

Guía docente de la asignatura

| | | | |
|------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------|-----------------------------|
| Asignatura | PROGRAMACIÓN ORIENTADA A OBJETOS | | |
| Materia | ENTORNO DE SOFTWARE | | |
| Módulo | COMUNES A LA INFORMÁTICA | | |
| Titulación | GRADO EN INGENIERÍA INFORMÁTICA | | |
| Plan | 545 | Código | 46914 |
| Periodo de impartición | 1 ^{er} . CUATRIMESTRE | Tipo/Carácter | COMPLEMENTOS DE INFORMÁTICA |
| Nivel/Ciclo | GRADO | Curso | 2º |
| Créditos ECTS | 6 ECTS | | |
| Lengua en que se imparte | CASTELLANO | | |
| Profesor/es responsable/s | Félix Prieto (Coordinador de teoría), Yania Crespo (Coordinadora de prácticas) | | |
| Datos de contacto (E-mail, teléfono...) | TELÉFONO: 983 423000 ext. 5617, ext. 5695 E-MAIL: felix@infor.uva.es , yania@infor.uva.es | | |
| Horario de tutorías | Véase www.uva.es → Centros → Campus de Valladolid → Escuela de Ingeniería Informática → Tutorías | | |
| Departamento | DEPARTAMENTO 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 Entorno Software, y con los conocimientos previos adquiridos en otras asignaturas de las materias de Fundamentos Básicos de Informática y Entorno 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 y Paradigmas de Programación.

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 la concepción, el desarrollo o la explotación de sistemas, servicios y aplicaciones informáticas. |
| CG2 | Capacidad para dirigir las actividades objeto de los proyectos del ámbito de la informática. |
| 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. |
| CG6 | Capacidad para concebir y desarrollar sistemas o arquitecturas informáticas centralizadas o distribuidas integrando hardware, software y redes. |
| CG10 | Conocimientos para la realización de mediciones, cálculos, valoraciones, tasaciones, peritaciones, estudios, informes, planificación de tareas y otros trabajos análogos de informática. |

2.2 Específicas

| Código | Descripción |
|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 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. |
| CI5 | Conocimiento, administración y mantenimiento de sistemas, servicios y aplicaciones informáticas. |

3. Objetivos

| Código | Descripción |
|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| RA1 | Comprender el paradigma de la programación orientada a objeto, su fundamentación teórica y las pautas de su aplicación práctica. |
| RA2 | 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. |
| RA3 | Entender los fundamentos de programación bajo contrato y las ventajas que aporta. |
| RA4 | Ser capaz de proyectar y realizar pruebas de programas en entornos específicos de objetos. |
| RA5 | 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: Programación orientada a objetos**Carga de trabajo en créditos ECTS: **a. Contextualización y justificación**

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

Posteriormente 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.

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 |
|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| RA1 | Comprender el paradigma de la programación orientada a objeto, su fundamentación teórica y las pautas de su aplicación práctica. |
| RA2 | 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. |
| RA3 | Entender los fundamentos de programación bajo contrato y las ventajas que aporta. |
| RA4 | Ser capaz de proyectar y realizar pruebas de programas en entornos específicos de objetos. |
| RA5 | Saber aplicar bibliotecas y frameworks de objetos al desarrollo de aplicaciones |

c. Contenidos

TEMA 1: Clases y Objetos

- 1.1 Introducción a la OO
- 1.2 Principios de la OO
- 1.3 Clases
- 1.4 Objetos

TEMA 2: Pruebas

- 2.1 Pruebas en OO
- 2.2 Tratamiento de excepciones
- 2.3 Pruebas de caja negra
- 2.4 Contratos y tratamiento de excepciones

TEMA 3: Genericidad y Herencia

- 3.1 Genericidad
- 3.2 Herencia
- 3.3 Polimorfismo
- 3.4 Ligadura dinámica
- 3.5 Bibliotecas y Frameworks

TEMA 4: Verificación formal

- 4.1 Conceptos básicos de la verificación formal
- 4.2 Demostraciones elementales de verificación formal
- 4.3 Asertos en el código: cofoja

d. Métodos docentes

Ver Anexo en el punto 8 de esta guía

e. Plan de trabajo

El plan de trabajo será publicado en el aula virtual de la escuela una vez coordinado con el resto de las asignaturas del curso

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* 4ª 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)

| BLOQUE TEMÁTICO | CARGA ECTS | PERIODO PREVISTO DE DESARROLLO |
|--------------------------------------------|------------|--------------------------------|
| Bloque 1: Programación Orientada a Objetos | 6 ECTS | Semanas 1 a 15 |

7. Sistema de calificaciones – Tabla resumen

| INSTRUMENTO/PROCEDIMIENTO | PESO EN LA NOTA FINAL | OBSERVACIONES |
|----------------------------------|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Examen tipo test sobre el tema 1 | 10% | Realizadas durante alguna de las sesiones de teoría |
| Examen tipo test sobre el tema 2 | 10% | |
| Examen tipo test sobre el tema 3 | 10% | |
| Entrega de la primera práctica | 15% | Resolución de problemas prácticos realizada en grupos de dos o tres personas (las entregas individuales sólo serán admitidas en casos excepcionales debidamente justificados y con autorización expresa). |
| 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 $\text{nota}(\text{examen}) \geq 4$, $\text{Nota final} = \text{Suma ponderada}$
 - Si $\text{nota}(\text{examen}) < 4$, $\text{Nota final} = \text{mínimo}(\text{Suma ponderada}; 4,5)$
- **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 | <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">• Clase magistral participativa• Realización en grupos de dos personas de dos pequeños sistemas de software que utilicen adecuadamente las técnicas presentadas en la asignatura |

