



ASIGNATURA: Análisis y Diseño de Sistemas.

Curso: 2004/2005

Carácter: Obligatoria **Temporalidad:** 1^{er} Cuatrimestre **Créditos:** 6 (4,5T+1,5P)
Profesor: Andrés Santiago Martín **Despacho:** Despacho 10
Web: <http://cum.unex.es/profesores/asanmar/default.htm> **E-mail:** asanmar@unex.es

NORMAS GENERALES:

- Las convocatorias de los exámenes serán fijadas por la Subdirección Académica del Centro.
- Todo alumno deberá entregar obligatoriamente una ficha al profesor de la asignatura.
- Los alumnos deberán formar grupos de 4 personas para la realización de la práctica.
- La composición de los grupos de prácticas deben de entregarse al profesor no más tarde del día 15 de Octubre de 2004.
- En los trabajos prácticos de la asignatura el profesor establecerá, en su momento, la fecha límite de entrega de cada uno de dichos trabajos.

CRITERIOS DE EVALUACIÓN:

- **Parte de Teoría:**
 - Habrá un examen que constará de dos partes: una parte de teoría y otra parte de problemas. La nota de este examen representará el 60% de la nota final.
 - Para poder superar el examen, en cada parte, teoría y problemas, deberá obtener el alumno, al menos, el 30 % de la nota máxima que puede obtenerse en cada parte.
- **Parte Práctica:**
 - Desarrollo de un proyecto siguiendo una metodología estructurada. La nota total de la práctica representará el 40% de la nota final.
 - Cualquier sospecha de copia sobre una práctica entregada o parte de la misma, implicará inexorablemente suspender la práctica completa.
- **Trabajos voluntarios presentados.**
 - Dichos trabajos versarán sobre temas relacionados directamente con la asignatura, y necesitarán de la aprobación previa del profesor.
 - Estos trabajos, tras su evaluación por el profesor, podrán subir de 0,5 a 1,5 puntos que se sumarán a la nota conseguida en los apartados anteriores, siempre que en los mismos se obtenga la calificación mínima exigida.
 - La entrega de trabajos sólo se aplicará a la convocatoria de Febrero.
- La nota final será igual a:
 - **Si** (teoría $\geq 4,5$) y (práctica $\geq 4,5$) y $([(teoría * 0,6) + (práctica * 0,4)] \geq 5)$
 - nota final = (teoría * 0,6) + (práctica * 0,4) + nota trabajos
 - **Si no**
 - nota final = Suspenso
 - **Fin si**
- En todo caso, sólo se aprobará si la nota final es igual o superior a 5.
- Si un alumno tiene una parte aprobada, nota mayor o igual a 5, y tiene la otra parte suspensa (no compensable), con nota menor que 4,5, la nota final obtenida será la de la parte suspensa.
- Tanto la nota del examen teórico como la nota de la práctica podrán ser guardadas hasta la convocatoria de septiembre del presente curso, siempre que sea superior a 5 puntos.
- Si un alumno presenta un trabajo voluntario en la convocatoria de Febrero, y en cambio no aprueba la asignatura, la nota de dicho trabajo se guarda hasta la convocatoria de septiembre del presente curso.



OBJETIVOS GENERALES:

- Descripción de las actividades técnicas y de ingeniería que se llevan a cabo en el ciclo de vida de un producto software.
- Descripción de los problemas, principios, métodos y tecnologías asociadas con la Ingeniería del Software.
- Presentación de la importancia de los requisitos en el ciclo de vida del Software.
- Introducción a las técnicas básicas de elicitación, documentación y especificación de los requisitos de un sistema software.
- Introducción a los métodos de análisis/diseño estructurado.
- Estudio y comprensión de los fundamentos del diseño de sistemas software.
- Aplicar de forma práctica de los conceptos teóricos sobre el desarrollo estructurado.
- Realización de un proyecto en grupo, aplicando los principios introducidos en la parte teórica de la asignatura.

METODOLOGÍA:

- En las tres clases de teoría semanales se expondrán los conceptos teóricos básicos sobre gestión y desarrollo de proyectos informáticos, y se introducen en mayor detalle técnicas y metodologías estructuradas de desarrollo del software.
- Cuando en el tema que se esté impartiendo en teoría haya problemas a resolver, de las tres horas de teoría semanales, una hora semanal se dedicará a la resolución de problemas. Para ello, se facilitará a los alumnos una serie de problemas sobre dicho tema, y algunos de los problemas de dicha se resolverán en dicha hora.
- Así mismo, están las horas de tutorías en las que los alumnos pueden consultar con el profesor la resolución de cualquier duda planteada sobre cualquier aspecto de la asignatura.
- En las clases teóricas se utilizan medios audiovisuales de divulgación informática.
- Las clases prácticas se destinan a introducir al alumno en el manejo de una herramienta CASE y a utilizar dicha herramienta para el desarrollo de software, utilizando metodologías estructuradas.
- En la página Web del profesor y en la copistería del centro se encuentra a disposición de los alumnos, los horarios de tutorías, temarios de las asignaturas, criterios de evaluación así como todo el material didáctico utilizado en la asignatura.



PROGRAMA TEÓRICO:

TEMA 1.- SISTEMAS DE INFORMACIÓN.

Conceptos básicos.
Elementos de un Sistema de Información.
Estructura de un Sistema de Información.
Tipos de sistemas de información automatizados.
Principios generales de sistemas.
Ingeniería de sistemas.
Adquisición de sistemas.

