Guía docente de la asignatura

Asignatura	Programación Orientada a la Integración de Sistemas		
Materia	TECNOLOGÍAS SOFTWARE		
Módulo			
Titulación	Grado en INGENIERÍA INFORMÁTICA DE SISTEMAS		
Plan	464	**Código**	45261
Periodo de impartición	S5	**Tipo/Carácter**	OB
Nivel/Ciclo	Grado	**Curso**	1
Créditos ECTS	6		
Lengua en que se imparte	Español		
Profesor/es responsable/s	Yania Crespo González-Carvajal		
Datos de contacto (E-mail, teléfono…)	TELÉFONO: 983 423000 ext. 5695 E-MAIL: yania@infor.uva.es		
Horario de tutorías	Véase www.uva.es → Centros → Campus de Valladolid → Escuela Técnica Superior de Ingeniería Informática → Tutorías		
Departamento	Informática		

1. Situación / Sentido de la Asignatura

1.1 Contextualización

Esta asignatura se encuentra situada en el tercer curso de Ingeniería Informática de Sistemas, junto a otras que conforman la materia “Tecnologías Software” (ver apartado 1.2), proporcionando enfoque conjunto sobre las tecnologías de desarrollo de software. El objetivo de esta asignatura es fijar de una forma clara los conceptos básicos necesarios para saber aplicar técnicas y herramientas útiles para la integración del software y el intercambio de datos, respetando los principios de calidad y dentro del paradigma Orientado a Objetos

1.2 Relación con otras materias

La asignatura está planteada como una parte de las disciplinas que componen la ingeniería de software y que se desarrollan en las asignaturas de la materia “Tecnologías Software”. Está situada en el primer semestre, mientras que la asignatura Diseño, Integración y Adaptación de Software lo está en el segundo semestre de modo que ambas se coordinarán y complementarán.

1.3 Prerrequisitos

Aunque no se han establecido prerrequisitos, es recomendable que el alumno haya aprobado la asignatura de Fundamentos de Programación, y Fundamentos de Ingeniería del Software, ambas de primer curso. Además es recomendable disponer de un nivel de inglés que permita al estudiante leer bibliografía de consulta.

2. Competencias

2.1 Generales

Código	Descripción
G03	Capacidad de análisis y síntesis
G04	Capacidad de organizar y planificar
G05	Comunicación oral y escrita en la lengua propia
G06	Conocimiento de una segunda lengua (preferentemente inglés)
G08	Habilidades de gestión de la información
G09	Resolución de problemas
G10	Toma de decisiones
G11	Capacidad crítica y autocrítica
G12	Trabajo en equipo
G14	Responsabilidad y compromiso ético
G15	Liderazgo
G16	Capacidad de aplicar los conocimientos en la práctica
G17	Habilidades de investigación
G18	Capacidad de aprender
G19	Capacidad de adaptarse a nuevas situaciones
G20	Capacidad de generar nuevas ideas
G21	Habilidad para trabajar de forma autónoma

2.2 Específicas

Código	Descripción
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
TI6	Capacidad de concebir sistemas, aplicaciones y servicios basados en tecnologías de red, incluyendo Internet, web, comercio electrónico, multimedia, servicios interactivos y computación móvil
SI1	Capacidad de integrar soluciones de Tecnologías de la Información y las Comunicaciones y procesos empresariales para satisfacer las necesidades de información de las organizaciones, permitiéndoles alcanzar sus objetivos de forma efectiva y eficiente, dándoles así ventajas competitivas.

3. Objetivos

Código	Descripción
TI2.1	Comprender el paradigma de diseño basado en patrones y entender su importancia en integración de aplicaciones, Identificar y saber aplicar los patrones de diseño más importantes.
TI2.2	Comprender y saber aplicar los principios de la programación bajo contrato en el diseño y programación de aplicaciones seguras y fáciles de integrar y evolucionar.
TI2.3	Ser capaz de aplicar estrategias al caso especial del diseño de interfaces de programación de aplicaciones.
TI6.1	Saber programar aplicaciones sencillas basadas en servicios.
SI1.1	Conocer y saber usar algún API de interés en el mercado.
SI1.2	Conocer y saber usar los estándares de representación de datos para intercambio entre aplicaciones.

