Guía docente de la asignatura

Asignatura	PROGRAMACIÓN ORIENTADA A OBJETOS		
Materia	DESARROLLO DE SOFTWARE		
Módulo			
Titulación	GRADO EN INGENIERÍA INFORMÁTICA (463)		
Plan	463	**Código**	45204
Periodo de impartición	1er. CUATRIMESTRE	**Tipo/Carácter**	OBLIGATORIA
Nivel/Ciclo	GRADO	**Curso**	3º
Créditos ECTS	6 ECTS		
Lengua en que se imparte	CASTELLANO		
Profesor/es responsable/s	Félix Prieto Arambillet		
Datos de contacto (E-mail, teléfono…)	TELÉFONO: 983 423000 ext. 5617 E-MAIL: felix@infor.uva.es		
Horario de tutorías	Véase www.uva.es → Centros → Campus de Valladolid → Escuela Técnica Superior de Ingeniería Informática → Tutorías		
Departamento	DEPARTAMTENTO DE INFORMÁTICA		

1. Situación / Sentido de la Asignatura

1.1 Contextualización

En la actualidad, la mayoría del desarrollo de software se realiza desde el paradigma Orientado a Objetos. El en contexto de la materia de desarrollo de software, y con los conocimientos previos adquiridos en otras asignaturas de las materias de Fundamentos Básicos de Informática y Entornos Software, el objetivo de esta asignatura es fijar de una forma clara los conceptos, técnicas y herramientas necesarias para realizar un desarrollo de software de calidad dentro del paradigma Orientado a Objetos.

1.2 Relación con otras materias

La asignatura se basa y amplía conceptos presentados en algunas asignaturas de la materia Fundamentos Básicos de Informática, como Fundamentos de Programación, o de la materia Entornos Software como Paradigmas de Programación, Fundamentos de Ingeniería de Software o Estructuras de Datos y Algoritmos.

1.3 Prerrequisitos

Aunque no se han establecido prerrequisitos, es recomendable que el alumno posea conocimientos básicos de programación, en particular haber cursado con aprovechamiento las asignaturas de Fundamentos de Programación, Paradigmas de Programación y Fundamentos de Ingeniería del Software.

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	Comunicación oral y escrita en la lengua propia
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
G22	Diseño y gestión de proyectos

2.2 Específicas

Código	Descripción
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.
CI8	Capacidad para analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados.

3. Objetivos

Código	Descripción
IS3.1	Comprender el paradigma de la programación orientada a objeto, su fundamentación teórica y las pautas de su aplicación práctica.
IS4.1	Emplear correctamente el concepto de objeto y de clase, las relaciones de genericidad y herencia y los mecanismos asociados al polimorfismo en la construcción de programas correctos y fáciles de mantener.
CI8.1	Entender los fundamentos de programación bajo contrato y las ventajas que aporta.
CI8.2	Ser capaz de proyectar y realizar pruebas de programas en entornos específicos de objetos.
IS3.2	Saber aplicar bibliotecas y frameworks de objetos al desarrollo de aplicaciones

4. Tabla de dedicación del estudiante a la asignatura

ACTIVIDADES PRESENCIALES	HORAS	ACTIVIDADES NO PRESENCIALES	HORAS
Clases teórico-prácticas (T/M)	**28**	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 (fuera del periodo oficial de exámenes)	**6**		
Total presencial	**60**	**Total no presencial**	**90**

5. Bloques temáticos

Bloque 1: Clases y objetos

Carga de trabajo en créditos ECTS: 1.3

a. Contextualización y justificación

Tras introducir los principios del paradigma Orientado a Objetos, abordaremos los conceptos básicos utilizados en el paradigma: Clase y Objeto

b. Objetivos de aprendizaje

Código	Descripción
IS3.1	Comprender el paradigma de la programación orientada a objeto, su fundamentación teórica y las pautas de su aplicación práctica.
IS4.1	Emplear correctamente el concepto de objeto y de clase, las relaciones de genericidad y herencia y los mecanismos asociados al polimorfismo en la construcción de programas correctos y fáciles de mantener.

