Cardenoso Payo, V., Pardo Aguilar, C., Fernández Díaz, R. Á.** “*Perspectiva Histórica de los Estudios de Ingenierías Informáticas en las Universidades Públicas de Castilla y León*”. En F. J. García Peñalvo (Ed.) *Los Estudios de Ingeniería Informática en el Espacio Europeo de Educación Superior*.

Contexto y Realidad en la Comunidad Autónoma de Castilla y León. Páginas 1-7. Ediciones Universidad de Salamanca. Colección Aquilafuente, Nº 101. ISBN 84-7800-436-X. Agosto, 2006.

- **García Peñalvo, F. J.** “*Pensando en ECTS. Un Caso Práctico para la Asignatura de Ingeniería del Software*”. En F. J. García Peñalvo (Ed.) *Los Estudios de Ingeniería Informática en el Espacio Europeo de Educación Superior. Contexto y Realidad en la Comunidad Autónoma de Castilla y León.* Páginas 139-156. Ediciones Universidad de Salamanca. Colección Aquilafuente, Nº 101. ISBN 84-7800-436-X. Agosto, 2006.