

GUÍA DOCENTE
EXPERIENCIA PILOTO DE APLICACIÓN DEL SISTEMA EUROPEO DE CRÉDITOS (ECTS)
UNIVERSIDADES ANDALUZAS

TITULACIÓN: INGENIERÍA INFORMÁTICA DE GESTIÓN

DATOS BÁSICOS DE LA ASIGNATURA

NOMBRE: INGENIERIA DEL SOFTWARE DE GESTION II

CÓDIGO: 916

AÑO DE PLAN DE ESTUDIO: 2004

TIPO (troncal/obligatoria/optativa): TRONCAL

Créditos totales (LRU/ECTS):

6

Créditos LRU/ECTS teóricos:

3

Créditos LRU/ECTS prácticos:

3

CURSO: 3º

CUATRIMESTRE(S): 1º

CICLO: 1º

EQUIPO DOCENTE

Responsable / Coordinador de la asignatura:

NOMBRE: NORBERTO DIAZ DIAZ

CENTRO/DEPARTAMENTO: ESCUELA POLITECNICA SUPERIOR/ DEPORTE E INFORMÁTICA

ÁREA: LENGUAJES Y SISTEMAS INFORMATICOS

CATEGORÍA: PROFESOR COLABORADOR

LA ASIGNATURA EN EL PROGRAMA FORMATIVO

1. DESCRIPTOR.

2. UBICACIÓN EN EL PROGRAMA FORMATIVO.

2.1. PRERREQUISITOS:

Es recomendable haber superado la asignatura de ISG1 y BD donde se imparten los conceptos básicos de diseño del software y se introduce al alumno nociones fundamentales del diseño de base de datos.

2.2. CONTEXTO DENTRO DE LA TITULACIÓN:

En las asignaturas de ISG1 y BD, los alumnos aprenden las nociones fundamentales del diseño software y de base de datos. En esta asignatura se introduce al alumno con técnicas para el análisis, gestión y mantenimiento de software. Además, se integran los conocimientos adquiridos en los primeros cursos.

2.3. RECOMENDACIONES:

Ninguna.

3. LA ASIGNATURA EN LA ADQUISICIÓN DE COMPETENCIAS.

3.1. COMPETENCIAS TRANSVERSALES/GENÉRICAS:

Conocimientos generales básicos.
Solidez en los conocimientos básicos de la profesión.
Habilidades elementales en informática.
Resolución de problemas.
Capacidad de aprender.
Habilidad para trabajar en grupo.
Capacidad de análisis y síntesis.
Capacidad de organizar y planificar.

3.2. COMPETENCIAS ESPECÍFICAS:

- **Cognitivas (Saber):** Enseñar al alumno el papel fundamental que desempeña el análisis dentro del proceso software. Comprender la importancia de identificar requisitos y realizar una gestión disminuya los riesgos de su desarrollo.
- **Procedimentales/Instrumentales (Saber hacer):** Enseñar al alumno las técnicas para el análisis de problemas mediante técnicas descriptivas de modelado. Usar y elegir correctamente un tipo de proceso de desarrollo software dependiendo de los requisitos a cumplir.
- **Actitudinales (Ser):** Fomentar la disciplina en la programación, siguiendo planteamientos que desarrollen la capacidad analítica del alumno para enfrentarse a problemas reales. Ser consciente de las ventajas de identificar requisitos software en etapas tempranas del ciclo de vida software y su utilidad en el proceso de desarrollo software.

4. OBJETIVOS.

El objetivo principal de la asignatura es que los alumnos aprendan a identificar e integrar las diferentes fases del proceso de desarrollo software, haciendo especial hincapié al proceso de análisis y a los documentos que acompañan en cada una de éstas.

DISTRIBUCIÓN DEL TRABAJO PRESENCIAL.

