

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

TITULACIÓN: INGENIERÍA TÉCNICA EN INFORMÁTICA DE GESTIÓN
(MODALIDAD PRESENCIAL)

DATOS BÁSICOS DE LA ASIGNATURA

NOMBRE: INGENIERIA DEL SOFTWARE DE GESTION I

CÓDIGO: 912

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: 2º

CUATRIMESTRE(S): 2º

CICLO: 1º

EQUIPO DOCENTE

Responsable / Coordinador de la asignatura:

NOMBRE: ROBERTO RUIZ SANCHEZ

CENTRO/DEPARTAMENTO: ESCUELA POLITECNICA SUPERIOR/ DEPORTE E INFORMÁTICA

ÁREA: LENGUAJES Y SISTEMAS INFORMATICOS

CATEGORÍA: PROFESOR TITULAR DE UNIVERSIDAD

Nº DESPACHO: 11.1.14

E-MAIL: robertoruiz@upo.es

TF: 954977591

URL WEB: <http://www.upo.es/eps/ruiz/>

LA ASIGNATURA EN EL PROGRAMA FORMATIVO

1. DESCRIPTOR.

Diseño, propiedades y mantenimiento del software de gestión.

2. UBICACIÓN EN EL PROGRAMA FORMATIVO.

2.1. PRERREQUISITOS:

Es recomendable haber superado la asignatura de EDI donde se imparten los conceptos básicos de programación orientada a objetos con e lenguaje Java.

2.2. CONTEXTO DENTRO DE LA TITULACIÓN:

En las asignaturas de EDI y MTPI, los alumnos aprenden rudimentos de la programación. En esta asignatura se introduce al alumno con técnicas para diseño de software, mantenimiento y prueba. Esta asignatura, junto con EDI, MTP1, MPT2 y BD, conforman una completa formación en la creación de aplicaciones informáticas y dotará al alumno de conocimientos de programación necesarios tanto en asignaturas de cursos superiores como en el mercado laboral. En la asignatura de tercero ISGII se integrarán 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 de forma autónoma.

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 diseño dentro del proceso software. Comprender los problemas que resuelven los patrones de diseños. Comprender los principales problemas asociados al mantenimiento y prueba del software.
- **Procedimentales/Instrumentales (Saber hacer):** Enseñar al alumno las técnicas para la resolución de problemas mediante técnicas descriptivas de modelado. Utilizar principios y patrones de diseños. Gestionar los problemas derivados del mantenimiento y generar la pruebas del software.
- **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 esquematizar el diseño de aplicaciones a partir de principios y de prácticas exitosas.

4. OBJETIVOS.

El objetivo principal de la asignatura es que los alumnos aprendan a diseñar software con orientación a objetos, empleando como notación el lenguaje UML y usando patrones de diseño.

DISTRIBUCIÓN DEL TRABAJO PRESENCIAL.

	Gran Grupo	Grupo de Docencia	Actividades dirigidas (seminarios)
Nº de grupos	1	4	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: 0 horas de trabajo

SEGUNDO 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 básicas 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) Pruebas de evaluación y/o exámenes escritos: 6
 - B) Pruebas de evaluación y/o exámenes orales (control del Trabajo Personal): ...

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 DISEÑO
- BLOQUE 2: PATRONES DE DISEÑO
- BLOQUE 3: MANTENIMIENTO Y PRUEBAS

8. BIBLIOGRAFÍA.

8.1 GENERAL:

1. E. Freeman, E. Freeman. Head First Design Patterns. O'reilly, 2004.
2. E. Gamma et al. Patrones de Diseño. Pearson Education, 2006.
3. Ian Sommerville. Ingeniería del software (7ª edición). Pearson Education, 2005.
4. M.G. Piattini et al. Análisis y diseño de aplicaciones informáticas de gestión. RA-MA, 2006.
5. R.S. Presman. Ingeniería del Software. Un enfoque práctico (6ª edición). McGraw-Hill, 2005.
6. R.C. Martin. UML para programadores Java. Pearson Education, 2004. ISBN: 8420541095.
7. J. Lewis, J. Chase. Estructura de datos con Java. Pearson Education, 2006.

8.2 ESPECÍFICA: (con remisiones concretas en lo posible)

Tema 1: Desarrollo software.

- J. Lewis, J. Chase. Estructura de datos con Java. Pearson Education, 2006. Capítulo 1.
- Ian Sommerville. Ingeniería del software (7ª edición). Pearson Education, 2005. Capítulo 1.
- M.G. Piattini et al. Análisis y diseño de aplicaciones informáticas de gestión. RA-MA, 2006.
- R.S. Presman. Ingeniería del Software. Un enfoque práctico (6ª edición). McGraw-Hill, 2005.
- R.C. Martin. UML para programadores Java. Pearson Education, 2004. Capítulos 1, 2 y 7.

