

DATOS DE IDENTIFICACIÓN

Titulación:	Grado en Ingeniería Informática		
Rama de Conocimiento:	Ingeniería y Arquitectura		
Facultad/Escuela:	Escuela Politécnica Superior		
Asignatura:	Programación Orientada a Objetos		
Tipo:	Formación Básica	Créditos ECTS:	6
Curso:	1	Código:	5612
Periodo docente:	Segundo semestre		
Materia:	Informática		
Módulo:	Formación Básica		
Tipo de enseñanza:	Presencial		
Idioma:	Castellano		
Total de horas de dedicación del alumno:	150		

Equipo Docente	Correo Electrónico
Iván Barcia Santos	i.barcia@ufv.es
Moisés Martínez Muñoz	moises.martinez@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.

Esta asignatura corresponde al módulo Formación Básica y, dentro de éste, a la materia Informática. 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.

Se compone de dos grandes bloques conceptuales:

- Bloque teórico para conocer el paradigma de programación orientada a objetos y sus principales ventajas
- Bloque práctico para el desarrollo de aplicaciones mediante dicho paradigma, 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.

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, encapsulación, 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, manejo de entrada/salida en consola y diseño de interfaces de usuario.

OBJETIVO

El principal objetivo de esta asignatura consiste en que el estudiante aprenda a analizar, diseñar, implementar y probar software usando el paradigma de la Programación Orientada a Objetos. De esta forma, 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 Introducción a la Programación, conociendo los siguientes conceptos:

- Estructura de un programa
- Expresiones
- Operadores
- Identificadores
- Sentencias de control: instrucciones de selección e iteración.
- Funciones.
- Arrays.
- Operaciones básicas sobre archivos.
- Punteros

CONTENIDOS

Tema 1. El paradigma de la Programación Orientada a Objetos

- Antecedentes y perspectiva histórica
- ¿Qué es la POO?
- El principio de abstracción

- Conceptos básicos: ClasesAtributosMétodos
- Los cuatro principios: AbstracciónEncapsulaciónHerenciaPolimorfismo

Tema 2. Fundamentos de la POO

- El framework .NET
- El lenguaje de programación C#
- Aspectos básicos de C#
- Condiciones y declaraciones
- Instrucciones de selección
- instrucciones de iteración
- Métodos
- Clases y Objetos
- El principio de encapsulación
- Clases y Objetos: UML
- Estructuras de datos: ArrayArrayListLinked ListDictionary

Tema 3. Relaciones en la Programación Orientada a Objetos

- Herencia: ConstructoresModificadores de accesibilidadSobreescritura de métodos
- Asociación: AgregaciónComposición
- Abstracción: Clases abstractasMétodos abstractosInterfaces

Tema 4. POO Extendida

- Polimorfismo
- Sobrecarga de métodos
- Miembros virtuales
- Archivos (de texto, csv, json)
- Clases y funciones auxiliares: Números aleatoriosArgumentos de la línea de comandosExcepciones

Tema 5. Interfaces gráficas de usuario (GUI)

ACTIVIDADES FORMATIVAS

La metodología seguida en esta asignatura está dirigida al aprendizaje por parte del alumno de los conceptos y técnicas fundamentales de la materia. Se combinan lecciones expositivas con clases prácticas, 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.

El trabajo presencial se completará con una importante carga de trabajo autónomo por parte del alumno, en algunos casos desarrollado en grupo, de manera que se fomente el aprendizaje colaborativo y cooperativo. Las actividades no presenciales, que pueden ser tanto de tipo individual como colectivo, serán supervisadas por el profesor en clases y tutorías (individuales o de grupo), estando dirigidas a favorecer el aprendizaje autónomo y colaborativo. Además, estas actividades de carácter no presencial incluyen el estudio y trabajo individual, que permitirán 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.

