

GUIA DOCENTE DE LA ASIGNATURA Arquitecturas Software y Familias de Productos DE LA TITULACION Master Oficial en Sistemas Telemáticos e Informáticos

Profesor/es: Dr. Rafael Capilla Sevilla

Web: <http://gsyc.es/master/asignaturas/Arquitecturas.pdf>

I.- Datos iniciales ¹

Código de la asignatura ²	
Tipo ³	Obligatoria
Período de impartición ⁴	1S
Créditos	4.5
Modalidad de impartición ⁵	
Departamento	Lenguajes y Sistemas Informáticos
Prerrequisitos de acceso ⁶	Ninguno
Conocimientos recomendados ⁷	Ingeniería Software

II.- Objetivos generales

Competencias genéricas	El alumno sabrá / comprenderá conceptos avanzados de la ingeniería software y de evolución del software, los cuales complementan a los impartidos en la mayoría de las enseñanzas de Ingeniería Informática.
	El alumno será capaz de aplicar los conceptos y técnicas impartidas a casos prácticos que profundicen su conocimiento en diversos aspectos de la ingeniería software.
Competencias específicas	El alumno sabrá / comprenderá conceptos relacionados con la ingeniería inversa, evolución del software, arquitecturas software, familias de productos y gestión y técnicas de la variabilidad aplicada a los sistemas software.
	El alumno será capaz de aplicar técnicas de ingeniería inversa para la reconstrucción de arquitecturas software, técnicas de variabilidad para el desarrollo de arquitecturas software y familias de productos. Todo ello como parte integrada de procesos de mantenimiento y evolución del software.

² A cumplimentar por la Universidad

³ Tipo: Obligatorias u optativas.

⁴ Período de impartición: En el caso del grado, la docencia se organiza por cursos y semestres. En el caso del posgrado hablamos de cuatro semestres: 1S-2S-3S o 4S

⁵ A determinar por la comisión de posgrado. Puede ser presencial o semi-presencial

⁶ Anotar las asignaturas llave para poder acceder a esta asignatura.

⁷ conocimientos que sería recomendable que el estudiante dominara para poder entender adecuadamente la materia

III.- Contenido

Temario de la asignatura

Bloque temático	Tema	Apartados
I.- "Evolución del software y mantenimiento"	Tema 1.1 "Concepto y Leyes de la Evolución del Software"	Se describen las leyes básicas de la evolución del software y los conceptos básicos.
	Tema 1.2. "Tipos de Mantenimiento y Proceso"	Asimismo se explican los tipos de mantenimiento software existentes y los procesos relacionados.
II.- "Técnicas y Herramientas de Reingeniería Software"	Tema 2.1. "Concepto y Tipos de Reingeniería"	Se explica el concepto de reingeniería, sus tipos y sus técnicas asociadas.
III.- "Reutilización del Software"	Tema 2.1. "Técnicas y Herramientas de Ingeniería Inversa y Directa"	Se detallan técnicas y formas de ingeniería inversa y directa y su aplicación práctica. Se trata de obtener diseño a partir de código y de generar código a partir de diseños UML mediante herramientas apropiadas.
	Tema 3.1. "Concepto de Reutilización de Software"	Describe el concepto e ideas de la reutilización del software.
	Tema 3.2. "Tipos y Formas de Reutilización"	Se proporciona una descripción de las distintas formas de reutilización existentes, con especial énfasis en los componentes y en los diseños.
IV.- "Arquitecturas Software y Evolución"	Tema 3.3. "Reutilización de código, Componentes y Diseños"	Se profundiza en técnicas para reutilizar código fuente, componentes software y diseños software.
	Tema 4.1. "Concepto y Tipos de Arquitecturas Software"	Se expone el concepto de arquitectura software y los tipos existentes. Todo ello dentro del paradigma de la ingeniería software y su relación con los sistemas.
	Tema 4.2. "Evolución en Arquitecturas Software. Variabilidad"	Se detalla el concepto de variabilidad y su uso dentro de los procesos de mantenimiento y evolución de los sistemas software, y en particular en las arquitecturas.
	Tema 4.3. "Arquitectura del Producto. Procesos de Personalización"	Se profundiza en la obtención de la arquitectura de un producto software concreto y como se personaliza la arquitectura software para obtener una específica.
V.- "Familias de Productos"	Tema 4.4. "Arquitecturas Dirigidas por Modelos (MDA)"	De forma complementaria se explica el concepto de MDA y su aplicación práctica.
	Tema 5.1. "Concepto de Familia de Productos Software"	Se describe el concepto de línea de productos (PL) o familia de productos y su relación con las arquitecturas software.
	Tema 5.2. "Métodos y Tipos de Líneas de Productos. Planes de Producción"	Se exponen los distintos tipos de PL y su relación con los planes de producción.
	Tema 5.3. "Reingeniería en Arquitecturas de Líneas de Productos: MAP y OAR"	Se relacionan los conceptos de reingeniería con las PL a través de los métodos MAP y OAR.
	Tema 5.4. "Gestión de la Variabilidad Software"	Se describen los conceptos de la gestión de la variabilidad en el contexto de las PL.
	Tema 5.5. "Derivación de Productos"	Se describe como se realiza la derivación de productos software en una PL.