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	60
Clases prácticas de aula (A)		Estudio y trabajo autónomo grupal	30
Laboratorios (L)	26		
Prácticas externas, clínicas o de campo			
Seminarios (S)			
Tutorías grupales (TG)			
Evaluación	4		
Total presencial	**60**	**Total no presencial**	**90**

5. Bloques temáticos

Bloque 1: Programación orientada a objetos y programación bajo contrato

Carga de trabajo en créditos ECTS: 2,4

a. Contextualización y justificación

Tras introducir los principios del paradigma Orientado a Objetos (OO), abordaremos los conceptos básicos utilizados en el paradigma: Clase y Objeto. Posteriormente se tratarán los principales mecanismos de OO como son la Genericidad y la Herencia, imprescindibles para la comprensión y elaboración de sistemas Orientados a Objetos de dificultad moderada. Además se tratarán las técnicas de programación bajo contrato, adecuadas para la verificación del software OO. Por último se abordará el uso de APIs.

b. Objetivos de aprendizaje

Código	Descripción
TI2.2	Comprender y saber aplicar los principios de la programación bajo contrato en el diseño y programación de aplicaciones seguras y fáciles de integrar y evolucionar.
TI2.3	Ser capaz de aplicar estrategias de programación al caso especial del diseño de interfaces de programación de aplicaciones.
TI6.1	Saber programar aplicaciones sencillas basadas en servicios.
SI1.1	Conocer y saber usar algún API de interés en el mercado.

c. Contenidos

TEMA 1: Clases y Objetos

1.1 Presentación

1.2 Principios de la OO

1.3 Clases

1.4 Objetos

TEMA 2: Herencia y Genericidad

2.1 Genericidad

2.2 Herencia

2.3 Polimorfismo

2.4 Ligadura dinámica

2.5 Bibliotecas y Frameworks

d. Métodos docentes

Ver anexo 8 al final

e. Plan de trabajo

Ver anexos 6 y 9 al final

f. Evaluación

Ver anexo 7 al final

g. Bibliografía básica

[Meyer] Meyer B. (2002) Construcción de software orientado a objetos, 2ª. ed., Prentice-Hall, ISBN 84-8322-040-7

[Eckel]"Bruce Eckel, *Piensa en Java* 4o Ed. Prentice-Hall, 2007 ISBN: 9788489660342

h. Bibliografía complementaria

[Meyer] Meyer B. (2009) Touch of class: learning to program well with objects and contracts Springer, cop. ISBN

[Deitel] Harvey Deitel, *Cómo programar en Java*, Pearson 2008, ISBN: 9789702611905

i. Recursos necesarios

Entorno de Desarrollo Integrado instalado en los laboratorios docentes y descargable a través la página web.

Bloque 2: Representación e intercambio de datos

Carga de trabajo en créditos ECTS: 1,6

a. Contextualización y justificación

En este bloque se presenta XML como estándar de intercambio de datos. Se describirán las características de los DTD y esquemas cómo XSL, XSLT y XPath que se utilizan para transformar datos. Se explicará cómo XML y DOM (modelo de objeto del documento) se usan para la integración y el intercambio de datos entre sistemas. En las clases de laboratorio se plantearán ejemplos de desarrollo de programas que utilicen SAX o DOM para analizar un documento XML obtenido de un sistema e integrar la misma información en otro sistema con otro formato.

b. Objetivos de aprendizaje

Código	Descripción
SI1.2	Conocer y saber usar los estándares de representación de datos para intercambio entre aplicaciones.

c. Contenidos

TEMA 3: Representación de datos

3.1 XML, DTD, esquemas XML

3.2 Validación de DTD y esquemas

3.3 XSL, XSLT y XPath

3.4 Análisis de documentos XML: SAX y DOM

TEMA 4: Intercambio de datos con XML

4.1 Serialización con XML

4.2 Persistencia con XML

4.3 APIs de Java para SAX y DOM

d. Métodos docentes

Ver anexo 8 al final

e. Plan de trabajo

Ver anexos 6 y 9 al final

f. Evaluación

Ver anexo 7 al final

g. Bibliografía básica

[Parsons] Parsons, David. Desarrollo de Aplicaciones Web Dinámicas Con Xml y Java. ANAYA, 2009.

h. Bibliografía complementaria

[Garcia] Garcia Acera, Miguel Angel. (2011) XML.edicion 2012. Anaya..

[Oracle] .Oracle Java API for XML Processing http://docs.oracle.com/javase/tutorial/jaxp/ (última consulta 16/7/2013)

i. Recursos necesarios

Entorno de Desarrollo Integrado instalado en los laboratorios docentes y descargable a través la página web.

