ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA

GUIA DOCENTE

CURSO 2016/2017

Grado en Ingeniería Informática itinerario Ingeniería del Software

DATOS DE LA ASIGNATURA				
Nombre:				
Arquitectura del Software Dirigida por Modelos				
Denominación en inglés:				
Model Driven Architecture				
Código:		**Carácter:**		
606010216		Obligatorio		
Horas:				
	Totales	**Presenciales**	**No presenciales**	
Trabajo estimado:	150	60	90	
Créditos:				
	Grupos reducidos			
Grupos grandes	**Aula estándar**	**Laboratorio**	**Prácticas de campo**	**Aula de informática**
3	0	0	0	3
Departamentos:		**Áreas de Conocimiento:**		
Tecnologías de la Información		Lenguaje y Sistemas Informáticos		
Curso:		**Cuatrimestre:**		
3º - Tercero		Segundo cuatrimestre		

DATOS DE LOS PROFESORES			
Nombre:	**E-Mail:**	**Teléfono:**	**Despacho:**
*Suárez Fábrega, Antonio José	asuarez@uhu.es	87677	39

*Profesor coordinador de la asignatura

Consultar los horarios de la asignatura

DATOS ESPECÍFICOS DE LA ASIGNATURA

1. Descripción de contenidos

1.1. Breve descripción (en castellano):

Patrones de diseño. Composición y clasificación de los patrones arquitectónicos. Arquitecturas orientadas a componentes y servicios. Técnicas de desarrollo dirigidas por modelos.

1.2. Breve descripción (en inglés):

Design patterns. Arquitectural patterns. Model driven architecture

2. Situación de la asignatura

2.1. Contexto dentro de la titulación:

Debido a sus contenidos y a su carácter práctico, usando las tecnologías más actuales para el desarrollo software, es una asignatura esencial dentro de la formación general de un informático dentro de la especialidad de ingeniería del software. Por todo ello, se debe prestar una especial atención a la enseñanza de los contenidos de esta asignatura, que se considera esencial para la futura incorporación al mercado laboral.

2.2. Recomendaciones:

Se recomienda que el alumno tenga superadas la asignatura Fundamentos y Principios de la Ingeniería del Software.

3. Objetivos (Expresados como resultados del aprendizaje):

Conocer los principales patrones de diseño, su estructura y programación. Profundizar en la iniciativa del OMG conocida como Model Driven Architecture (MDA), dominar los conceptos de modelado de software desde los modelos más abstractos de captura y análisis de requisitos hasta los más concretos de diseño detallado e implementación.

4. Competencias a adquirir por los estudiantes

4.1. Competencias específicas:

- **CE3-IS:** Capacidad de dar solución a problemas de integración en función de las estrategias, estándares y tecnologías disponibles
- **CE4-IS:** Capacidad de identificar y analizar problemas y diseñar, desarrollar, implementar, verificar y documentar soluciones software sobre la base de un conocimiento adecuado de las teorías, modelos y técnicas actuales
- **CE6-IS:** Capacidad para diseñar soluciones apropiadas en uno o más dominios de aplicación utilizando métodos de la ingeniería del software que integren aspectos éticos, sociales, legales y económicos.

4.2. Competencias básicas, generales o transversales:

- **CB4:** Que los estudiantes puedan transmitir información, ideas, problemas y soluciones a un público tanto especializado como no especializado
- **CG0:** Capacidad de análisis y síntesis: Encontrar, analizar, criticar (razonamiento crítico), relacionar, estructurar y sintetizar información proveniente de diversas fuentes, así como integrar ideas y conocimientos.
- **G03:** Capacidad para la resolución de problemas
- **G04:** Capacidad para tomar decisiones basadas en criterios objetivos (datos experimentales, científicos o de simulación disponibles) así como capacidad de argumentar y justificar lógicamente dichas decisiones, sabiendo aceptar otros puntos de vista
- **G05:** Capacidad de trabajo en equipo.
- **G07:** Motivación por la calidad y la mejora continua, actuando con rigor, responsabilidad y ética profesional.
- **T02:** Conocimiento y perfeccionamiento en el ámbito de las TIC’s

5. Actividades Formativas y Metodologías Docentes

5.1. Actividades formativas:

- Sesiones de Teoría sobre los contenidos del Programa.
- Sesiones de Resolución de Problemas.
- Sesiones Prácticas en Laboratorios Especializados o en Aulas de Informática.