Lecturas obligatorias¹

No hay lecturas obligatorias pero si algunos libros más recomendados que otros.

Prácticas o actividades obligatorias²

Habrán 3 prácticas obligatorias. Una de reingeniería, otra de arquitecturas software y una tercera sobre líneas de productos.

IV.- Bibliografía³

General

Título	Software Architecture in Practice 2nd edition
Autor	L. Bass, P. Clements, R. kazman
Editorial	Addison-Wesley
Título	Design and Use of Software Architectures. Adopting and Evolving a Product Line Approach
Autor	J. Bosch
Editorial	Addison-Wesley
Título	Software Architecture
Autor	D. Garlan, M. Shaw
Editorial	Prentice-Hall
Título	Managing Corporate Information Systems. Evolution and Maintenance
Autor	K. Khan, Y. Zhang
Editorial	Idea Group Publishing

Por temas

Título	Software Architecture in Practice 2nd edition
Autor	L. Bass, P. Clements, R. kazman
Editorial	Addison-Wesley
Título	Software Product Line Engineering
Autor	D.M. Weiss, C.T.R. Lai
Editorial	Addison-Wesley
Título	The three R's of Software Automation
Autor	C. McClure
Editorial	Prentice-Hall
Título	Program Evolution - Processes of Software Change
Autor	M M Lehman and L A Belady
Editorial	Academic Press

Direcciones web de interés

Dirección 1 Software Engineering Institute <http://www.sei.cmu.edu>

Dirección 2 <http://www.softwareproductlines.com>

Dirección 3 Fraunhofer Institute <http://www.fraunhofer.de>

¹ Cuando sean lecturas evaluables.

² Cuando sean evaluables.

³ Se recomienda no exceder de 20 títulos

V.- Tiempo de trabajo ¹

Asistencia a clases teóricas	15
Asistencia a clases prácticas	30
Asistencia a clases de problemas	0
Realización de exámenes	0
Asistencia a tutorías	15
Asistencia a actividades relacionadas: jornadas, seminarios, etc	7,5
Preparación de clases teóricas	10
Preparación de clases prácticas y/o problemas	57,5
Preparación de exámenes	0
Total de horas de trabajo del estudiante	135

a = 10 x número de créditos de la asignatura

b = 5 x número de créditos de la asignatura

c = 15 x número de créditos de la asignatura

d = 30 x número de créditos de la asignatura

Distribución horaria de la parte presencial:

Con carácter general, se recomienda establecer para una asignatura semestral de 6 créditos, (60 horas de clase), cuatro horas semanales de clase presencial. Para asignaturas con menos créditos, se distribuirán las clases respetando la proporción anterior (1,5 créditos, 1 hora de clase semanal al semestre).