	Gran Grupo	Grupo de Docencia	Actividades dirigidas (seminarios)
Nº de grupos	1	3	4
Nº de horas	23	20	8
Nº de sesiones	15	10	4

5. METODOLOGÍA.

NÚMERO TOTAL DE HORAS DE TRABAJO DEL ALUMNO: 165

PRIMER SEMESTRE: 165 horas de trabajo

Nº de Horas:

- Enseñanzas básicas (Gran Grupo): **23**
- Enseñanzas prácticas y de desarrollo (Grupo de Docencia): **20**
- Actividades académicas dirigidas (Seminarios-Grupo de Trabajo): **8**
- Tutorías especializadas (presenciales o virtuales): **2**
 - A) Colectivas: 2
 - B) Individuales: ...
- Trabajo personal autónomo: **106**
 - A) Horas de estudio de enseñanzas básicas: 46
 - B) Horas de estudio-preparación de las enseñanzas prácticas y de desarrollo: 40
 - C) Horas de trabajo personal o en grupo derivadas de las actividades académicas dirigidas: 20
- Otras actividades (visitas, excursiones, etc.): ...
- Realización de pruebas de evaluación y/o exámenes: **6**
 - A) Prueba de evaluación y/o exámenes escritos: 6
 - B) Pruebas de evaluación y/o exámenes orales (control del Trabajo Personal): ...

SEGUNDO SEMESTRE: 0 horas de trabajo

6. TÉCNICAS DOCENTES. (Señale con una X las técnicas que va a utilizar en el desarrollo de su asignatura. Puede señalar más de una).

Sesiones académicas teóricas: X	Exposición y debate: X	Tutorías especializadas: X
Sesiones académicas prácticas: X	Visitas y excursiones:	Controles de lecturas obligatorias:

Otras (especificar):

DESARROLLO Y JUSTIFICACIÓN:

Todas las Actividades Prácticas y de Desarrollo así como las Actividades Académicas Dirigidas llevarán asociada una documentación que se proporcionará al alumno a través de la página Web de la asignatura dentro del aula virtual WebCT.

7. BLOQUES TEMÁTICOS. (Dividir el temario en grandes bloques temáticos; no hay número mínimo ni máximo).

- BLOQUE 1: INTRODUCCIÓN AL ANÁLISIS
- BLOQUE 2: DISEÑO ARQUITECTÓNICO DEL SOFTWARE
- BLOQUE 3: PROCESO UNIFICADO DE DESARROLLO
- BLOQUE 4: ANÁLISIS DE PROYECTOS SOFTWARE

8. BIBLIOGRAFÍA.

8.1 GENERAL:

1. Ingeniería del Software (Edición 7 traducida). Ian Somerville. Editorial Pearson, Addison Wesley.
2. UML Xtra-Light. How to Specify Your Software. Milan Kratochvil, Barry McGibbon. Cambridge University Press
3. El proceso Unificado de Desarrollo de Software (traducida). Ivar Jacobson, Grady Boock, James Rumbaugh. Editorial Addison Wesley
4. UML y Patrones (2ª edición traducido). Craig Larman. Editorial Pearson, Prentice Hall
5. Ingeniería del Software. Un enfoque práctico (Edición 6 Traducido). Roger S. Pressman. Editorial Mac Graw Hill. ISBN 970-10-573-3
6. Ingeniería del Software. Una perspectiva orientada a objetos. Eric J.Braude. Adaptado por Macario Polo Usaola. Editorial Ra-Ma. ISBN 970-15-0851-3

8.2 ESPECÍFICA: (con remisiones concretas en lo posible)

Tema 1: Introducción

- Ingeniería del Software (Edición 7 traducida). Ian Somerville. Editorial Pearson, Addison Wesley.
 - Ingeniería software y sus fases: Parte I, páginas 1-100
- UML Xtra-Light. How to Specify Your Software. Milan Kratochvil, Barry McGibbon. Cambridge University Press. Completo.
- UML y Patrones (2ª edición traducida). Craig Larman. Editorial Pearson, Prentice Hall
 - Patrones GRASP: Capítulo 16, páginas 201-230
- Ingeniería del Software. Una perspectiva orientada a objetos. Eric J.Braude. Adaptado por Macario Polo Usaola. Editorial Ra-Ma. ISBN 970-15-0851-3
 - Patrones de diseño: Capítulo 6, páginas 303-354

Tema 2: Diseño arquitectónico del software.

