

**Guía docente de la asignatura**

Asignatura	DISEÑO, INTEGRACIÓN Y ADAPTACIÓN DE SOFTWARE		
Materia	TECNOLOGÍAS DE LA INFORMACIÓN		
Módulo			
Titulación	GRADO EN INGENIERÍA INFORMÁTICA		
Plan	545	Código	46937
Periodo de impartición	2º CUATRIMESTRE	Tipo/Carácter	OP
Nivel/Ciclo	GRADO	Curso	3º
Créditos ECTS	6 ECTS		
Lengua en que se imparte	CASTELLANO		
Profesor/es responsable/s	José Manuel Marqués Corral		
Datos de contacto (E-mail, teléfono...)	TELÉFONO: 983 423000 / ext. 5638 E-MAIL: jmmc@infor.uva.es		
Horario de tutorías	Véase www.uva.es		
Departamento	Departamento de Informática.		



1. Situación / Sentido de la Asignatura

El diseño de software desempeña un papel fundamental en el desarrollo del software¹ y constituye una de las principales áreas de conocimiento de la ingeniería del software. El diseño de software incluye el conjunto de conceptos, técnicas, estrategias, representaciones y patrones utilizados para determinar los componentes o sistemas a implementar.

La propia evolución y complejidad de los actuales productos y sistemas software ha llevado a la necesidad de considerar los sistemas desde una perspectiva arquitectónica. En la actualidad los productos y sistemas software están formados por un gran número de componentes heterogéneos que interactúan en muchas ocasiones en entornos distribuidos. La experiencia demuestra que un buen diseño arquitectónico puede reducir drásticamente las tasas de fallo..

En esta asignatura se proporcionará a los estudiantes una introducción a los principios y las metodologías de diseño de las arquitecturas software de mayor difusión y aceptación tanto en el ámbito académico como en el industrial. En una primera aproximación se pueden clasificar las arquitecturas de tecnologías de la información en dos niveles: arquitecturas empresariales (*Enterprise Architecture*) y arquitecturas de aplicación (*Application Architecture*). Es cierto que puede considerarse, algunos autores y soluciones concretas así lo hacen, un nivel intermedio, la arquitectura de proyecto.

Considerando que esta es una asignatura de grado, la formación previa de los alumnos y, sobre todo, las limitaciones temporales, se estima que el énfasis ha de ponerse en la formación relacionada con el nivel de arquitectura de aplicación software.

Para la formación en arquitecturas de aplicación, la asignatura se centrará en el estudio de los fundamentos teóricos y prácticos básicos para el modelado y documentación del diseño de alto nivel de un sistema. Se persigue que el estudiante comprenda el concepto de arquitectura software, que conozca alguno de los patrones arquitectónicos de mayor utilización y difusión y, por supuesto, que sea capaz de desarrollar y documentar el diseño arquitectónico y detallado de un sistema software.

1.1 Contextualización

Esta asignatura forma parte de la materia "Tecnologías de la Información". La asignatura está claramente orientada a conseguir que los alumnos sean capaces de adquirir las habilidades necesarias para abordar el diseño de un sistema informático aplicando principios, métodos, herramientas y prácticas propias de la ingeniería.

1.2 Relación con otras materias

¹ Software Engineering Body of Knowledge (SWEBOK), Bourque, P. & Dupuis, R. (ed.). IEEE Computer Society, 2004



Fundamentos de Ingeniería del Software, Fundamentos de Programación, Programación Orientada a Objetos, Interacción Persona – Computador, Tecnología y Diseño de Bases de Datos.

1.3 Prerrequisitos

Se recomienda que los alumnos hayan cursado con aprovechamiento al menos las siguientes asignaturas del vigente plan de estudios: Fundamentos de Ingeniería del Software, Programación Orientada a Objetos, Interacción Persona-Computadora y Tecnología y Diseño de Bases de Datos.

Se recomienda que el alumno tenga un nivel de inglés que le permita la comprensión de bibliografía básica y complementaria recomendada para la asignatura.

