

# Guía Docente

## DATOS DE IDENTIFICACIÓN

Titulación:	Ingeniería Informática		
Rama de Conocimiento:	Ingeniería y Arquitectura		
Facultad/Escuela:	Escuela Politécnica Superior		
Asignatura:	Programación Orientada a Objetos		
Tipo:	Obligatoria	Créditos ECTS:	6
Curso:	1	Código:	3621
Periodo docente:	Segundo semestre		
Materia:	Diseño y Desarrollo de Software		
Módulo:	Común a la Rama de Informática		
Tipo de enseñanza:	Presencial		
Idioma:	Castellano		
Total de horas de dedicación del alumno:	150		

Equipo Docente	Correo Electrónico
Carlos Montero Llamas	c.montero@ufv.es

## DESCRIPCIÓN DE LA ASIGNATURA

La asignatura Programación Orientada a Objetos permite conocer la idoneidad del paradigma computacional orientado a objetos y aplicarlo para mejorar la calidad del software, en concreto para favorecer la reutilización, extensibilidad y fiabilidad de los programas de forma independiente al lenguaje.

La asignatura se compone de dos grandes bloques conceptuales, uno teórico y otro de carácter más práctico, orientados a conocer el paradigma de programación orientada a objetos, sus principales ventajas y la construcción de programas, desde el diseño y definición de las clases y objetos que los forman hasta su implementación y posterior depuración con un lenguaje y entorno de desarrollo determinado.

Esta asignatura corresponde al módulo Común a la Rama de Informática y, dentro de éste, a la materia Diseño y Desarrollo de Software. Se imparte en el segundo semestre del primer curso de los estudios de Grado en Ingeniería Informática, y requiere de una dedicación de 150 horas por parte del alumno.

En el bloque teórico se ofrecerán los conocimientos necesarios para entender qué es la orientación a objetos y cómo debe programarse en este paradigma. Estos conocimientos versan sobre: clases y objetos, abstracción y clasificación, estructuras de objetos, herencia y polimorfismo. Por otra parte, el bloque práctico tratará sobre cómo hay que aplicar los conocimientos teóricos en un lenguaje orientado a objetos: diseño de clases orientadas a objetos, manipulación y optimización de código orientado a objetos, diseño de interfaces gráficas de usuario, (modelo vista controlador), manejo de entrada/salida, redes y conexiones a bases de datos relacionales.

## OBJETIVO

El principal objetivo de esta asignatura consiste en que el estudiante aprenderá a analizar, diseñar, implementar y probar software usando el paradigma de Orientación a Objetos. Por ende será capaz de emplear métodos, técnicas y herramientas propias del ingeniero para la construcción de aplicaciones robustas y de fácil mantenimiento mediante el uso de este paradigma. Se hará especial énfasis en la fase de implementación, usando el lenguaje de programación C#.

## CONOCIMIENTOS PREVIOS

Los propios de acceso a Grado. Es altamente recomendable haber superado la asignatura de Algoritmos.

## CONTENIDOS

Tema 1. Introducción al paradigma Programación Orientada a Objetos.

- \*Abstracción.
- \*El Paradigma Orientado a Objetos.
- \*Lenguajes Orientados a Objetos.

Tema 2. Fundamentos de la Programación Orientada a Objetos.

- \*Clase.
- \*Atributos.
- \*Interfaces.
- \*Objetos.
- \*Metaclases.
- \*Relaciones entre clases y objetos.

Tema 3. Herencia.

- \*Herencia simple.
- \*Herencia múltiple.
- \*Herencia de interfaz.
- \*Herencia de implementación.
- \*Beneficios de la herencia.
- \*Reutilización de código.

Tema 4. Polimorfismo.

- \*Polimorfismo.
- \*Sobrecarga de operadores.
- \*Variables polimórficas.

Tema 5. Otras características del paradigma.

- \*Gestión de errores.
- \*Persistencia.
- \*Concurrencia.

Tema 6. Introducción a las interfaces gráficas.

- \*Librerías gráficas.
- \*Tangible Using Interface.
- \*Graphical User Interface.

\*Construcción de GUIs elementales.

