



## Guía docente de la asignatura

<b>Asignatura</b>	DISEÑO, INTEGRACIÓN Y ADAPTACIÓN DE SOFTWARE		
<b>Materia</b>	TECNOLOGÍAS DE LA INFORMACIÓN		
<b>Módulo</b>	TECNOLOGÍAS ESPECÍFICAS		
<b>Titulación</b>	GRADO EN INGENIERÍA INFORMÁTICA		
<b>Plan</b>	545	<b>Código</b>	46937
<b>Periodo de impartición</b>	2º CUATRIMESTRE	<b>Tipo/Carácter</b>	OBLIGATORIA (Mención TI)
<b>Nivel/Ciclo</b>	GRADO	<b>Curso</b>	3º
<b>Créditos ECTS</b>	6 ECTS		
<b>Lengua en que se imparte</b>	CASTELLANO		
<b>Profesor/es responsable/s</b>	José Manuel Marqués Corral		
<b>Datos de contacto (E-mail, teléfono...)</b>	TELÉFONO: 983 423000 / ext. 5638 E-MAIL: jmmc@infor.uva.es		
<b>Horario de tutorías</b>	Véase <a href="http://www.uva.es">www.uva.es</a> → Centros → Campus de Valladolid → Escuela Técnica Superior de Ingeniería Informática → Tutorías		
<b>Departamento</b>	Departamento de Informática.		

## 1. Situación / Sentido de la Asignatura

---

Los productos y sistemas software son estructuras constituidas por un gran número de componentes heterogéneos que interactúan en muchas ocasiones en entornos distribuidos. El incremento en tamaño y complejidad de los sistemas software obliga a asegurarse de que los atributos de calidad del producto final tales como eficiencia, usabilidad, confiabilidad, seguridad, etc., pueden ser verificados y estimados antes de comenzar la producción del software. Esto obliga a un modelado y diseño cuidadoso, y desde las primeras fases del proceso, de la arquitectura del producto software. La elaboración de modelos facilita información acerca de las consecuencias de construir un producto software antes de proceder a su desarrollo.

El modelado y el diseño de las estructuras software son dos aspectos básicos del cuerpo de conocimiento de la ingeniería del software y forma parte del conjunto de materias esenciales de la formación en informática<sup>1</sup>. En base a las consideraciones anteriores, se plantea una asignatura en la se proporcionará a los estudiantes formación en los principios y las metodologías del modelado y el diseño software. Se parte del estudio de las características del modelado en la ingeniería del software y del papel que el modelado desempeña en el desarrollo de las estructuras de los sistemas software. La concepción y desarrollo de estas estructuras y sus componentes serán los temas a desarrollar en el resto de la asignatura.

Teniendo en cuenta que esta es una asignatura de grado, la formación previa de los alumnos y, sobre todo, las limitaciones temporales, se ha considerado 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 para el modelado y documentación del diseño de un sistema. Se persigue que el estudiante comprenda el concepto de arquitectura software, que conozca alguno de los patrones y estilos arquitectónicos de mayor utilización y difusión y, por supuesto, que sea capaz de concebir, desarrollar y documentar el diseño arquitectónico y detallado de un sistema software.

### 1.1 Contextualización

---

Esta asignatura forma parte del módulo “Tecnologías Específicas” dentro 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

---

Fundamentos de Ingeniería del Software, Sistemas distribuidos, Fundamentos de Programación, Programación Orientada a la Integración de Sistemas, Interacción Persona – Computador, Diseño de Bases de Datos.

### 1.3 Prerrequisitos

---

<sup>1</sup> "Swebok - IEEE Computer Society." 2015. 2 Oct. 2015 <<http://www.computer.org/web/swebok>>

"SE2004, Software Engineering 2004 - IEEE Computer Society." 2002. 2 Oct. 2015 <<http://sites.computer.org/ccse/>>

"Computing Curricula 2001 Computer Science - ACM." 2015. 2 Oct. 2015 <[https://www.acm.org/education/curric\\_vols/cc2001.pdf](https://www.acm.org/education/curric_vols/cc2001.pdf)>

Se recomienda que los alumnos hayan cursado con aprovechamiento al menos las siguientes asignaturas del vigente plan de estudios: Fundamentos de Ingeniería del Software, Sistemas Distribuidos, 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 colativo 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
CG1	Capacidad para concebir, redactar, organizar, planificar, desarrollar y firmar proyectos en el ámbito de la ingeniería en informática que tengan por objeto, de acuerdo con los conocimientos adquiridos según lo establecido en las competencias de formación especificadas a continuación en esta sección de la memoria, la concepción, el desarrollo o la explotación de sistemas, servicios y aplicaciones informáticas.
CG3	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.
CG5	Capacidad para concebir, desarrollar y mantener sistemas, servicios y aplicaciones informáticas empleando los métodos de la ingeniería de software como instrumento para el aseguramiento de su calidad, de acuerdo con los conocimientos adquiridos según lo establecido en las competencias de formación especificadas a continuación en esta sección de la memoria
CG6	Capacidad para concebir y desarrollar sistemas o arquitecturas informáticas centralizadas o distribuidas integrando hardware, software y redes de acuerdo con los conocimientos adquiridos según lo establecido en las competencias de formación especificadas a continuación en esta sección de la memoria.
CG9	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 Técnico en 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.
------	--



<b>TI2.</b>	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.
<b>TI3.</b>	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.
<b>IS3.</b>	Capacidad de dar solución a problemas de integración en función de las estrategias, estándares y tecnologías disponibles.
<b>IS4.</b>	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

- Conocer los principios y técnicas del análisis de requisitos y del modelado de sistemas software.
- Conocer los principios y conceptos fundamentales de la arquitectura y el diseño de sistemas software.
- Ser capaces de modelar y diseñar la arquitectura de sistemas TI, ajustándose a un conjunto de requisitos funcionales y no funcionales
- 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		
<b>Total presencial</b>	<b>60</b>	<b>Total no presencial</b>	<b>90</b>



## 5. Bloques temáticos

### Bloque 1: Arquitecturas y diseño de aplicaciones software.

Carga de trabajo en créditos ECTS:

#### a. Contextualización y justificación

La arquitectura y el diseño software son dos materias fundamentales en la formación en ingeniería del software de los graduados en informática. El *Computing Curriculum Software Engineering*<sup>2</sup> 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.