- Ingeniería del Software (Edición 7 traducida). Ian Somerville. Editorial Pearson, Addison Wesley.
 - Diseño arquitectónico: Parte 3, páginas 217-307
- Ingeniería del Software. Un enfoque práctico (Edición 6 Traducido). Roger S. Pressman. Editorial Mac Graw Hill. ISBN 970-10-573-3
 - Diseño Interfaz Usuario: Capítulo 12, páginas 350-381.
 - Diseño arquitectónico: Capítulo 10, páginas 275-314.
- Ingeniería del Software. Una perspectiva orientada a objetos. Eric J.Braude. Adaptado por Macario Polo Usaola. Editorial Ra-Ma. ISBN 970-15-0851-3
 - Arquitectura del software: Capítulo 5, páginas 247-302

- JPA 101: Java Persistence Explained. Christopher Maki. Editorial Lulu.com, 2006. ISBN 0976553414, 9780976553410

Tema 3: El Proceso Unificado de Desarrollo

- El proceso Unificado de Desarrollo de Software (traducida). Ivar Jacobson, Grady Boock, James Rumbaugh. Editorial Addison Wesley. Completo
- UML y Patrones (2ª edición traducido). Craig Larman. Editorial Pearson, Prentice Hall
 - Completo; el libro muestra los artefactos a del Proceso Unificado y evolución temporal. Además, de mostrar las relaciones entre los artefactos del Proceso Unificado
- Ingeniería del Software. Un enfoque práctico (Edición 6 Traducido). Roger S. Pressman. Editorial Mac Graw Hill. ISBN 970-10-573-3
 - Ingeniería de Requisitos: Capítulo 7, páginas 155-190
 - Análisis: Capítulo 8, páginas 191-244
 - Diseño: Capítulo 9, páginas 245-276
- Ingeniería del Software. Una perspectiva orientada a objetos. Eric J.Braude. Adaptado por Macario Polo Usaola. Editorial Ra-Ma. ISBN 970-15-0851-3
 - Análisis de requisitos: Capítulo 3-4, páginas 135-246

Tema 4: Análisis de proyectos software

- Ingeniería del Software (Edición 7 traducida). Ian Somerville. Editorial Pearson, Addison Wesley.
 - Gestión Personal: Parte 6, capítulo 25, páginas 543-560
 - Estimación de costes software: Parte 6, capítulo 26, páginas 561-586
- Ingeniería del Software. Un enfoque práctico (Edición 6 Traducido). Roger S. Pressman. Editorial Mac Graw Hill. ISBN 970-10-573-3
 - Estimación para proyectos de software: Capítulo 23, páginas 690-723
- Ingeniería del Software. Una perspectiva orientada a objetos. Eric J.Braude. Adaptado por Macario Polo Usaola. Editorial Ra-Ma. ISBN 970-15-0851-3
 - Organización del personal: Capítulo 2, Parte 1, páginas 81-84

9. TÉCNICAS DE EVALUACIÓN.

- La evaluación se basará principalmente en los conocimientos adquiridos tanto en clase de teoría como en aula de informática. La participación también será evaluada.
- Para la realización de cualquier prueba evaluable no se permitirá el uso o consulta de documentación, salvo indicación expresa del profesor en convocatoria oficial

Criterios de evaluación y calificación: (referidos a las competencias trabajadas durante el curso)

La nota oscilará entre 0 y 10 puntos, los cuales se acumularán en función de los porcentajes descritos a continuación:

- Enseñanzas básicas: 30%
- Actividades prácticas y de desarrollo: 45%
- Actividades académicas dirigidas: 25%

Nota: Título II. Capítulo II. Artículo 14.2 y 14.3 de la Normativa de Régimen Académico y de Evaluación del Alumnado (aprobada en Consejo de Gobierno de la UPO el 18 de julio de 2006): “En la realización de trabajos, el **plagio** y la utilización de material no original, incluido aquél obtenido a través de Internet, sin indicación expresa de su procedencia y, si es el caso, permiso de su autor, podrá ser considerada causa de calificación de **suspense** de la asignatura, sin perjuicio de que pueda derivar en **sanción académica**.”

Corresponderá a la Dirección del Departamento responsable de la asignatura, oídos el profesorado responsable de la misma, los estudiantes afectados y cualquier otra instancia académica requerida por la Dirección del Departamento, decidir sobre la posibilidad de solicitar la apertura del correspondiente **expediente sancionador**”.