TEMA 2.- INTRODUCCIÓN A LA INGENIERÍA DEL SOFTWARE.

El software.
Conceptos básicos de la Ingeniería del Software.
El proceso software.
Modelos de proceso software.
Herramientas CASE.

TEMA 3.- INTRODUCCIÓN A LA INGENIERÍA DE REQUISITOS.

Introducción.
Ingeniería de Requisitos.
Requisitos.
Especificación de Requisitos del Software.
MDB. Una metodología de elicitación de requisitos.
Vista de casos de uso.
Modelo de requisitos. Modelo de casos de uso.
Caso de estudio.

TEMA 4.- ANÁLISIS ESTRUCTURADO.

Introducción.
Modelado funcional.
Modelado de la información.
Modelo de comportamiento.
Balanceo de modelos.
Método de análisis de Yourdon.

TEMA 5.- PRINCIPIOS DE DISEÑO DEL SOFTWARE.

Introducción.
Proceso de diseño del software.
Conceptos básicos del diseño.
Cohesión.
Acoplamiento.
Otros principios y conceptos de diseño.
Métodos de diseño.

TEMA 6.- DISEÑO ESTRUCTURADO.

Introducción.
Diagrama de Estructura de Cuadros.
Diseño arquitectónico.
Diseño detallado.
Diseño de los datos.
Diseño adicional.
Guías adicionales de diseño.
Guías de un buen diseño.
Principales errores en el diseño.

TEMA 7.- MODELOS DE PROCESO DE SOFTWARE.

Modelos lineales o secuenciales.
Modelos basados en prototipos.
Modelos basados en métodos formales.
Modelos evolutivos.
Modelos basados en reutilización.

TEMA 8.- METODOLOGÍAS DE DESARROLLO DEL SOFTWARE.

Introducción. Definiciones.
Objetivos de las metodologías.
Características deseables de una metodología.
Clasificación de las metodologías.

TEMA 9.- PRUEBAS DEL SOFTWARE.

Objetivos y límites de las pruebas.
Tipos de pruebas.
Análisis y seguimiento de errores.
Documentación y diseño de las pruebas.
Planificación de las pruebas. Hitos y tareas.
Depuración.

PROGRAMA PRÁCTICO:

- La práctica consistirá en una serie de trabajos prácticos relacionados con el contenido teórico de la asignatura, y en el desarrollo de un proyecto de desarrollo de software siguiendo una metodología estructurada.
- Para llevar a cabo las prácticas se utilizará la herramienta CASE, **EasyCASE**, y para la actividad de elicitación de requisitos se utilizará la herramienta CASE, **REM versión 1.2.2** que es una herramienta experimental gratuita de Gestión de Requisitos diseñada para soportar la fase de Ingeniería de Requisitos de un proyecto de desarrollo software, definida en la Tesis Doctoral “*Un Entorno Metodológico de Ingeniería de Requisitos para Sistemas de Información*”, presentada por Amador Durán en la Universidad de Sevilla. Dicha herramienta se puede bajar de:

http://www.lsi.us.es/descargas/descarga_programas.php?id=3

BIBLIOGRAFÍA:

• Referencias principales.

1. *Pfleeger, S. L.*,
"Ingeniería del Software. Teoría y Práctica",
Prentice Hall, 2002.
2. *Pressman, Roger*,
"Ingeniería del Software. Un enfoque práctico", 5ª edición,
McGraw-Hill, 2002.
3. *Sommerville, Y.*,
"Ingeniería de software", 6ª edición,
Addison Wesley, 2002.
4. *Yourdon, E.*,
"Análisis estructurado moderno",
Prentice Hall, 1993.

• Otras referencias.

1. *Booch, G., Rumbaugh, J., Jacobson, I.*,
"El Lenguaje Unificado de Modelado",
Addison Wesley, 1999.
2. *Jacobson, I., Booch, G., Rumbaugh, J.*,
"El Proceso Unificado de Desarrollo de Software",
Addison-Wesley, 2000
3. **Metodología de planificación y desarrollo de sistemas de información. Métrica Versión 3**,
Ministerio para las Administraciones Públicas 2001.
Enlace: <http://www.csi.map.es/csi/metrica3/index.html>
4. *Meyer, B.*,
"Construcción de Software Orientado a Objetos", 2ª Edición,
Prentice Hall, 1999.
5. *Piattini, M. G., Calvo-Manzano, J. A., Cervera, J., Fernández, L.*,
"Análisis y Diseño Detallado de Aplicaciones Informáticas de Gestión",
RaMa, 1996.
6. *Rumbaugh, J., Jacobson, I., Booch, G.*,
"El Lenguaje Unificado de Modelado. Manual de Referencia",
Addison-Wesley. 2000.

Otras fuentes.

- *Durán Toro, A., Bernárdez, B.*
"Metodología para la Análisis de Requisitos de Sistemas Software (versión 2.2)".
Universidad de Sevilla. Diciembre 2001.
- *Durán, Toro, A., Martín Díaz, O.*
"Metodología para la Elaboración de Requisitos de Sistemas Software (versión 3.0)".
Universidad de Sevilla. Abril 2003.
- *Durán, Toro, A., Bernárdez, B.*
"Metodología para la Elicitación de Requisitos de Sistemas Software (versión 2.3)".
Universidad de Sevilla. Abril 2002.

Además en cada tema se especificará las referencias bibliográficas que se hayan utilizado.