## ACTIVIDADES FORMATIVAS

La metodología seguida en esta asignatura está dirigida a conseguir un aprendizaje significativo por parte del alumno de los conceptos y técnicas fundamentales de la materia. Por ese motivo se combinan lecciones expositivas con clases prácticas, sesiones de laboratorio y presentación de trabajos, de manera que se favorezca la participación del alumno y la interacción alumno-profesor y alumno-alumno como vía para fomentar el aprendizaje colaborativo y la capacidad de autoaprendizaje, todo ello mediante estrategias de resolución de problemas y metodologías de intervención. Las actividades no presenciales, que pueden ser tanto de tipo individual como colectivo, serán supervisadas por el profesor en clases y tutorías, tanto individuales como de grupo, estando encaminadas a favorecer el aprendizaje autónomo y colaborativo.

El trabajo presencial se completará con una importante carga de trabajo autónomo por parte del alumno, en muchos casos desarrollado en grupo, de manera que se fomente el aprendizaje colaborativo y cooperativo. Las actividades de carácter no presencial previstas incluyen el estudio y trabajo individual, que permitirá trabajar en la fijación de los conceptos abordados en las clases expositivas, así como de la aplicación que de los mismos se realiza en las clases prácticas, laboratorios y talleres. Todo el estudio y trabajo realizado por el alumno será supervisado y guiado por el profesor mediante tutorías, individuales o en grupo. En algunos casos, el alumno tendrá que realizar en clase la exposición de las principales conclusiones de su estudio o trabajo, lo que permitirá el intercambio de conocimientos y experiencias entre alumnos que fomentan la necesidad de comunicación efectiva y la capacidad de síntesis.

Finalmente, con el fin de facilitar al alumno el acceso a los materiales y la planificación de su trabajo, así como la comunicación con el profesor y el resto de alumnos, se empleará el Aula Virtual, que es una plataforma de aprendizaje que ofrece diferentes recursos electrónicos para complementar, de forma muy positiva, el aprendizaje del alumno.

## DISTRIBUCIÓN DE LOS TIEMPOS DE TRABAJO

ACTIVIDAD PRESENCIAL	TRABAJO AUTÓNOMO/ACTIVIDAD NO PRESENCIAL
67 horas	83 horas
Lección expositiva 14h Clase práctica 9h Laboratorio 29h Presentación de trabajos 8h Tutorías 4h Evaluación 3h	Estudio y trabajo individual 48h Trabajo en grupo 35h

## COMPETENCIAS

### Competencias básicas

Que los estudiantes hayan demostrado poseer y comprender conocimientos en un área de estudio que parte de la base de la educación secundaria general, y se suele encontrar a un nivel que, si bien se apoya en libros de texto avanzados, incluye también algunos aspectos que implican conocimientos procedentes de la vanguardia de su campo de estudio

Que los estudiantes sepan aplicar sus conocimientos a su trabajo o vocación de una forma profesional y posean las competencias que suelen demostrarse por medio de la elaboración y defensa de argumentos y la resolución de problemas dentro de su área de estudio

Que los estudiantes tengan la capacidad de reunir e interpretar datos relevantes (normalmente dentro de su área de estudio) para emitir juicios que incluyan una reflexión sobre temas relevantes de índole social, científica o ética

Que los estudiantes puedan transmitir información, ideas, problemas y soluciones a un público tanto especializado como no especializado

Que los estudiantes hayan desarrollado aquellas habilidades de aprendizaje necesarias para emprender estudios posteriores con un alto grado de autonomía

### **Competencias generales**

Conocimientos para la realización de mediciones, cálculos, valoraciones, tasaciones, peritaciones, estudios, informes, planificación de tareas y otros trabajos análogos de informática.

Capacidad para concebir, desarrollar y mantener sistemas, servicios y aplicaciones informáticas empleando los métodos de la ingeniería del software como instrumento para el aseguramiento de su calidad.

Conocimiento de las materias básicas y tecnologías, que capaciten para el aprendizaje y desarrollo de nuevos métodos y tecnologías, así como las que les doten de una gran versatilidad para adaptarse a nuevas situaciones.