Dadas las características inherentes a todo proyecto de diseño de un sistema software, son muy recomendables la constancia, la iniciativa personal y la predisposición al trabajo colectivo en grupo. La asimilación de los contenidos teóricos de la asignatura implica, por una parte, la capacidad de lectura crítica de los textos básicos y complementarios puestos a su disposición, pero además la búsqueda proactiva del material complementario en la red publicado por los grandes fabricantes de hardware y software, y que resulta imprescindible para el desarrollo de un buen diseño.





2. Competencias

2.1 Generales

Código	Descripción
G01	Capacidad para concebir, redactar, organizar, planificar, desarrollar y firmar proyectos en el ámbito de la ingeniería en informática que tengan por objeto el desarrollo o la explotación de sistemas, servicios y aplicaciones informáticas.
G03	Capacidad para diseñar, desarrollar, evaluar y asegurar la accesibilidad, ergonomía, usabilidad y seguridad de los sistemas, servicios y aplicaciones informáticas, así como de la información que gestionan
G04	Capacidad para definir, evaluar y seleccionar plataformas hardware y software para el desarrollo y la ejecución de sistemas, servicios y aplicaciones informáticas.
G08	Conocimiento de las materias básicas y tecnologías, que capaciten para el aprendizaje y desarrollo de nuevos métodos y tecnologías, así como las que les doten de una gran versatilidad para adaptarse a nuevas situaciones.
G09	Capacidad para resolver problemas con iniciativa, toma de decisiones, autonomía y creatividad. Capacidad para saber comunicar y transmitir los conocimientos, habilidades y destrezas de la profesión de Ingeniero en Informática
G10	Conocimientos para la realización de mediciones, cálculos, valoraciones, tasaciones, peritaciones, estudios, informes, planificación de tareas y otros trabajos análogos de informática.

2.2 Específicas

TI1.	Capacidad para comprender el entorno de una organización y sus necesidades en el ámbito de las tecnologías de la información y las comunicaciones.
TI2.	Capacidad para seleccionar, diseñar, desplegar, integrar, evaluar, construir, gestionar, explotar y mantener las tecnologías de hardware, software y redes, dentro de los parámetros de coste y calidad adecuados.
TI3.	Capacidad para emplear metodologías centradas en el usuario y la organización para el desarrollo, evaluación y gestión de aplicaciones y sistemas basados en tecnologías de la información que aseguren la accesibilidad, ergonomía y usabilidad de los sistemas.
IS3.	Capacidad de dar solución a problemas de integración en función de las estrategias, estándares y tecnologías disponibles
IS4.	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



3. Objetivos

Código	Descripción
RA1	Conocer los principios y técnicas del análisis de requisitos y del modelado de sistemas software
RA2	Conocer los principios y conceptos fundamentales de la arquitectura y el diseño de sistemas software.
RA3	Ser capaces de modelar y diseñar la arquitectura de sistemas TI, ajustándose a un conjunto de requisitos funcionales y no funcionales
RA4	Ser capaces de aplicar patrones arquitectónicos y de diseño en el proceso de desarrollo de aplicaciones TI

4. Tabla de dedicación del estudiante a la asignatura

ACTIVIDADES PRESENCIALES	HORAS	ACTIVIDADES NO PRESENCIALES	HORAS
Clases teórico-prácticas (T/M)	30	Estudio y trabajo autónomo individual	45
Clases prácticas de aula (A)		Estudio y trabajo autónomo grupal	45
Laboratorios (L)	28		
Prácticas externas, clínicas o de campo			
Seminarios (S)	2		
Tutorías grupales (TG)			
Evaluación (fuera del periodo oficial de exámenes)	0		
Total presencial	60	Total no presencial	90

5. Bloques temáticos

Bloque 1: Arquitecturas y diseño de aplicaciones software.

Carga de trabajo en créditos ECTS:

a. Contextualización y justificación

La mayoría de los estándares y recomendaciones curriculares relacionados con la enseñanza universitaria en ingeniería del software establecen la arquitectura y el diseño software como troncales. El *Computing Curriculum Software Engineering*² recomienda el diseño y la arquitectura software como una de sus diez áreas esenciales. El ACM Computing Curricula 2004 recomienda el diseño software tanto para la especialización en ingeniería del software como en la especialización en Computer Science.

