



1. Identificación

+ Identificación de la Asignatura

Asignatura CONSTRUCCIÓN DE SOFTWARE

Codigo: 08BD **Curso:** 3 **Grupos:** 1

Tipo: Optativa

Modalidad: Presencial

Creditos LRU: 3 Créditos teóricos, 3 Créditos prácticos ,0 Créditos clínicos

Créditos ECTS de la asignatura: 4,8

Estimación del volumen de trabajo del alumno(1Cr.ECTS=25 Horas): 120

Duración: Cuatrimestre

Nº de Alumnos: 0

Nº de grupos de prácticas en clase:0

Nº de grupos de prácticas en laboratorio:1

Nº de grupos de tutorías:1

Idiomas en los que se imparte: Castellano

+ Identificación del Profesorado

Profesor JOAQUIN NICOLAS ROS

Área: LENGUAJES Y SIST.INFORMATICOS

Departamento: INFORMÁTICA Y SISTEMAS

Puesto: PROFESORES TITULARES DE ESCUELAS UNIVERSITARIAS

Despacho: 2.35, Facultad de Informática

Teléfono: 968398525 **E-mail:** jnr@um.es

Páginas Web:

http://dis.um.es/~jnicolas/08BD_CSW.html

Horario de atención al alumnado:

DIA	HORA INICIO	HORA FIN	PERIODO
Martes	16:30 h	18:30 h	Anual
Martes	18:30 h	19:30 h	Anual
Viernes	10:30 h	12:30 h	Anual
Viernes	12:30 h	14:30 h	Anual



2. Presentación

Con Construcción de Software (CSW) se trata de introducir al alumno en la disciplina de la ingeniería del software en general, y más específicamente en el análisis y diseño de software orientado a objetos. En otras palabras, se trata de proporcionar al alumno un conjunto de técnicas, principios y heurísticas de ingeniería del software que le permitan abordar de forma sistemática el proceso de construcción de software, en particular desde el paradigma de desarrollo orientado a objetos. CSW es la única asignatura de la titulación de I. T. Sistemas que trata con cuestiones de ingeniería del software. Dado que un porcentaje considerable de los titulados en sistemas realiza labores de desarrollo durante su práctica profesional, esta asignatura puede ayudar a complementar su formación proporcionando una visión general de las técnicas más aceptadas en la actualidad en el desarrollo de software orientado a objetos. En el ámbito de las asignaturas de la intensificación de Informática Gráfica y Multimedia, CSW juega un papel transversal a todas ellas: el desarrollo de un sistema multimedia se puede considerar como un caso particular de proyecto de desarrollo de software. CSW, en particular, complementaría la visión de la asignatura 'Desarrollo de Aplicaciones Interactivas', que presenta aspectos más cercanos a la implementación. Aquellos alumnos que continúen sus estudios en segundo ciclo estudiarán tres asignaturas troncales relacionadas con ingeniería del software: 'Ingeniería de Requisitos', 'Análisis y Diseño de Software' y 'Gestión de Proyectos Informáticos', todas ellas asignaturas troncales de cuarto curso. En 'Análisis y Diseño de Software', en particular, se reforzarán los contenidos de análisis y diseño orientados a objetos presentados en CSW, se profundizará mucho más en patrones de diseño, y se discutirá el problema de la persistencia en aplicaciones orientadas a objetos. En segundo ciclo, además, hay una intensificación dedicada exclusivamente a la ingeniería del software.

3. Conocimientos Previos

Conocimientos Esenciales

Es altamente recomendable que el alumno haya cursado la asignatura obligatoria del primer cuatrimestre 'Programación Orientada a Objetos' (POO), conociendo así los conceptos esenciales de la programación orientada a objetos: objeto, clase, método, mensaje, herencia, polimorfismo y ligadura dinámica. Se puede considerar que CSW complementa POO subiendo el nivel de abstracción y estudiando el paradigma orientado a objetos desde el punto de vista del análisis y, en menor medida, del diseño.

4. Competencias



+ Competencias Transversales

- 1. Ser capaz de expresarse correctamente en español en su ámbito disciplinar.
- 3. Ser capaz de gestionar la información y el conocimiento en su ámbito disciplinar, incluyendo saber utilizar como usuario las herramientas básicas en TIC.
- 6. Ser capaz de trabajar en equipo y para relacionarse con otras personas del mismo o distinto ámbito profesional.