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. 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
70 horas	80 horas
Lección Expositiva 16h Clase Práctica 20h Laboratorio 20h Presentación y Defensa de Trabajos 4h Tutorías 4h Evaluación 6h	Estudio y Trabajo Individual 50h Trabajo en Grupo 30h

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

Capacidad para concebir, redactar, organizar, planificar, desarrollar y firmar proyectos en el ámbito de la ingeniería en informática que tengan por objeto, de acuerdo con los conocimientos adquiridos según lo establecido en el apartado 5 de esta memoria, la concepción, el desarrollo o la explotación de sistemas, servicios y aplicaciones informáticas.

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, de acuerdo con los conocimientos adquiridos según lo establecido en el apartado 5 de esta memoria.

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.

Competencias específicas

Capacidad para comprender y dominar los conceptos básicos de matemática discreta, lógica, algorítmica y complejidad computacional, y su aplicación para la resolución de problemas propios de la ingeniería.

Conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos, bases de datos y programas informáticos con aplicación en ingeniería.

Conocimiento de la estructura, organización, funcionamiento e interconexión de los sistemas informáticos, los fundamentos de su programación, y su aplicación para la resolución de problemas propios de la ingeniería.

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.

SISTEMA DE EVALUACIÓN DEL APRENDIZAJE

El sistema de evaluación contempla tres tipos de pruebas:

- [1] Prueba escrita: presenta un peso del 20% en la nota final. Es necesario obtener un mínimo de 5 puntos sobre 10 para poder aprobar la asignatura.
- [2] Examen práctico: presenta un peso del 30% en la nota final. Es necesario obtener un mínimo de 5 puntos sobre 10 para poder aprobar la asignatura.

- [3] Tareas individuales y grupales de diversa índole, incluyendo prácticas de programación y otros ejercicios relacionados con la asignatura: presenta un peso del 40% en la nota final. Es necesario obtener un mínimo de 5 puntos en cada práctica y/o trabajo para poder aprobar la asignatura. Una o varias pruebas podrán ser realizadas en grupos según lo indicado por el profesor. Si el profesor lo estima oportuno, la calificación quedará afectada por la defensa del trabajo, al alza o a la baja, para asegurar la autoría de los trabajos.
- [4] Participación en clase e implicación en la asignatura: presenta un peso del 10% 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. Se evaluará el interés mostrado por el alumno, computando la realización de ejercicios propuestos en clase, el índice de asistencia a tutorías 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.

Las pruebas [1] y [2] se realizarán sin transparencias, apuntes, libros ni cualquier otro material relacionado con la asignatura. Además, en ningún caso se podrán utilizar herramientas de Inteligencia Artificial a lo largo del desarrollo de la asignatura, salvo autorización expresa por parte del profesor.

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 10% de la participación en clase podrán obtenerlo asistiendo al menos a tres tutorías con el profesor de la asignatura.

Recuperación en convocatoria extraordinaria: los alumnos que no hayan alcanzado la nota mínima en la prueba escrita, en el examen práctico y/o en alguna de las 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.

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. Siempre que el alumno se presente al examen final, la convocatoria se considerará consumida, al igual que si se guarda alguna parte aprobada para la convocatoria extraordinaria. 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.

Cualquier tipo de fraude o plagio por parte del alumno en una actividad evaluable, será sancionado según se recoge en la Normativa de Convivencia de la UFV. A estos efectos, se considerará "plagio" cualquier intento de defraudar el sistema de evaluación, como copia en ejercicios, exámenes, prácticas, trabajos o cualquier otro tipo de entrega, bien de otro compañero, bien de materiales o dispositivos no autorizados, con el fin de hacer creer al profesor que son propios.

BIBLIOGRAFÍA Y OTROS RECURSOS

Básica

autores, José F. Vélez Serrano [y otros 3]. Diseñar y programar, todo es empezar :una introducción a la programación orientada a objetos usando UML Y JAVA /

Ceballos Sierra, Francisco Javier,autor. Programación orientada a objetos con C++ / Quinta edición.

López Goytia, José Luis. Programación orientada a objetos C++ y Java :un acercamiento interdisciplinario /

Complementaria

Blasco, Francisco. Programación orientada a objetos en Java /

autores, José F. Vélez Serrano [y otros 3]. Diseñar y programar, todo es empezar :una introducción a la programación orientada a objetos usando UML Y JAVA /