Tema 2: Introducción al diseño.

- J. Lewis, J. Chase. Estructura de datos con Java. Pearson Education, 2006. Capítulo 2.
- R.C. Martin. UML para programadores Java. Pearson Education, 2004. Capítulo 6.

Tema 3: Patrones de diseño

- E. Freeman, E. Freeman. Head First Design Patterns. O'reilly, 2004.
- E. Gamma et al. Patrones de Diseño. Pearson Education, 2006.

Tema 4: Pruebas del software

- Ian Sommerville. Ingeniería del software (7ª edición). Pearson Education, 2005. Capítulo 23.
- M.G. Piattini et al. Análisis y diseño de aplicaciones informáticas de gestión. RA-MA, 2006.
- R.S. Presman. Ingeniería del Software. Un enfoque práctico (6ª edición). McGraw-Hill, 2005.

Tema 5: Mantenimiento del software.

- Ian Sommerville. Ingeniería del software (7ª edición). Pearson Education, 2005. Capítulo 21.
- M.G. Piattini et al. Análisis y diseño de aplicaciones informáticas de gestión. RA-MA, 2006.
- R.S. Presman. Ingeniería del Software. Un enfoque práctico (6ª edición). McGraw-Hill, 2005.

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) Nº de horas	Enseñanzas básicas y de desarrollo (Grupo de Docencia) Nº de horas	Actividades académicas dirigidas (Seminarios-Grupos de Trabajo) Nº de horas	Visita y excursiones Nº de horas	Tutorías especializadas Nº de horas	Control de lecturas obligatorias Nº de horas	Exámenes	Temas del temario a tratar
Segundo Cuatrimestre								
SEMANA 1 (14-18 feb)	1,5							T1
SEMANA 2 (21-25 feb)	1,5							T1
SEMANA 3 (28 feb - 4 mar)	1,5							T1
SEMANA 4 (7-11 mar)	1,5	2						T2 / APD1
SEMANA 5 (14-18 mar)	1,5	2						T2 / APD2
SEMANA 6 (21-25 mar)	1,5	2						T3 / APD3
SEMANA 7 (28 mar - 1 abr)	1,5	2	2					T3 / APD4
SEMANA 8 (4-8 abr)	1,5	2						T3 / APD5
SEMANA 9 (11-16 abr)	1,5	2	2					T3 / APD6
SEMANA 10 (25-29 abr)	1,5	2			2			T3 / APD7
SEMANA 11 (9-13 may)	1,5	2	2					T3 / APD8
SEMANA 12 (16-20 may)	1,5	2						T4 / APD9
SEMANA 13 (23-26,27 may)	1,5	2						T4 / APD10
SEMANA 14 (30 may-3 jun)	1,5		2					T5
SEMANA 15 (6-10 jun)	1,5							T5
SEMANA 16, 17, 18, 19 y 20 (13 jun – 13 jul) Evaluaciones finales							6	

11. TEMARIO DESARROLLADO. (Con indicación de las competencias que se van a trabajar en cada tema).

BLOQUE 1: INTRODUCCIÓN AL DISEÑO

Tema 1: Desarrollo software

1. Ingeniería del software.
2. Calidad del software.
3. Modelos del ciclo de desarrollo.
4. El lenguaje UML.

Tema 2: Introducción al diseño

1. Conceptos fundamentales de la programación orientada a objetos.
2. Diseño orientado a objetos.
3. Principios de diseño.
4. Representación del diseño en UML.

BLOQUE 2: PATRONES DE DISEÑO

Tema 3: Patrones de diseño

1. Introducción.
2. Patrones de diseño clásico.
3. Catálogo de patrones.
4. Patrones creacionales.
5. Patrones estructurales.
6. Patrones de comportamiento.

BLOQUE 3: PRUEBAS Y MANTENIMIENTO DEL SOFTWARE

Tema 4: Pruebas del software

1. Conceptos generales.
2. Estrategias y técnicas de prueba.

Tema 5: Mantenimiento del software

1. Introducción.
2. Tipos de mantenimiento.
3. Técnicas de mantenimiento.
4. Cambios en el software.
5. Gestión de la configuración.

ACTIVIDADES PRÁCTICAS Y DE DESARROLLO
APD 1: Diagramas de Clase en UML. Herramientas. APD 2: Diagramas de Interacción: Secuencia y Colaboración. APD 3: Patrones de diseño. APD 4: Patrones de diseño. APD 5: Patrones de diseño. APD 6: Patrones de diseño. APD 7: Patrones de diseño. APD 8: Pruebas del software. APD 9: Pruebas del software. APD 10: Mantenimiento del software.
ACTIVIDADES ACADÉMICAS DIRIGIDAS
AAD 1: Diagramas en UML. AAD 2: Principios y patrones de diseño. AAD 3: Patrones de diseño. AAD 4: Pruebas del software.

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).