+ Competencias de Asignatura

Competencia de Asignatura	Competencias Específicas de la Titulación relacionadas
1. Introducir al alumno en la disciplina de la ingeniería del software en general, y más específicamente en el análisis y diseño de software orientado a objetos.	1 2 4
2. Conocer UML (Lenguaje Unificado de Modelado).	1 2 4 8 10
3. Ser capaz de aplicar un proceso basado en UML.	1 2 4 8 10 11
4. Conocer los principales patrones de diseño.	1 2 4 8 10

(Los números corresponden a las competencias específicas de la titulación del Libro Blanco de la Titulación y que se muestran a continuación)

Lista de competencias específicas de la titulación:



1. Aprender de manera autónoma nuevos conocimientos y técnicas adecuados para la concepción, el desarrollo o la explotación de sistemas informáticos.
2. Comunicar de forma efectiva, tanto por escrito como oral, conocimientos, procedimientos, resultados e ideas relacionadas con las TIC y, concretamente de la Informática, conociendo su impacto socioeconómico.
4. Concebir y llevar a cabo proyectos informáticos utilizando los principios y metodologías propios de la ingeniería.
8. Concebir, desarrollar y mantener sistemas y aplicaciones software empleando diversos métodos de ingeniería del software y lenguajes de programación adecuados al tipo de aplicación a desarrollar manteniendo los niveles de calidad exigidos.
10. Proponer, analizar, validar, interpretar, instalar y mantener soluciones informáticas en situaciones reales en diversas áreas de aplicación dentro de una organización.
11. Concebir, desplegar, organizar y gestionar sistemas y servicios informáticos en contextos empresariales o institucionales para mejorar sus procesos de negocio, responsabilizándose y liderando su puesta en marcha y mejora continua, así como valorar su impacto económico y social.

5. Contenidos

Teóricos

Bloque 1: UML

TEMA 1. El Lenguaje Unificado de Modelado, UML

Presentación de UML. Necesidad del modelado. Modelado de casos de uso. Modelado estructural. Paquetes. Vistas de UML. Modelado dinámico. Modelado de flujos de trabajo. Modelado del estado. Modelado de la implementación. Colaboraciones. OCL (Object Constraint Language).

Bloque 2: Modelo de proceso

TEMA 2. Un proceso basado en UML

Definición de proceso. Un proceso sencillo basado en UML. Modelo del negocio: los procesos del negocio. Modelo de requisitos: modelo de casos de uso, modelo conceptual y contratos. Modelo de análisis y patrones GRASP. Modelo de diseño. Implementación. Validación. Una visión general del Proceso Unificado de Rational, RUP.

Bloque 3: Patrones



TEMA 3. Patrones de diseño

Concepto de patrón. Clasificación de patrones: Creación, Estructurales y de Comportamiento.

Estudio de los principales patrones GoF.

Prácticos

Bloque 1: Análisis y diseño orientado a objetos

PRÁCTICA 0. Un CASE para el desarrollo orientado a objetos con UML

Manejo de las principales funciones de una herramienta CASE de soporte a UML, Rational Rose.

PRÁCTICA 1. Aplicación de un proceso basado en UML

Aplicación del proceso basado en UML a la especificación de los requisitos de un caso práctico.

Modelado de casos de uso. Modelado conceptual. Modelado del análisis. Diagramas de secuencia del sistema. Contratos y colaboraciones de las operaciones del sistema. Aplicación de patrones de diseño. Generación de esqueletos de código con la herramienta CASE.

Bloque 2: Trabajo en ingeniería del software

PRÁCTICA 2. Trabajo teórico/práctico en ingeniería del software

Profundización en algún aspecto interesante de la ingeniería del software, de forma complementaria al programa de teoría, realizando un trabajo de corte teórico/práctico: teórico en la revisión bibliográfica y práctico en la aplicación de los conceptos estudiados a casos de estudio accesibles por los alumnos. La temática del trabajo será preferente el análisis y diseño de aspectos relacionados con interfaces de usuario, informática gráfica y multimedia.