5.2. Metologías docentes:

- Clase Magistral Participativa.
- Desarrollo de Prácticas en Laboratorios Especializados o Aulas de Informática en grupos reducidos.
- Resolución de Problemas y Ejercicios Prácticos.
- Tutorías Individuales o Colectivas. Interacción directa profesorado-estudiantes.
- Evaluaciones y Exámenes.

5.3. Desarrollo y justificación:

- Sesiones Académicas de Teoría:
Consisten en clases magistrales donde se impartirá la base teórica de la asignatura y se expondrán ejemplos aclaratorios de la misma al grupo, que se supone compuesto de no más de 80 alumnos. Las sesiones de teoría se irán intercalando con las sesiones de problemas a lo largo del curso, de manera que una vez finalizado un tema teórico con sus correspondientes sesiones académicas de teoría, se impartirán sesiones de problemas. El profesor solicitará la participación activa del alumno mediante preguntas rápidas.
- Sesiones de Problemas:
Consisten en la realización de problemas relacionados con los conceptos y métodos operativos de la asignatura. Las sesiones serán de una hora. El profesor explicará uno o varios problemas tipo. En estas sesiones se fomentará la participación del alumnado en la resolución de los problemas planteados.
Prácticas de Laboratorio:
Consisten en el diseño e implementación de programas. Los alumnos dispondrán con antelación la relación de problemas a resolver y la metodología de trabajo. Los grupos de prácticas serán de no más de 24 alumnos y el trabajo se realizará de forma individual.
La asistencia a las sesiones de laboratorio es obligatoria. La participación activa de los alumnos en la resolución de problemas será valorada.

6. Temario desarrollado:

Tema 1. Arquitecturas Software
1.1. Estilos arquitectónicos.
1.2. Notaciones actuales para representación de las arquitecturas software.
1.3. Arquitecturas orientas a components y servicios.
Tema 2. Desarrollo dirigido por modelos
2.1. Introducción al desarrollo dirigido por modelos.
2.2. MDA (Model Driven Architecture) y factorías de software.
Tema 3. Desarrollo utilizando patrones-software
3.1. Análisis y diseño con patrones.
3.2. Catálogo de patrones.

7. Bibliografía

7.1. Bibliografía básica:

- Craig Larman. UML y patrones Una introducción al análisis y diseño orientado a objetos y al PU. Prentice Hall
- Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides. Patrones de Diseño. Addison Wesley
- Kleppe, Anneke G. MDA explained : the model driven architecture : practice and promise / Addison-Wesley,2007.
- Stephen J. Mellor MDA distilled: principles of model-driven architecture Addison-Wesley, cop. 2004.

7.2. Bibliografía complementaria:

- García Rubio. Desarrollo software dirigido por modelos: conceptos, métodos y herramientas. RA-Ma. 2013

8. Sistemas y criterios de evaluación.

8.1. Sistemas de evaluación:

- Examen de teoría/problemas
- Defensa de Prácticas
- Examen de prácticas

8.2. Criterios de evaluación y calificación:

- Examen final que constará de preguntas teóricas y problemas: 60% de la calificación, es necesario obtener al menos un 24% para poder obtener el aprobado. - Realización de prácticas en laboratorio: 40% de la calificación final

9. Organización docente semanal orientativa:

Semanas	Grupos Grandes	Grupos Reducidos Aula Estándar	Grupos Reducidos Aula de Informática	Grupos Reducidos Laboratorio	Grupos Reducidos prácticas de campo	Pruebas y/o actividades evaluables	Contenido desarrollado
#1	2	0	2	0	0		Tema 1
#2	2	0	2	0	0		Tema 1
#3	2	0	2	0	0		Tema 1
#4	2	0	2	0	0		Tema 2
#5	2	0	2	0	0		Tema 2
#6	2	0	2	0	0		Tema 2
#7	2	0	2	0	0		Tema 2
#8	2	0	2	0	0		Tema 2
#9	2	0	2	0	0		Tema 2
#10	2	0	2	0	0		Tema 3
#11	2	0	2	0	0		Tema 3
#12	2	0	2	0	0		Tema 3
#13	2	0	2	0	0		Tema 3
#14	2	0	2	0	0		Tema 3
#15	2	0	2	0	0		Tema 3
	30	0	30	0	0		