c. Contenidos

TEMA 1: Clases y Objetos

1.1 Presentación

1.2 Principios de la OO

1.3 Clases

1.4 Objetos

d. Métodos docentes

Ver Anexo en el punto 8 de esta guía

e. Plan de trabajo

Ver cronograma en el punto 9 de esta guía.

f. Evaluación

Ver tabla y criterios en el punto 7 de esta guía.

g. Bibliografía básica

- Bertrand Meyer, *Construcción de software orientado a objetos*, 2ª. ed., Prentice-Hall, 2002 ISBN 8483220407
- Bruce Eckel, *Piensa en Java* 4º Ed. Prentice-Hall, 2007 ISBN: 9788489660342

.

h. Bibliografía complementaria

- Bertrand Meyer, *Touch of class: learning to program well with objects and contracts,* Springer, 2009. ISBN 9783540921448
- Harvey Deitel, *Cómo programar en Java*, Pearson 2008, ISBN: 9789702611905

i. Recursos necesarios

Herramientas de programación instaladas en los laboratorios docentes y descargables a partir del aula virtual y/o la página web de la asignatura.

Bloque 2: Genericidad y Herencia

Carga de trabajo en créditos ECTS: 2.0

a. Contextualización y justificación

En el segundo bloque de la asignatura se abordan conceptos más avanzados del paradigma Orientado a Objetos, como son la Genericidad y la Herencia, imprescindibles para la elaboración de sistemas Orientados a Objetos de dificultad moderada.

b. Objetivos de aprendizaje

Código	Descripción
IS3.1	Comprender el paradigma de la programación orientada a objeto, su fundamentación teórica y las pautas de su aplicación práctica.
IS4.1	Emplear correctamente el concepto de objeto y de clase, las relaciones de genericidad y herencia y los mecanismos asociados al polimorfismo en la construcción de programas correctos y fáciles de mantener.
IS3.2	Saber aplicar bibliotecas y frameworks de objetos al desarrollo de aplicaciones

c. Contenidos

TEMA 2: Genericidad y Herencia

2.1 Genericidad
2.2 Herencia
2.3 Polimorfismo
2.4 Ligadura dinámica
2.5 Bibliotecas y Frameworks

d. Métodos docentes

Ver Anexo en el punto 8 de esta guía

e. Plan de trabajo

Ver cronograma en el punto 9 de esta guía.

f. Evaluación

Ver tabla y criterios en el punto 7 de esta guía.

g. Bibliografía básica

- Bertrand Meyer, *Construcción de software orientado a objetos*, 2ª. ed., Prentice-Hall, 2002 ISBN 84-8322-040-7
- Bruce Eckel, *Piensa en Java* 4º Ed. Prentice-Hall, 2007 ISBN: 9788489660342

h. Bibliografía complementaria

- Bertrand Meyer, *Touch of class: learning to program well with objects and contracts,* Springer, 2009. ISBN 9783540921448
- Harvey Deitel, *Cómo programar en Java*, Pearson 2008, ISBN: 9789702611905

i. Recursos necesarios

Bloque 3: Verificación y Validación

Carga de trabajo en créditos ECTS: 2.7

a. Contextualización y justificación

La aplicación de técnicas Orientadas a Objetos no excluye la necesidad de garantizar la calidad del software construido, más bien al contrario, puesto que las técnicas Orientadas a Objetos permiten abordar problemas de programación más complejos, es más necesario utilizar estrategias que garanticen en la medida de lo posible la calidad del software construido. En este contexto, el tema tres aborda las técnicas de diseño bajo contrato, apropiadas para la verificación de Software Orientado a Objetos, y la forma de adaptar las técnicas de Validación a este paradigma de programación.

