

PLAN DOCENTE DE LA ASIGNATURA

Curso académico: 2024/2025

Identificación y características de la asignatura			
Código	501275	Créditos ECTS	6
Denominación (español)	Desarrollo de Programas		
Denominación (inglés)	Program Development		
Titulaciones	Grado en Ingeniería Informática en Ingeniería de Computadores Grado en Ingeniería Informática en Ingeniería del Software		
Centro	Escuela Politécnica		
Semestre	3	Carácter	Obligatoria
Módulo	Común a la rama de Informática		
Materia	Programación		
Profesor/es			
Nombre	Despacho	Correo-e	Página web
José M ^a Conejero Manzano	9 (Pabellón Telecomunicaciones)	chemacm@unex.es	http://epcc.unex.es
María Encarnación Sosa Sánchez	41 (Pabellón Informática)	esosa@unex.es	http://epcc.unex.es
Área de conocimiento	Lenguajes y Sistemas Informáticos		
Departamento	Ingeniería de Sistemas Informáticos y Telemáticos		
Profesor coordinador (si hay más de uno)	Encarnación Sosa Sánchez		
Competencias ^{1*}			
<p>1. Competencias básicas establecidas para Grado en el Anexo I 3.2 del RD 861/2010:</p> <p>CB1: 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.</p>			

^{1*}Los apartados relativos a competencias, breve descripción del contenido, actividades formativas, metodologías docentes, resultados de aprendizaje y sistemas de evaluación deben ajustarse a lo recogido en la memoria verificada del título.

CB2: 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.

CB3: 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.

CB4: Que los estudiantes puedan transmitir información, ideas, problemas y soluciones a un público tanto especializado como no especializado.

CB5: Que los estudiantes hayan desarrollado aquellas habilidades de aprendizaje necesarias para emprender estudios posteriores con un alto grado de autonomía.

2. Competencias generales:

CG01 - 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 del anexo II de la resolución antes mencionada para la tecnología específica de Ingeniería del Software y de Ingeniería de Computadores, la concepción, el desarrollo o la explotación de sistemas, servicios y aplicaciones informáticas.

CG02 - Capacidad para dirigir las actividades objeto de los proyectos del ámbito de la Informática de acuerdo con los conocimientos adquiridos según lo establecido en el apartado 5 del anexo II de la resolución antes mencionada para la tecnología específica de Ingeniería del Software y de Ingeniería de Computadores.

CG03 - 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.

CG04 - Capacidad para definir, evaluar y seleccionar plataformas hardware y software para el desarrollo y la ejecución de sistemas, servicios y aplicaciones informáticas, de acuerdo con los conocimientos adquiridos según lo establecido en el apartado 5 del anexo II de la resolución antes mencionada para la tecnología específica de Ingeniería del Software y de Ingeniería de Computadores.

CG05 - 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 del anexo II de la resolución antes mencionada para la tecnología específica de Ingeniería del Software y de Ingeniería de Computadores.

CG08 - 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.

CG09 - 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/a Técnico/a en Informática.

CG10 - Conocimientos para la realización de mediciones, cálculos, valoraciones, tasaciones, peritaciones, estudios, informes, planificación de tareas y otros trabajos análogos de informática, de acuerdo con los conocimientos adquiridos según lo establecido en el apartado 5 del anexo II de la resolución antes mencionada para la tecnología específica de Ingeniería del Software y de Ingeniería de Computadores.

3. Competencias específicas:

CI07: Conocimiento, diseño y utilización de forma eficiente de los tipos y estructuras de datos más adecuados a la resolución de un problema.

CI08: Capacidad para analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados.

4. Competencias transversales:
CT03: Capacidad para resolver problemas.
CT07: Capacidad de análisis y síntesis.

Contenidos

Breve descripción del contenido*

Fundamentos teóricos de programación y lenguajes de programación. Utilización de lenguajes orientados a objeto para el desarrollo de sistemas software. Estructuras de datos básicas, sus aplicaciones y propiedades. Técnicas de verificación y validación de programas.