Bloque 3: Patrones de diseño

Carga de trabajo en créditos ECTS: 2

a. Contextualización y justificación

En este bloque se aborda el problema de utilizar soluciones generales a problemas generales, evitando repetir lo ya hecho y planteando la solución de una forma estándar. Los patrones de diseño de software son soluciones reutilizables de problemas recurrentes que aparecen durante el proceso de diseño de software orientado a objetos. Además se utilizarán dichos patrones para resolver cierto tipo de problemas planteados en los bloques anteriores, especialmente en las conexiones cliente-servidor.

b. Objetivos de aprendizaje

Código	Descripción
TI2.1	Comprender el paradigma de diseño basado en patrones y entender su importancia en integración de aplicaciones, Identificar y saber aplicar los patrones de diseño más importantes.

c. Contenidos

TEMA 5: Patrones de diseño

5.1 Introducción, patrón Singleton
5.2 Patrón Iterador
5.3 Patrón Observador
5.4 Patrón Adaptador
5.5 Patrón Compuesto
5.6 Patrón Factory Method

d. Métodos docentes

Ver anexo 8 al final

e. Plan de trabajo

Ver anexos 6 y 9 al final

f. Evaluación

Ver anexo 7 al final

g. Bibliografía básica

[Gamma] Gamma E. et al. (2002) Patrones de diseño. Addison Wesley.

h. Bibliografía complementaria

[Grand] Mark Grand (2002). Patterns in Java: A Catalog of Reusable Design Patterns Illustrated with UML.

i. Recursos necesarios

Herramientas de programación instaladas en los laboratorios docentes y descargables a partir la página web.

6. Temporalización (por bloques temáticos)

BLOQUE TEMÁTICO	CARGA ECTS	PERIODO PREVISTO DE DESARROLLO
Bloque 1: Programación Orientada a Objetos y Programación bajo Contrato	2,4 ECTS	Semanas 1 a 6
Bloque 2: Representación e intercambio de datos	1,6 ECTS	Semanas 7 a 10
Bloque 3: Patrones de diseño	2,0 ECTS	Semanas 11 a 15

7. Sistema de calificaciones – Tabla resumen

INSTRUMENTO/PROCEDIMIENTO	PESO EN LA NOTA FINAL	OBSERVACIONES
Examen tipo test sobre los temas 1 y 2	10,00%	Aproximadamente semana 7
Examen tipo test sobre el tema 3 y 4	10,00%	Aproximadamente semana 12
Examen tipo test sobre el tema 5	10,00%	Aproximadamente semana 15
Entrega de la primera práctica	10,00%	Aproximadamente semana 7
Entrega de la segunda práctica	10,00%	Aproximadamente semana 15
Entrega de la tercera práctica	10,00%	
Examen de problemas	40,00%	Periodo de exámenes (ordinario y extraordinario).

CRITERIOS DE CALIFICACIÓN

Convocatoria ordinaria: Cuando la nota del examen sea igual o superior a 4/10 se hará la suma ponderada de los cuestionarios (30%), prácticas en parejas (30%) y examen (40%). En otro caso, la calificación será la menor de las puntuaciones entre 4,5 y la suma ponderada:

- **Si nota(examen) >= 4**, Nota final= Suma ponderada
- **Si nota(examen) < 4**, Nota final= mínimo(Suma ponderada; 4,5)

Convocatoria extraordinaria: Para la convocatoria extraordinaria se mantendrá la ponderación de las calificaciones de la convocatoria ordinaria con las siguientes puntualizaciones:

- Obligatoriamente se realizará el examen de problemas
- Opcionalmente se realizará un examen de tipo test sobre los conceptos teóricos de la asignatura. En caso de no optar por la realización de ese test, la calificación considerada en ese apartado será la obtenida en la convocatoria ordinaria
- Si no se han entregado las prácticas de la asignatura se podrá optar a una entrega extraordinaria de las mismas, en las mismas condiciones de la convocatoria ordinaria.

8. Anexo: Métodos docentes

Actividad	Metodología
Clase de teoría	Clase magistral participativa Estudio de casos en aula Resolución de problemas
Clase práctica	Clase magistral participativa Realización en grupos de dos personas de dos pequeños sistemas de software que utilicen adecuadamente las técnicas presentadas en la asignatura

9. Anexo: Cronograma de actividades previstas

Semana	Fecha	Teoría	Prácticas	Entrega Trabajos	Evaluación
1		Tema 1			
2		Tema 1	Clases y objetos		
3		Tema 1	Clases y objetos		
4		Tema 2	Genericidad y Herencia		
5		Tema 2	Genericidad y Herencia		
6		Tema 2	Genericidad y Herencia	Práctica 1	
7		Tema 3	Representación de datos		Test 1
8		Tema 3	Representación de datos		
9		Tema 4	Intercambio de datos con XML		
10		Tema4	Intercambio de datos con XML		
11		Tema 4	Intercambio de datos con XML	Practica 2	
12		Tema 5	Patrones de diseño		Test 2
13		Tema 5	Patrones de diseño		
14		Tema 5	Patrones de diseño		
15		Tema 5	Patrones de diseño	Práctica 3	Test 3

Nota: Esta tabla es de carácter orientativo. Las fechas concretas de realización de los cuestionarios y entregas de prácticas se anunciarán a través del aula virtual.