10. ORGANIZACIÓN DOCENTE SEMANAL. (Sólo hay que indicar el número de horas que a ese tipo de sesión va a dedicar el estudiante cada semana)								
SEMANA	Enseñanzas básicas (Gran Grupo) N° de horas	Enseñanzas básicas y de desarrollo (Grupo de Docencia) N° de horas	Actividades académicas dirigidas (Seminarios-Grupos de Trabajo) N° de horas	Visita y excursiones N° de horas	Tutorías especializadas N° de horas	Control de lecturas obligatorias N° de horas	Exámenes	Temas del temario a tratar
Primer Cuatrimestre								
SEMANA 1 (26-30 sep)	1,5							T1
SEMANA 2 (3-7 oct)	1,5	2						T2 / ADP1
SEMANA 3 (10,12-14 oct)	1,5	2						T2 / APD2
SEMANA 4 (17-21 oct)	1,5	2						T2 / APD3
SEMANA 5 (24-28 oct)	1,5	2	2					T3 / APD4
SEMANA 6 (31 oct -1,4 nov)	1,5							T3
SEMANA 7 (7-11 nov)	1,5	2						T3 / APD5
SEMANA 8 (14-18 nov)	1,5	2						T4 / APD6
SEMANA 9 (21-25 nov)	1,5	2	2					T4 / APD7
SEMANA 10 (28 nov - 2 dic)	1,5	2						T4 / APD8
SEMANA 11 (5,6,8-9 dic)	1,5							T4
SEMANA 12 (12-16 dic)	1,5	2	2		2			T4 / APD9
SEMANA 13 (9-13 ene)	1,5	2						T4 / APD10
SEMANA 14 (16-20 ene)	1,5		2					T5
SEMANA 16 (23-27 ene)	1,5							T5
Evaluaciones finales								
SEMANA 17 (30 ene - 3 feb)							6	
Evaluaciones finales								
SEMANA 18 (6-10 feb)								
Evaluaciones finales								

11. TEMARIO DESARROLLADO. (Con indicación de las competencias que se van a trabajar en cada tema).	
TEMA 1: INTRODUCCIÓN	
1.	Ingeniería del software. Ciclos de Desarrollo.
TEMA 2: ANÁLISIS Y EXTRACCIÓN DE REQUISITOS	
1.	Análisis de Requisitos
2.	Extracción de Requisitos
TEMA 3: DISEÑO ARQUITECTÓNICO DEL SOFTWARE	
1.	Introducción.
2.	Organización del sistema
3.	Estilos de descomposición modular
4.	Estilos de Control
5.	Arquitecturas de sistemas distribuidos
6.	Arquitectura en tres capas. Patrón Agente.
7.	Patrón MVC.
TEMA 4: EL PROCESO UNIFICADO DE DESARROLLO	
1.	Fases del proceso Unificado de Desarrollo
2.	Algunos modelos importantes
3.	Ejemplo guiado.
TEMA 4: ANÁLISIS DE PROYECTOS SOFTWARE	
1.	Análisis temporal, personal y costes
2.	Calidad software
ACTIVIDADES PRÁCTICAS Y DE DESARROLLO	
APD 1: Análisis Temporal I APD 2: Análisis Temporal II APD 3: Patrón Arquitectónico MVC I (Modelo Vista Controlador). APD 4: Patrón Arquitectónico MVC II (Modelo Vista Controlador). APD 5: Entidades Persistentes: Patrón DAO (Data Acces Object) APD 6: Acceso a Datos con JDBC. APD 7: Uso de JDBC basado en DAO bajo una arquitectura MVC. APD 8: Aplicación de Facturación I. Un ejemplo completo. APD 9: Aplicación de Facturación II. Un ejemplo completo. APD 10: Desarrollo de proyectos	

ACTIVIDADES ACADÉMICAS DIRIGIDAS
AAD 1: Toma de Requisitos AAD 2: Iteración 1 en PUD AAD 3: Iteración 2 en PUD. AAD 4: Iteración 3 en PUD
12. MECANISMOS DE CONTROL Y SEGUIMIENTO. (Al margen de los contemplados a nivel general para toda la Experiencia Piloto, se recogerán aquí los mecanismos concretos que los docentes propongan para el seguimiento de cada asignatura).