² <http://sites.computer.org/ccse/>



En este bloque se estudiarán aquellos aspectos más relevantes y de mayor difusión en la industria del software de la disciplina de la arquitectura y el diseño software. Se trabajará, desde un enfoque coherente e integrado, en las técnicas, métodos y herramientas relacionadas con el diseño de alto nivel de aplicaciones software.

Los contenidos de este bloque se estructuran de la siguiente manera: en primer lugar se presentarán los conceptos, objetivos y definiciones del diseño, para a continuación introducir el concepto de arquitectura del software, los estilos arquitectónicos y las estructuras genéricas para los tipos de aplicaciones más habituales. Por último se aborda el modelado y descripción del diseño conceptual y técnico desde una perspectiva de desarrollo soportado por patrones, utilizando los métodos del diseño orientado al objeto y el lenguaje de modelado UML 2.0 para la documentación de los diseños elaborados.

b. Objetivos de aprendizaje

Código	Descripción
RA1	Conocer los principios y técnicas del análisis de requisitos y del modelado de sistemas software
RA2	Conocer los principios y conceptos fundamentales de la arquitectura y el diseño de sistemas software.
RA3	Ser capaces de modelar y diseñar la arquitectura de sistemas TI, ajustándose a un conjunto de requisitos funcionales y no funcionales
RA4	Ser capaces de aplicar patrones arquitectónicos y de diseño en el proceso de desarrollo de aplicaciones TI

c. Contenidos

Programa de Teoría.

1. Introducción al modelado y diseño del software.
2. Especificación y análisis de requisitos software.
3. Diseño de la Arquitectura Software. Patrones arquitectónicos
4. Diseño de software. Métodos y técnicas.

d. Métodos docentes

Ver anexo: Métodos docentes.

Plan de trabajo

e.

Ver anexo: Cronograma

f. Evaluación

Ver apartado 7. Sistema de calificaciones.



g. Bibliografía básica

M. Fowler, *Patterns of Enterprise Application Architecture*, 1.ª ed. Addison-Wesley Professional, 2002

C. Larman, *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development, 3rd Edition*, Dimensions: 8x10. 2004.

I. Sommerville, *Ingeniería Del Software*. Pearson Educación, 2005.

h. Bibliografía complementaria

F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, M. Stal, y M. Stal, *Pattern-Oriented Software Architecture Volume 1: A System of Patterns*, Volume 1. Wiley, 1996.

E. H., Richard; Johnson, Ralph; Vlissides, John Gamma, *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional.

i. Recursos necesarios

- » «Microsoft Application Architecture Guide, 2nd Edition». [Online]. Available: <http://msdn.microsoft.com/en-us/library/ff650706.aspx>. [Accessed: 31-may-2012].
- » «OpenUP». [Online]. Available: <http://epf.eclipse.org/wikis/openup/>. [Accessed: 31-may-2012].
- » «Software Architecture | Overview». [Online]. Available: <http://www.sei.cmu.edu/architecture/?location=secondary-nav&source=18823>. [Accessed: 30-may-2012].
- » «What is a software architecture?» [Online]. Available: <http://www.ibm.com/developerworks/rational/library/feb06/eeles/>. [Accessed: 30-may-2012].

6. Temporalización (por bloques temáticos)

BLOQUE TEMÁTICO	CARGA ECTS	PERIODO PREVISTO DE DESARROLLO
Arquitecturas y diseño de aplicaciones software	6 ECTS	Semanas 1 a 15



7. Sistema de calificaciones – Tabla resumen

INSTRUMENTO/PROCEDIMIENTO	PESO EN LA NOTA FINAL	OBSERVACIONES
Trabajos evaluables de laboratorio Análisis de Requisitos. Modelos. Modelado de la arquitectura del sistema. Diseño de la capa de negocio Diseño de la capa de acceso a datos Diseño detallado. Patrones GoF. Presentación y defensa de los trabajos de laboratorio	30%	Fecha aproximada de realización: 1. Semanas 1 a 4 2. Semanas 5 a 6 3. Semanas 7 a 8 4. Semanas 9 a 10 5. Semanas 11 a 12 6. Semanas 13 a 15
Examen final escrito	70%	Periodo de exámenes