En este bloque se estudiarán los principios, métodos y técnicas que facilitan la concepción y el desarrollo de la arquitectura y el diseño software. Los contenidos estarán enfocados a proporcionar al estudiante una base que le permita adaptarse rápidamente a cualquier plataforma tecnológica. Esta base de conocimientos se complementará con una visión de los métodos y técnicas actuales y de mayor difusión en la industria del software. Se trabajará, desde un enfoque coherente e integrado en las técnicas, métodos y herramientas relacionadas con el modelado y el diseño 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 modelado y el diseño. A continuación, se completa la formación en análisis de requisitos que los alumnos han recibido en la asignatura de “Fundamentos de Ingeniería del Software”. En la tercera parte, se introduce 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 estudio del diseño software desde la perspectiva de desarrollo soportado por patrones, utilizando métodos del diseño orientado al objeto y el lenguaje de modelado UML2.

#### b. Objetivos de aprendizaje

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

<sup>2</sup> <http://sites.computer.org/ccse/>



### c. Contenidos

---

#### Programa de Teoría

1. Modelado y diseño de software.
  - a. Introducción
  - b. Conceptos de la arquitectura y el diseño
  - c. Modelado de sistemas software
  - d. El lenguaje de modelado UML 2.x
2. Especificación y análisis de requisitos.
  - a. Introducción.
  - b. Modelado de la estructura.
  - c. Modelado del comportamiento.
3. Diseño Arquitectónico.
  - a. Introducción
  - b. Estilos y patrones arquitectónicos.
  - c. Arquitecturas de referencia.
4. Diseño de software.
  - a. Introducción.
  - b. Diseño de los sistemas de información. Patrones PoEAA.
  - c. Diseño detallado. Patrones GoF.

### d. Métodos docentes

---

Ver anexo: Métodos docentes.

### e. Plan de trabajo

---

El cronograma de actividades aparecerá en el Aula Virtual o página web asociada a la asignatura,

### f. Evaluación

---

Ver apartado 7. Sistema de calificaciones.



### **g. Bibliografía básica**

---

G. Booch et al. *Object-Oriented Analysis and Design with Applications (3rd Edition)*. Addison Wesley Longman Publishing Co., 2004

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*. Prentice-Hall, 2004.

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

### **h. Bibliografía complementaria**

---

J. Arlow & I. Neustadt. "UML 2". Ed. Anaya Multimedia, 2006.

G. Booch; J. Rumbaugh & I. Jacobson. "The Unified Modeling Language User Guide". 2nd ed. Addison-Wesley, 2005

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. Gamma et al. "Patrones de Diseño. Elementos de software orientado al objeto reutilizable". Addison-Wesley, 2002.

### **i. Recursos complementarios**

---

- » «Core J2EE Patterns! ». [Online]. Available: <http://www.oracle.com/technetwork/java/corej2eepatterns-136991.html>. [Accessed: 21-Jun-2016]
- » «Microsoft Application Architecture Guide, 2nd Edition». [Online]. Available: <http://msdn.microsoft.com/en-us/library/ff650706.aspx>. [Accessed: 21-Jun-2016].
- » «OpenUP». [Online]. Available: <http://epf.eclipse.org/wikis/openup/>. [Accessed: 21-Jun-2016].

## 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		
1. Análisis de requisitos software.	5%	
2. Especificación de la arquitectura software.	5%	
3. Diseño de la capa de lógica de negocio	10%	
4. Diseño de la capa de acceso a datos	10%	
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 análisis y 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 modelado y diseño.
- La evaluación de la parte práctica se realizará sobre el trabajo de laboratorio realizado en grupo a lo largo del curso académico. La evaluación de la parte práctica, al tener carácter de evaluación continua, solamente se puede realizar durante el periodo de actividad docente.
  - En la **convocatoria extraordinaria** se podrá entregar una nueva versión de aquellos trabajos presentados a la convocatoria ordinaria y que no hayan obtenido la calificación mínima establecida. La entrega y posible defensa de la nueva versión del trabajo desarrollado se realizará antes de la fecha de examen.
  - La calificación de estas nuevas versiones de los trabajos elaborados no estará condicionada por la calificación que se hubiese obtenido en la versión anterior.
- Para aprobar la asignatura es necesario haber aprobado la práctica de laboratorio y el examen final.
  - Para la práctica de laboratorio, 5 puntos sobre 10.
  - Para el examen final de teoría, 5 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  $N_{Total} = 0.3 \cdot N_{PL} + 0.7 \cdot N_{EF}$ ;
    - NPL es la nota, sobre 10 puntos, de la práctica de laboratorio.
    - NEF la calificación, sobre 10 puntos, del examen final.

La nota final en la asignatura,  $N_{Final}$ , para un alumno será:

- $N_{Final} = N_{Total}$ , si se cumple la condición 3, ha obtenido la calificación mínima en la práctica de laboratorio y examen final.
- $N_{Final} = \min(3.9, N_{Total})$ , en caso contrario.





## 8. Anexo: Métodos docentes

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