VI.- Metodología y plan de trabajo

Clases teóricas

Fecha ²	Temas	Metodología
1 semana octubre 2006	I.- "Evolución del software y mantenimiento"	Clases magistrales Modelo participativo
2 semana octubre 2006	II.- "Técnicas y Herramientas de Reingeniería Software"	Clases magistrales Estudio de casos Modelo participativo
2 semana noviembre 2006	III.- "Reutilización del Software"	Clases magistrales Estudio de casos Modelo participativo
3 semana noviembre 2006	IV.- "Arquitecturas Software y Evolución"	Clases magistrales Estudio de casos Modelo participativo
3 semana diciembre 2006	V.- "Familias de Productos"	Clases magistrales Estudio de casos Modelo participativo

¹ El volumen de trabajo está referido al trabajo del estudiante. La dedicación de los profesores a las diferentes actividades docentes permite reconocer y valorar más adecuadamente su carga de trabajo, y por ello es conveniente desarrollar herramientas que permitan conocer el tiempo que efectivamente dedica a sus alumnos más allá de las horas lectivas, pero no son objeto de las guías docentes.

² Especificar la semana o período en que está previsto desarrollar el tema.

Clases prácticas

Fecha	Temas	Metodología
3 y 4 semana octubre y 1 semana de noviembre de octubre 2006	II.- "Técnicas y Herramientas de Reingeniería Software"	Experimentación (laboratorio)
4 y 5 semana de noviembre 2006	IV.- "Arquitecturas Software y Evolución"	Experimentación (laboratorio)
Enero de 2007	V.- "Familias de Productos"	Experimentación (laboratorio)

Clases de problemas

Fecha	Temas	Metodología

Tutorías

Fecha	Martes y Miércoles de 10:30-13:30

Otras actividades

Fecha		
Por definir	1 o 2 seminarios temáticos	

VII.- Métodos de evaluación:

Criterio	Ponderación ¹	Fecha	Temas / Contenido
Examen escrito			
Examen oral	20 %		Presentación de la práctica obligatoria
Asistencia a clase			
Actividades fuera del aula	80 %		Prácticas obligatorias
Situaciones de prueba			
Otros			

¹ La ponderación se establecerá otorgando a cada criterio de evaluación el porcentaje estimado por el profesor.

Los criterios establecidos son orientativos, por lo que no todos se utilizan en todas las asignaturas, y se pueden incluir otros métodos no incluidos.

VIII.- Profesorado

Nombre y apellidos	Rafael Capilla Sevilla
Materia	Arquitecturas Software y Familias de Programas
Categoría	Profesor Colaborador Doctor
Universidad	Para master universitarios
Titulación Académica	Dr. En Informática, Ldo. En Informática
Experiencia Docente	Profesor asociado. EU Informática – UPM 04/04/00-18/10/00 Profesor Asociado. EUIT Telecomunicación – UPM 19/10/00-30/09/01 Profesor Asociado. Universidad Rey Juan Carlos 01/10/01-30/04/06 Profesor Colaborador. Universidad Rey Juan Carlos 01/05/06-actualidad
Experiencia Investigadora	Mas de 25 artículos en congresos internacionales con revisión Co-autor de 1 libro nacional y 1 capítulo de libro internacional 9 meses como becario de investigación en 1 proyecto europeo Esprit IV FLEX 29158 Participación en dos proyectos CICYT a tiempo parcial y tres de la URJC 2 estancias Erasmus en Italia y Turquía de 2 semanas cada una 1 estancia de 3 meses en la Universidad de Groningen (Holanda)
Experiencia profesional	Analista-Programador en INTTESA 19/03/89-09/04/91 Programador (Técnico de Sistemas). Facultad de Informática de la Universidad de Sevilla 24/10/91-05/09/99