Temario de la asignatura

Denominación del tema 1: **Clases y Objetos.**

Contenidos del tema 1:

1. Clases y Objetos
2. Atributos
3. Estado
4. Métodos y Parámetros
5. Instancias
6. Constructores
7. Ámbito y vida útil
8. Sentencias de asignación
9. Anexos:
 - 9.1. Tipos de datos primitivos
 - 9.2. Operadores
 - 9.3. Sobrecarga de operadores
 - 9.4. Operadores interesantes

Descripción de las actividades prácticas del tema 1:

- Explicación de la metodología que se seguirá en las sesiones de laboratorio.
- Ejercicios de toma de contacto con el IDE (*Entorno de Desarrollo Integrado*) que usaremos en la asignatura.
- Resolución de problemas que impliquen aplicar los contenidos tratados.

Denominación del tema 2: **Interacción de objetos.**

Contenidos del tema 2:

1. Modularización y abstracción
2. Tipos de datos: Primitivos y Objetos
3. Diagrama de clase
4. Creación de objetos
5. Diagrama de objetos
6. Sobrecarga de constructores
7. Sobrecarga de métodos
8. Interacción de objetos
9. Ocultando información
10. Valores nulos
 - 10.1. Valores por defecto
11. Gestión interna de la memoria

<p>11.1. Tipos de datos primitivos 11.2. Tipos de datos objeto 11.3. Liberación de memoria 11.4. Paso de parámetros: tipos primitivos vs objetos.</p> <p>12. Bibliotecas de Clases 13. Documentación de clases 14. Anexos 14.1. Depurar programas</p> <p>Descripción de las actividades prácticas del tema 2:</p> <ul style="list-style-type: none"> Resolución de problemas que impliquen aplicar los contenidos tratados.
<p>Denominación del tema 3: Buen comportamiento de los objetos. Contenidos del tema 3:</p> <ol style="list-style-type: none"> Prevención y detección de errores Pruebas (testing) <ol style="list-style-type: none"> Automatización de pruebas unitarias Depuración (debugging) <p>Descripción de las actividades prácticas del tema 3:</p> <ul style="list-style-type: none"> Resolución de problemas que impliquen aplicar los contenidos tratados.
<p>Denominación del tema 4: Agrupando y ordenando objetos. Contenidos del tema 4:</p> <ol style="list-style-type: none"> Colecciones de tamaño fijo Colecciones dinámicas Iteración Orden en colecciones de elementos Procesamiento funcional de colecciones Anexos <ol style="list-style-type: none"> Objetos Anónimos Arrays bidimensionales y bucles anidados Utilidades de IDEs: completar código <p>Descripción de las actividades prácticas del tema 4:</p> <ul style="list-style-type: none"> Resolución de problemas que impliquen aplicar los contenidos tratados.
<p>Denominación del tema 5: Herencia. Contenidos del tema 5:</p> <ol style="list-style-type: none"> Herencia <ol style="list-style-type: none"> Constructores Polimorfismo <ol style="list-style-type: none"> Subtipos y Sustitución Colecciones Polimórficas Casting Tipo Estático y Dinámico Sobreescritura Polimorfismo en Métodos Clases y métodos abstractos Extendiendo jerarquías de clases Herencia multiple Acceso Protegido (protected) <p>Descripción de las actividades prácticas del tema 5:</p> <ul style="list-style-type: none"> Resolución de problemas que impliquen aplicar los contenidos tratados.
<p>Denominación del tema 6: Comportamientos más avanzados. Contenidos del tema 6:</p>