b. Objetivos de aprendizaje

Código	Descripción
CI8.1	Entender los fundamentos de programación bajo contrato y las ventajas que aporta.
CI8.2	Ser capaz de proyectar y realizar pruebas de programas en entornos específicos de objetos.

c. Contenidos

TEMA 3: Verificación y validación

3.1 Conceptos básicos de la verificación formal
3.2 Demostraciones elementales de verificación formal
3.3 Asertos en el código: cofoja
3.4 Pruebas en OO, introducción
3.5 Pruebas de métodos y clases
3.6 Pruebas basadas en el estado
3.7 Modalidades de clases
3.8 Patrones de prueba
3.9 Uso de excepciones para la implementación de pruebas

d. Métodos docentes

Ver Anexo en el punto 8 de esta guía

e. Plan de trabajo

Ver cronograma en el punto 9 de esta guía.

f. Evaluación

Ver tabla y criterios en el punto 7 de esta guía.

g. Bibliografía básica

- Bertrand Meyer, *Construcción de software orientado a objetos*, 2ª. ed., Prentice-Hall, 2002 ISBN 8483220407
- Bruce Eckel, *Piensa en Java* 4º Ed. Prentice-Hall, 2007 ISBN: 9788489660342
- Myers, Glenford J., *The art of software testing* , John Wiley & Sons 2004 ISBN: 0471469122
- Wirth, Niklaus, *Introducción a la programación sistemática,* El Ateneo, 1986, ISBN: 9500252341

h. Bibliografía complementaria

- Bertrand Meyer, *Touch of class: learning to program well with objects and contracts,* Springer, 2009. ISBN 9783540921448
- Harvey Deitel, *Cómo programar en Java*, Pearson 2008, ISBN: 9789702611905
- Binder, Robert V., Testing *object-oriented systems : models, patterns, and tools*, Addison-Wesley, 2000

i. Recursos necesarios

Herramientas de programación instaladas en los laboratorios docentes y descargables a partir del aula virtual y/o la página web de la asignatura.

6. Temporalización (por bloques temáticos)

El número de semanas de un cuatrimestre son 15. Ejemplo:

BLOQUE TEMÁTICO	CARGA ECTS	PERIODO PREVISTO DE DESARROLLO
Bloque 1: Clases y objetos	1.3 ECTS	Semanas 1 a 3
Bloque 2: Genericidad y herencia	2.0 ECTS	Semanas 4 a 9
Bloque 3: Verificación y validación	2.7 ECTS	Semanas 10 a 15

7. Sistema de calificaciones – Tabla resumen

INSTRUMENTO/PROCEDIMIENTO	PESO EN LA NOTA FINAL	OBSERVACIONES
Examen tipo test sobre el tema 1	10%	
Examen tipo test sobre el tema 2	10%	
Examen tipo test sobre el tema 3	10%	
Entrega de la primera práctica	15%	
Entrega de la segunda práctica	15%	
Examen de problemas	40%	

CRITERIOS DE CALIFICACIÓN

- **Convocatoria ordinaria:** Suma ponderada de los cuestionarios (30%), prácticas en parejas 30%) y examen (40%), debiendo obtener una suma igual o mayor a 5. Será necesaria una calificación mínima de 4/10 en el examen.
  - 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 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	Clases y objetos		
5		Tema 2			Test 1
6		Tema 2	Genericidad, Herencia,…		
7		Tema 2	Genericidad, Herencia,…		
8		Tema 2	Genericidad, Herencia,…	Práctica 1	
9		Tema 2	Genericidad, Herencia,…		
10		Tema 3	Genericidad, Herencia,…		Test 2
11		Tema 3	Genericidad, Herencia,…		
12		Tema 3	Genericidad, Herencia,…		
13		Tema 3	Verificación y Validación		
14		Tema 3	Verificación y Validación		
15		Tema 3		Práctica 2	Test 3

Nota: Las fechas indicadas en esta tabla para pruebas y entregas son aproximadas. Tanto pruebas como entregas serán convocadas con la suficiente antelación mediante la plataforma Moodle de la escuela.