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.

### **Competencias específicas**

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

Capacidad para diseñar y evaluar interfaces persona computador que garanticen la accesibilidad y usabilidad a los sistemas, servicios y aplicaciones informáticas.

Capacidad para diseñar, desarrollar, seleccionar y evaluar aplicaciones y sistemas informáticos, asegurando su fiabilidad, seguridad y calidad, conforme a principios éticos y a la legislación y normativa vigente.

## **RESULTADOS DE APRENDIZAJE**

Identifica el paradigma de programación orientada a objetos y sus principales ventajas.

Desarrolla programas que faciliten la reutilización de código.

Desarrolla programas basados en el paradigma de programación orientada a objetos, desde el diseño y definición de las clases y objetos, que los forman hasta su implementación y posterior depuración con un lenguaje y entorno de desarrollo determinado.

Diseña la interfaz de una aplicación tanto en modo consola o línea de comandos como mediante interfaces gráficas o entornos de ventanas.

Aplica metodologías y buenas prácticas en la construcción de aplicaciones en el paradigma de programación orientada a objetos con diferentes entornos y herramientas de desarrollo.

Utiliza correctamente las excepciones en programas orientados a objetos.

Codifica programas guiados por eventos utilizando el modelo de eventos basado en la delegación.

Realiza programas orientados a objetos que intercambian información a través de redes mediante la arquitectura cliente/servidor.

Desarrolla programas orientados a objetos que accedan y procesen información almacenada en bases de datos.

## **SISTEMA DE EVALUACIÓN DEL APRENDIZAJE**

El sistema de evaluación contempla cuatro tipos de pruebas:

- [1] Examen escrito teórico-práctico: presenta un peso del 50% en la nota final.
- [2] Examen práctico de laboratorio: presenta un peso del 25% en la nota final.
- [3] Prácticas y otros trabajos relacionados con la asignatura: presenta un peso del 20% en la nota final.
- [4] Participación en clase e implicación en la asignatura: presenta un peso del 5% en la nota final, siendo requisito imprescindible haber asistido como mínimo al 80% de las sesiones. En caso contrario este tipo de prueba se calificará con 0 puntos.

En las tres primeras pruebas es necesario obtener un mínimo de 5 puntos sobre 10 para poder aprobar la asignatura.

Aquellos alumnos que estén exentos de la obligación de asistir a clase, bien por segunda matrícula en la asignatura o sucesivas, bien por contar con autorización expresa de la Dirección del Grado, serán evaluados por el mismo tipo de pruebas. El 5% de la participación en clase podrán obtenerlo asistiendo al menos a tres tutorías con el profesor responsable de la asignatura.

Recuperación en convocatoria ordinaria:

Los alumnos que no hayan alcanzado la nota mínima en el examen escrito y/o el examen de laboratorio, podrán optar a una recuperación al final del semestre.

Recuperación en convocatoria extraordinaria:

Los alumnos que no hayan alcanzado la nota mínima en el examen escrito, el examen de laboratorio y/o prácticas, habiendo suspendido por tanto en la convocatoria ordinaria, podrán optar a una recuperación en la convocatoria extraordinaria.

En ambas recuperaciones (ordinaria y extraordinaria) el alumno se presentará solo a las partes que tenga evaluadas por debajo de 5.

La nota ponderada de la evaluación continua será un valor entre 0 y 10 y se calculará como sigue:  $0,50*[1] + 0,25*[2] + 0,20*[3] + 0,05*[4]$ .

A efecto de cómputo de convocatorias en una asignatura, solamente se contabilizarán como consumidas aquellas en las que el alumno se haya presentado a todas las pruebas de evaluación, o a una parte de las mismas, siempre que su peso en la nota final supere el 50%, aunque no se presente al examen final. Se entenderá que un alumno se ha presentado a una prueba aunque la abandone una vez comenzada la misma. La condición de No Presentado en la convocatoria extraordinaria estará ligada a la no asistencia o entrega de ninguna prueba, práctica o trabajo que esté pendiente.

---

[1] Examen a mitad de cuatrimestre de carácter teórico-práctico, con cuestiones cortas, preguntas de desarrollo y ejercicios prácticos de programación (Consola y/o Formulario). Este examen representará un 50% de la calificación final y evaluará la primera mitad del temario. El examen se puntuará de 0 a 10, repartiendo esta puntuación de manera equitativa entre todos los ejercicios y apartados, salvo que se indique lo contrario. Se evaluará el planteamiento de los problemas, así como la corrección, presentación e interpretación de los resultados obtenidos.

[2] Examen de carácter práctico a final de curso con preguntas de desarrollo y ejercicios prácticos de programación (ficheros, bases de datos y sockets) que representará un 25% de la calificación final. Se evaluará la segunda mitad del temario, si bien, debido a la relación de todos los conceptos vistos en la asignatura, se recomienda encarecidamente repasar los contenidos de la primera parte. También se puntuará de 0 a 10, repartiendo esta puntuación de manera equitativa entre todos los ejercicios y apartados, salvo que se indique lo contrario. Se evaluará el planteamiento de los problemas, así como la corrección, presentación e interpretación de los resultados obtenidos.

Ambas pruebas [1] y [2] se realizarán sin transparencias, apuntes, libros ni cualquier otro material relacionado con la asignatura.

[3] Realización, y entrega, del dossier de prácticas de programación y otros trabajos (20% de la calificación final). Tareas individuales de diversa índole, incluyendo prácticas de programación y otros ejercicios relacionados con la asignatura. En el caso que el profesor estime oportuno, la calificación quedará afectada por la defensa oral del trabajo, al alza o a la baja, para asegurar la autoría de los trabajos.

[4] Participación e implicación: 5% de la calificación final. Se evaluarán los ejercicios y otras actividades en grupo, el interés mostrado por el alumno, concretamente se computará el índice de asistencia a tutorías tanto individuales o grupales, el grado de participación activa en las clases mediante la respuesta a preguntas del profesor, el estudio de temas avanzados no vistos en clase, la recopilación de noticias aparecidas en los medios de comunicación relacionadas con la asignatura, etc. La calificación de este apartado será un valor numérico entre 0 y 10. Aunque esta nota sea inferior a 5, no se podrá optar a recuperación.

Cualquier tipo de fraude o plagio (\*) por parte del alumno en una actividad evaluable, será sancionado e implicará un 0 en la calificación de esa parte de la asignatura, anulando la convocatoria en curso. Esta situación, además, será comunicada a la Dirección de la Carrera, que a su vez comunicará a Secretaría General, siguiendo el protocolo establecido en la Universidad Francisco de Vitoria.

(\*) Se considerará "plagio" cualquier tipo de copia de ejercicios de examen, memorias de trabajos, ejercicios, etc.,

ya sea de manera total o parcial, de trabajos ajenos al alumno con el engaño de hacer creer al profesor que son propios.

## **BIBLIOGRAFÍA Y OTROS RECURSOS**

### **Básica**

CLARK, Dan. Introducción a la Programación Orientada a Objetos. Editorial Anaya Multimedia. Madrid. 2003. ISBN: 978-84-415-1470-6.

SCHILDT, Herbert. C++: Manual de Referencia. Editorial McGraw-Hill. Madrid. 1995. ISBN: 978-84-481-0321-7.

Material disponible en el Aula Virtual.

### **Complementaria**

JOYANES AGUILAR, Luis. Programación en C++. Serie Schaum. Editorial McGraw-Hill. Madrid. 2006. ISBN: 978-84-481-4643-6.

BRUEGGE, Bernd y DUTOIT, Allen. Ingeniería del Software Orientados a Objetos. Editorial Prentice Hall. México. 2002. ISBN: 978-970-26-0010-7.

SIERRA URRECHO, Alejandro y ALFONSECA MORENO, Manuel. Programación en C/C++. Editorial Anaya Multimedia. Madrid. 1999. ISBN: 978-84-415-0847-7.

SMITH, Jo Ann. C++: Programación Orientada a Objetos. Editorial Thomson Paraninfo. Madrid. 1999. ISBN: 978-84-283-2650-6.