<ol style="list-style-type: none"> 1. Colecciones de elementos 2. Clases wrapper (envoltorio). 3. Comportamiento aleatorio (Random) 4. Variables y métodos de clase. Constantes 5. Tipos enumerados 6. Definición y uso del método principal de un programa <p>Descripción de las actividades prácticas del tema 6:</p> <ul style="list-style-type: none"> • Resolución de problemas que impliquen aplicar los contenidos tratados.
<p>Denominación del tema 7: Manejo de errores.</p> <p>Contenidos del tema 7:</p> <ol style="list-style-type: none"> 1. Programación Defensiva. <ol style="list-style-type: none"> 1.1. Anticipar posibles errores 2. Lanzamiento y manejo de Excepciones. 3. Informar de errores. <p>Descripción de las actividades prácticas del tema 7:</p> <ul style="list-style-type: none"> • Resolución de problemas que impliquen aplicar los contenidos tratados.
<p>Denominación del tema 8: Manejo de ficheros (Entrada/Salida).</p> <p>Contenidos del tema 8:</p> <ol style="list-style-type: none"> 1. Entrada y salida de datos usando ficheros 2. Escritura en ficheros 3. Lectura de ficheros 4. Serialización de objetos <p>Descripción de las actividades prácticas del tema 8:</p> <ul style="list-style-type: none"> • Resolución de problemas que impliquen aplicar los contenidos tratados.
<p>Denominación del tema 9: Mejorando la calidad del código.</p> <p>Contenidos del tema 9:</p> <ol style="list-style-type: none"> 1. Acoplamiento 2. Cohesión 3. Diseño dirigido por responsabilidad 4. Acoplamiento implícito 5. Refactorización 6. Formas de Cohesión <p>Descripción de las actividades prácticas del tema 9:</p> <ul style="list-style-type: none"> • Resolución de problemas que impliquen aplicar los contenidos tratados.
<p>Denominación del tema 10: Principios y Patrones de Diseño.</p> <p>Contenidos del tema 10:</p> <ol style="list-style-type: none"> 1. Características de un Buen Diseño 2. Principios de Diseño 3. Principios SOLID 4. Patrones de Diseño <ol style="list-style-type: none"> 4.1. Clasificación de los Patrones 4.2. Algunos Patrones para conocer en profundidad: <ol style="list-style-type: none"> 4.2.1. Singleton 4.2.2. Template Method 4.2.3. Observer 4.2.4. Strategy 4.2.5. Decorator <p>Descripción de las actividades prácticas del tema 10:</p> <ul style="list-style-type: none"> • Resolución de problemas que impliquen aplicar los contenidos tratados.

Denominación del tema 11: **Antipatrones de Diseño.**

Contenidos del tema 11:

1. Clasificación de los Antipatrones
2. Algunos AntiPatrones a conocer en profundidad:
 - 2.1. Cortar y Pegar
 - 2.2. Código Spaguetti
 - 2.3. Blob
 - 2.4. Martillo Dorado

Descripción de las actividades prácticas del tema 11:

- Resolución de problemas que impliquen aplicar los contenidos tratados.

Actividades formativas*

Horas de trabajo del estudiante por tema		Horas teóricas	Actividades prácticas				Actividad de seguimiento	No presencial
Tema	Total	GG	PCH	LAB	ORD	SEM	TP	EP
1	8	3		2				3
2	9	3		2				4
3	7	1		2				4
4	14	4		4				6
5	22	4		4			1	14
6	8	2		2				4
7	5	1		2				2
8	5	1		2				2
9	8	2		2				4
10	29	6		6			1	16
11	2	1		0				1
Evaluación^{2**}	33	2		2				28
TOTAL	150	30		30			2	88

GG: Grupo Grande (100 estudiantes).

PCH: prácticas clínicas hospitalarias (7 estudiantes)

LAB: prácticas laboratorio o campo (15 estudiantes)

ORD: prácticas sala ordenador o laboratorio de idiomas (30 estudiantes)

SEM: clases problemas o seminarios o casos prácticos (40 estudiantes).

TP: Tutorías Programadas (seguimiento docente, tipo tutorías ECTS).

EP: Estudio personal, trabajos individuales o en grupo, y lectura de bibliografía.

Metodologías docentes*

- En Clases teórico-prácticas en el aula. Clases expositivas para el desarrollo de los contenidos fundamentales de las materias.

^{2**} Indicar el número total de horas de evaluación de esta asignatura.