6. Metodología y Estimación del volumen de trabajo

Metodología Docente

En las clases de teoría se combinará la metodología expositiva con el estudio de casos y con clases de problemas. Se ofrece la posibilidad a los alumnos de resolver ejercicios tipo examen en la pizarra. Se dice que 'se ofrece la posibilidad' porque estos ejercicios serán resueltos en la pizarra siempre por parte de los propios alumnos, de forma interactiva con el profesor y con el resto de compañeros. De esta manera los contenidos de la asignatura se introducirán de forma práctica, a través de estudio de casos y de problemas resueltos en el aula. En prácticas la Práctica 0 se realiza en la modalidad de laboratorio cerrado, es decir, con el profesor y los alumnos en el laboratorio de prácticas, mientras que



la Práctica 1 se realiza en la modalidad de laboratorio abierto, en la cual los alumnos trabajan con libertad en un enunciado de prácticas siguiendo un control exhaustivo por parte del profesor (tres entrevistas intermedias y una entrevista final), con el fin de que los alumnos obtengan continua retroalimentación. Para facilitar la labor de los alumnos, se intentará que el enunciado del caso práctico de la Práctica 1 (que normalmente se extrae de un caso de estudio real) se comparta con la asignatura de Desarrollo de Aplicaciones Interactivas, de manera que cada asignatura se centre más en unos aspectos del desarrollo que en otros y los alumnos sólo deban conocer en profundidad un caso de estudio. La Práctica 2 se realiza también en la modalidad de laboratorio abierto, y en ella los alumnos deben realizar un trabajo de revisión bibliográfica, normalmente en inglés, de aplicación de los conocimientos adquiridos a ejemplos reales accesibles por ellos, y de síntesis de los tales conocimientos y ejemplos en un trabajo escrito.

Estimación de volumen de trabajo del estudiante (ECTS)

Volumen de Trabajo

Actividad	Hora presencial	Factor	Trabajo Personal	Volumen de Trabajo
CLASES TEÓRICO/PRÁCTICAS				
Clases de problemas	14	1	14	28
CLASES TEÓRICAS				
Metodología expositiva con presentación de casos	16	1	16	32
CLASES PRÁCTICAS				
Prácticas presenciales	4		0	4
Prácticas no presenciales	52		0	52
TUTORÍAS ECTS				
Examen y controles de prácticas	4		0	4
REPASO FINAL				
			0	0
Total				120
Relación: Horas de trabajo / ECTS	120 / 120 = 25			

7. Cronograma



Bloque temático	Temas	Título o contenidos	CT	CP	T	SubTotal	Fechas previstas
Créditos Teóricos							
1	1	El Lenguaje Unificado de Modelado, UML	6	6		12	
2	2	Un proceso basado en UML	6	6		12	
3	3	Patrones de diseño	4	2		6	
Créditos Prácticos							
1	0	Un CASE para el desarrollo orientado a objetos con UML	4			4	
	1	Aplicación de un proceso basado en UML		36	3	39	
2	2	Trabajo teórico/práctico en ingeniería del software		16		16	
		Evaluación final		1		1	
		Totales	21	66	3	90	


8. Evaluación

+ Evaluación del aprendizaje.

Instrumentos	Criterios de Evaluación	Ponderación
Corrección de documentación y entrevistas de prácticas.	Las prácticas se evalúan a través de la Práctica 1 (70% de la nota de prácticas) y de la Práctica 2 (30%). La Práctica 1 se evalúa mediante la corrección de tres entregas parciales y una final, seguidas de sus correspondientes entrevistas. La Práctica 2 se evalúa mediante la corrección del trabajo y una entrevista final. La presentación del trabajo, la redacción, la adecuación de los contenidos y la aplicación a casos prácticos accesibles por los alumnos serán valoradas especialmente. Las prácticas de la asignatura se organizarán en grupos de dos personas, pero las notas son individuales.	100%
Participación en clase.	La participación en clase, y especialmente en las clases de problemas, será bonificada en la nota final	+1 punto extra.



Instrumentos	Criterios de Evaluación	Ponderación
	con hasta un punto extra.	





 Evaluación de la docencia.

El profesor realiza una encuesta al final del cuatrimestre para detectar posibles mejoras y cambios de enfoque, y posibles problemas durante el desarrollo de la asignatura que le hayan pasado desapercibidos.

9. Bibliografía

Grupo 1

Bibliografía Básica:

-  Booch, Grady--El lenguaje unificado de modelado(1999)
-  Larman, Craig--UML y patrones : introducción al análisis y diseño orientado(1999)
-  Larman, Craig--UML y patrones : una introducción al análisis y diseño orien(2002)
-  Booch, Grady -- El lenguaje unificado de modelado; Addison Wesley (2006)

Bibliografía Complementaria: