



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:

606010216

Carácter:

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	3	0	0

Departamentos:

Tecnologías de la Información

Áreas de Conocimiento:

Lenguajes y Sistemas Informáticos

Curso:

3º - Tercero

Cuatrimestre:

Segundo cuatrimestre

DATOS DE LOS PROFESORES

Nombre:

*Suárez Fábrega, Antonio
José

E-Mail:

asuarez@uhu.es

Teléfono:

959217677

Despacho:

ETP136

*Profesor coordinador 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. Architectural 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.
- **CT2:** Desarrollo de una actitud crítica en relación con la capacidad de análisis y síntesis.
- **CT3:** Desarrollo de una actitud de indagación que permita la revisión y avance permanente del conocimiento.
- **CT4:** Capacidad de utilizar las Competencias Informáticas e Informacionales (CI2) en la práctica profesional.

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. Metodologías 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.

6. Temario desarrollado:

Tema 1. Arquitecturas Software

- 1.1. Estilos arquitectónicos.
- 1.2. Notaciones actuales para representación de las arquitecturas software.
- 1.3. Arquitecturas orientadas a componentes 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:

Criterios de evaluación continua del alumno:

- Examen final teórico. (EF) que constará de preguntas teóricas: 30% de la calificación. En este ítem se evaluarán las siguientes competencias:
 - 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.
 - CG07 - 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
 - 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.
- Examen final de problemas: 30% de la calificación. En este ítem se evaluarán las siguientes competencias:
 - 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.
 - CG07 - 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
 - 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.
- Realización de prácticas en laboratorio (RPL): 40% de la calificación final,
 - de los cuales un 30% corresponde a defensa de prácticas. En este ítem se evaluarán las siguientes competencias:
 - CG03 - Capacidad para la resolución de problemas
 - CG04 - 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
 - CG05 - Capacidad de trabajo en equipo.
 - CG07 - Motivación por la calidad y la mejora continua, actuando con rigor, responsabilidad y ética profesional.
 - CB4 - Que los estudiantes puedan transmitir información, ideas, problemas y soluciones a un público tanto especializado como no especializado
 - T02 - Conocimiento y perfeccionamiento en el ámbito de las TIC
 - 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

- Y un 10% a examen de prácticas. En este ítem se evaluarán las siguientes competencias:

CG04 - 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

CB4 - Que los estudiantes puedan transmitir información, ideas, problemas y soluciones a un público tanto especializado como no especializado

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.

Aquellos estudiantes que así lo consideren pueden optar por la realización de una evaluación única final. En este caso deberá presentar una solicitud en el REGISTRO GENERAL de la Universidad, en cualquiera de sus REGISTROS AUXILIARES o en el REGISTRO TELEMÁTICO, dirigida a la dirección del departamento y al coordinador de la asignatura. La evaluación única final consistirá, para todas las convocatorias oficiales, en un solo acto académico a celebrar en las fechas indicadas por el centro y que, para todas las convocatorias, estará formado por las siguientes pruebas:

- Examen Final de teoría (60 %), donde se evaluarán las competencias definidas en EF: 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, tiene un carácter presencial e individual y una duración de una tres horas. No se podrá utilizar ningún tipo de documentación. En la medida de lo posible, se realizará en un aula de informática.
- Examen/Defensa de prácticas de Laboratorio (40 %), donde se evaluarán las competencias definidas en RPL: La prueba consistirá en un examen en el que se presentará un enunciado práctico, tienen un carácter presencial e individual y una duración de tres horas. No se podrá utilizar ningún tipo de documentación. Se realizará en un aula de informática.
- La calificación final de la asignatura para la evaluación única final se obtendrá sumando las calificaciones parciales obtenidas en cada una de las pruebas que la forman siempre y cuando se alcance, al menos, el 40% de la nota máxima del bloque de teoría.

Criterios para otorgar Matrícula de Honor:

- La mención de Matrícula de Honor podrá ser otorgada a alumnos que hayan obtenido una calificación igual o superior a 9.0. Como norma general, estas menciones se irán otorgando en orden descendente a la nota final obtenida. En ningún caso el número de Matrículas de Honor concedidas será superior al máximo establecido para la asignatura en el curso académico en curso. En caso de empate, primará aquel alumno que haya obtenido mayor número de 10. Si el empate persistiera, se convocaría a los alumnos implicados a una nueva prueba de evaluación. Para todos los materiales entregados por parte de los estudiantes se asume de forma implícita la declaración de originalidad de los mismos, entendida en el sentido de que no ha utilizado fuentes sin citarlas debidamente. La detección de plagio en cualquiera de estos materiales, y en aplicación del artículo 15 del Reglamento de evaluación para las titulaciones de grado y máster oficial de la Universidad de Huelva, conllevará la calificación numérica de cero en la asignatura, independientemente del resto de calificaciones que los alumnos hubieran obtenido. Además, se iniciará el procedimiento disciplinario oportuno ante la Comisión de Docencia del Departamento.

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	0	2	0		Tema 1	
#2	2	0	0	2	0		Tema 1	
#3	2	0	0	2	0		Tema 1	
#4	2	0	0	2	0		Tema 2	
#5	2	0	0	2	0		Tema 2	
#6	2	0	0	2	0		Tema 2	
#7	2	0	0	2	0		Tema 2	
#8	2	0	0	2	0		Tema 2	
#9	2	0	0	2	0		Tema 2	
#10	2	0	0	2	0		Tema 3	
#11	2	0	0	2	0		Tema 3	
#12	2	0	0	2	0		Tema 3	
#13	2	0	0	2	0		Tema 3	
#14	2	0	0	2	0		Tema 3	
#15	2	0	0	2	0		Tema 3	
	30	0	0	30	0			