- En Clases teórico-prácticas en el aula. Actividades breves, individuales o en grupo que permitan aplicar los conceptos expuestos y resolver problemas, facilitando la participación activa de los estudiantes.
- En sesiones de laboratorio. Actividades prácticas, sesiones de laboratorio guiadas, seminarios de resolución de problemas, etc. en grupos bajo la dirección de un profesor. Se podrán incluir actividades previas y posteriores a las sesiones de laboratorio y seminario que ayuden a conseguir los objetivos propuestos. Se fomentarán especialmente las actividades encaminadas al desarrollo de proyectos, supuestos prácticos, informes, etc.
- En tutorías programadas. individuales o en grupos pequeños se realizará un seguimiento más individualizado del estudiante, con actividades de formación y orientación. Principalmente, se utilizarán para el seguimiento de los trabajos planteados, debate sobre alternativas y evaluación de los objetivos alcanzados.
- Realización de actividades, trabajos y estudio por parte del estudiante, de manera autónoma, individualmente o en grupo. Las actividades que el estudiante desarrollará de manera no presencial estarán orientadas principalmente a la adquisición de conocimientos básicos en el ámbito de la Informática y al desarrollo de los proyectos y trabajos solicitados, bien individualmente o en grupo.

Más concretamente, el desarrollo de la asignatura se realizará en sesiones presenciales de dos tipos: sesiones de teoría y sesiones de práctica.

- Con respecto a las **sesiones de teoría**, en estas se comenzarán resolviendo las dudas sobre la parte de los materiales del tema a tratar que los estudiantes deben haber trabajado antes de cada sesión (siendo la dedicación aconsejada a este trabajo previo de aproximadamente media hora). A continuación, se procederá a trabajar el resto del material alternando explicaciones teóricas y distintos tipos de actividades que fomenten la participación activa de los estudiantes con el objetivo de facilitar el aprendizaje de los conceptos del tema.
- En cuanto a las **sesiones prácticas**, el estudiante parte de un guión previo en el que tiene detallados los conceptos que se van a trabajar en la sesión, una serie de ejercicios previos aconsejados, así como una serie de ejercicios que deberá realizar durante la sesión. En este guión, se detallarán también los ejercicios propuestos no presenciales que el estudiante debe realizar después de cada sesión para afianzar los conceptos trabajados durante la sesión.

Resultados de aprendizaje*

Al completar la materia, el estudiante:

- Puede utilizar de manera eficaz un entorno de programación que incluya herramientas de edición, compilación, depuración y documentación de programas.
- Justifica la utilización de distintos paradigmas de programación y plataformas de desarrollo de software en un determinado contexto.
- Busca, analiza, sintetiza y critica nueva información para aprender nuevos lenguajes, algoritmos, técnicas, paradigmas y metodologías de programación aplicables a distintas áreas, teniendo como objetivo la actualización continua de los conocimientos y competencias.
- Analiza, planifica, diseña y desarrolla soluciones algorítmicas y programas robustos y correctos a problemas planteados, argumentando las decisiones tomadas, evaluando el resultado final y documentando el código y el proceso.

- Conoce y aplica en actividades de nivel medio las competencias transversales fundamentales de la profesión

Sistemas de evaluación*

El sistema de evaluación se rige por la Normativa de Evaluación de la Universidad de Extremadura (DOE 06/10/2020).

Así, cada estudiante podrá ser calificado en la asignatura atendiendo a dos modalidades diferentes:

- Evaluación **continua**: sistema de evaluación constituido por diversas actividades distribuidas a lo largo del semestre de docencia de una asignatura. Esta modalidad incluye una prueba final, entendida como el conjunto de actividades de evaluación que tienen condicionada su celebración a la fecha oficial de examen para cada convocatoria.
- Evaluación **global**: sistema de evaluación constituido exclusivamente por una prueba final, que englobe todos los contenidos de la asignatura y que se realizará en la fecha oficial de cada convocatoria.

El estudiante podrá elegir, a través del Campus Virtual, la modalidad con la que quiere ser evaluado para cada convocatoria. En caso de ausencia de solicitud expresa por parte del estudiante, la modalidad asignada será la de evaluación **continua**.

En ambas modalidades es posible obtener la máxima calificación. La principal diferencia entre ambas modalidades radica en la distinta ponderación de los **bloques de calificación** que componen la asignatura.

Estos bloques se han realizado basándose en la metodología de la asignatura y son los siguientes:

- Bloque 1. Evaluación de proyectos de programación y sus informes técnicos
- Bloque 2. Evaluación de conceptos
- Bloque 3. Cuadernos de laboratorio
- Bloque 4. Prueba de evaluación final del proyecto de programación

Bloque 1: proyecto de programación e informe técnico

La realización de proyectos de programación es la parte más importante de la asignatura y permitirá evaluar la mayoría de las competencias técnicas y transversales desarrolladas por el estudiante.