CRITERIOS DE CALIFICACIÓN

• **Convocatorias ordinaria y extraordinaria:**

- El examen de la parte teórica se realizará mediante examen escrito sobre las materias incluidas en el programa de la asignatura. Incluirá la elaboración de modelos de diseño correspondientes a supuestos prácticos. Se valorará el ajuste de la solución elaborada al problema planteado y la correcta utilización de los principios y notaciones de diseño.
- La evaluación de la parte práctica se realizará sobre el trabajo de laboratorio realizado en grupo a lo largo del curso.
- Para poder superar la asignatura será necesario tener superadas por separado cada una de las partes de la asignatura, la teórica y la práctica.
 - La parte de laboratorio se supera obteniendo más 5 puntos sobre 10 y habiendo realizado todos los trabajos evaluables de laboratorio.
 - La parte de teoría se supera obteniendo más de 4 puntos sobre 10.
- Para establecer la calificación final de un alumno se tiene en cuenta la calificación obtenida en la práctica de laboratorio y la obtenida en el examen escrito de teoría.
Sea
 - $NTotal = 0.3 \cdot NPL + 0.7 \cdot NEF$; donde NPL es la nota, sobre 10 puntos, de la práctica de laboratorio y NEF la calificación, sobre 10 puntos, del examen final.

La nota final en la asignatura, NFinal, para un alumno será:

- Si ha obtenido la calificación mínima en la práctica de laboratorio y examen final, entonces: **NFinal=NTotal**
- En caso contrario: **NFinal = min (3.9, NTotal)**

8. Anexo: Métodos docentes

Actividad	Metodología
Clase de teoría	<ul style="list-style-type: none"> Clase magistral participativa Estudio de casos en aula Resolución de problemas
Clase práctica	<ul style="list-style-type: none"> Realización de ejercicios y pequeños supuestos prácticos. El trabajo se realizará en grupos (2/3 alumnos), siguiendo un enfoque colaborativo.
Seminarios	<ul style="list-style-type: none"> Talleres de aprendizaje
Tutoría	<ul style="list-style-type: none"> Evaluación de los contenidos teóricos y de los proyectos



--	--

9. Anexo: Cronograma de actividades previstas

Semana	Teoría	Seminario	Laboratorio	Contenidos /Actividades	
1	2	2		2T	Introducción. Conceptos básicos.
					.
				2S	UML. Herramienta CASE
2	2		2	2T	Introducción. Modelado y diseño software
				2L	Elaboración de modelos de análisis de requisitos.
3	2		2	2T	Arquitecturas Software. Visión general
				2L	Elaboración de modelos de análisis de requisitos.
4	2		2	2T	Organización y descomposición de los sistemas software.
				2L	Elaboración de modelos de análisis de requisitos
5	2		2	2T	Diseño de la arquitectura. Estructuración del sistema. POSA (I)
				2L	Modelado de la arquitectura del sistema.
6	2		2	2T	Diseño de la arquitectura. Estructuración del sistema. POSA (II)
				2L	Modelado de la arquitectura del sistema
7	2		2	2T	Diseño de la arquitectura lógica del sistema.
				2L	Diseño de la capa de negocio.
8	2		2	2T	Diseño software. Patrones de diseño de la lógica de negocio.
				2L	Diseño de la capa de negocio.
9	2		2	2T	Diseño software. Patrones de diseño de interacción con la capa de datos.
				2L	Diseño de la capa de acceso a datos
10	2		2	2T	Diseño software. Patrones de diseño de la lógica de presentación
				2L	Diseño de la capa de acceso a datos.



Semana	Teoría	Seminario	Laboratorio	Contenidos /Actividades	
11	2		2	2T	Diseño de software. Patrones GoF(I).
				2L	Diseño con patrones GoF
12	2		2	2T	Diseño de software. Patrones GoF(II)
				2L	Diseño con patrones GoF.
13	2		2	2T	Diseño detallado.
				2L	Presentación y defensa de trabajos de laboratorio
14	2		2	2T	Diseño detallado.
				2L	Presentación y defensa de trabajos de laboratorio
15	2		2	2T	Diseño detallado.
				2L	Presentación y defensa de trabajos de laboratorio