A lo largo del semestre, los estudiantes desarrollarán proyectos de programación guiados y divididos en partes. Estos proyectos podrán desarrollarse desde cero o partir de proyectos previos semi implementados. Durante el desarrollo de los proyectos los grupos tendrán un seguimiento por parte del equipo docente. Durante las sesiones prácticas y teóricas se impartirán todos los conocimientos y conceptos relacionados con la asignatura. La comprensión de estos conceptos es clave para que los estudiantes puedan aplicarlos en los proyectos de programación.

Los proyectos deberán entregarse funcionando correctamente y documentado en las fechas indicadas que serán publicadas en tiempo y forma a través del Campus Virtual. Los proyectos y sus informes técnicos deben ajustarse a los criterios especificados por el equipo docente de la asignatura.

Los proyectos se realizarán en grupos de 3 estudiantes.

Bloque 2. Evaluación de conceptos básicos

Se realizará una evaluación individual sobre los conceptos básicos (teóricos y prácticos) estudiados durante el semestre. Esta evaluación podrá realizarse de forma virtual y podría constar de pruebas tipo test, resolución de problemas, preguntas cortas,

resolución de ejercicios o resolución de problemas prácticos relacionados con el proyecto o desarrollo escrito.

Bloque 3: Evaluación de cuadernos de laboratorio

Se propondrán a lo largo del semestre ejercicios en las sesiones de laboratorio que serán realizados de forma individual por los estudiantes. Dichos ejercicios podrán realizarse de forma virtual y estarán relacionados con los conceptos teóricos y prácticos que se estudian en la asignatura, así como con el desarrollo de los proyectos de programación.

Bloque 4: Prueba de evaluación final del proyecto de programación

Es una prueba de evaluación final relacionada directamente con los proyectos entregados por los estudiantes. Esta evaluación podrá realizarse de forma virtual y podría constar de resolución de preguntas tipo test, preguntas cortas, resolución de ejercicios o resolución de problemas. Esta prueba solo se evaluará si se supera el bloque 1.

Ponderación y normativa específica de los bloques en cada Modalidad de evaluación:

Evaluación continua.

Evaluación del Bloque 1:

- Este bloque es **recuperable** (puede ser entregado en todas las convocatorias).
- Su calificación es un 45% de la calificación final de la asignatura.
- Para superar este bloque, el estudiante por la modalidad de evaluación continua debe:
 - o Entregar **todos los proyectos de programación e informes técnicos**.
 - o Obtener una calificación mínima de 5 en todas las entregas.
 - o En caso de superarlo, la calificación de este bloque se mantiene durante las siguientes convocatorias del mismo curso académico.

Evaluación del Bloque 2:

- Este bloque es **recuperable** (puede repetirse en todas las convocatorias).
- La calificación de este bloque supone un 45% de la calificación final de la asignatura.
- Para superar este bloque, el estudiante debe obtener una calificación mínima de 5.
- En caso de superarlo, la calificación de este bloque se mantiene durante las siguientes convocatorias del mismo curso académico.

Evaluación del Bloque 3:

- Este bloque **no es recuperable**, de modo que una vez finalizada la convocatoria oficial de enero, no se propondrán más actividades en laboratorio.
 - o Sin embargo, si un estudiante ha obtenido puntos en este bloque, la nota del mismo será tenida en cuenta en las siguientes convocatorias del mismo curso académico.
- La puntuación de este bloque supondrá el 10% de la calificación final de la asignatura.

Evaluación del Bloque 4:

- Este bloque es **recuperable** (puede repetirse en todas las convocatorias).

- La prueba de evaluación final será evaluada como **APTO/NO APTO**.
- En caso de superar este bloque y el bloque 1, la calificación de este bloque se mantiene durante las siguientes convocatorias del mismo curso académico.

Cálculo de la calificación final del estudiante por Evaluación Continua:

Si el estudiante supera cada bloque de acuerdo a las especificaciones anteriores, la calificación final se calculará según la siguiente fórmula:

$$\text{Calificación final} = \text{Bloque1} * 0,45 + \text{Bloque2} * 0,45 + \text{Bloque3} * 0,10$$

Evaluación global.

Evaluación del Bloque 1:

- Este bloque es **recuperable** (puede ser entregado en todas las convocatorias).
- Su calificación es un 30% de la calificación final de la asignatura.
- Para superar este bloque, el estudiante por la modalidad de evaluación global debe:
 - o Entregar **todos los proyectos de programación e informes técnicos** en la convocatoria oficial de la asignatura
 - o Obtener una calificación mínima de 5.
 - o En caso de superarlo, la calificación de este bloque se mantiene durante las siguientes convocatorias del mismo curso académico.

Evaluación del Bloque 2:

- Este bloque es **recuperable** (puede repetirse en todas las convocatorias).
- La calificación de este bloque supone un 70% de la calificación final de la asignatura.
- Para superar este bloque, el estudiante debe obtener una calificación mínima de 5.
- En caso de superarlo, la calificación de este bloque se mantiene durante las siguientes convocatorias del mismo curso académico.

Evaluación del Bloque 3:

- Los estudiantes en modalidad de evaluación global **no tienen posibilidad de realizar este bloque.**

Evaluación del Bloque 4:

- Este bloque es **recuperable** (puede repetirse en todas las convocatorias).
- La prueba de evaluación final será evaluada como **APTO/NO APTO**.
- En caso de superar este bloque y el bloque 1, la calificación de este bloque se mantiene durante las siguientes convocatorias del mismo curso académico.

Cálculo de la calificación final del estudiante por Evaluación Global:

Si el estudiante supera cada bloque de acuerdo a las especificaciones anteriores, la calificación final se calculará según la siguiente fórmula:

$$\text{Calificación final} = \text{Bloque1} * 0,30 + \text{Bloque2} * 0,70$$

En la siguiente tabla puede verse un resumen final del sistema de evaluación para ambas modalidades de evaluación:

Bloque	Ponderación ev. continua	Ponderación ev. global	Calificación mínima	Recuperable
1. Proyecto de programación e informe técnico	45%	30%	5	Sí
2. Evaluación conceptos básicos	45%	70%	5	Sí
3. Evaluación de cuadernos de laboratorio	10%	-	-	No
4. Prueba de evaluación final	-	-	APTO	Sí

Si no se cumplen los requisitos mínimos para los bloques 1, 2 y 4, y la calificación final obtenida es menor que 3, esa será la calificación en esa convocatoria. En otro caso, la calificación en esa convocatoria será "Suspenso – 3".

Si el estudiante entrega o se presenta a alguna de las actividades que tienen una calificación mínima, se entenderá que se ha presentado a esa convocatoria. En otro caso se considerará "No presentado".

La copia o el plagio demostrados en cualquier actividad supone una nota final de SUSPENSO (0) en la convocatoria y una nota de 0 en los bloques no recuperables para todos los implicados, además de las actuaciones legales indicadas según la normativa vigente. Es responsabilidad del estudiante o grupo de estudiantes la custodia y protección de su proyecto (se utilizará un software específico de detección de copias en programas).

Bibliografía (básica y complementaria)

Bibliografía básica:

- David J. Barnes, Michael Kolling. Programación Orientada a objetos con Java usando BlueJ. 6a Edición. Pearson. 2017
- Alexander Shvets. Sumérgete en los Patrones de Diseño. Refactoring Guru. 2020

Bibliografía complementaria:

- Roberto Rodríguez, Encarna Sosa, Álvaro Prieto. Programación Orientada a Objetos. Editado por Librería Álvaro. 2004
- Bertrand Meyer. Construcción de Software Orientada a Objetos. 2ª Edición. Prentice Hall. 1999
- Bruce Eckel. Piensa en Java. 4ª Edición. Pearson. 2007
- Erich Gamma, Richard Helm, Ralph Johnson, John M. Vlissides. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley. 1994
- Eric Freeman, Elizabeth Freeman, Kathy Sierra, Bert Bates. Head First Design Patterns. O'Reilly. 2004
- Tony Bevis. Java Design Patterns Essentials. 2nd Edition. Ability First. 2012
- Robert C. Martin. Código Limpio: Manual de estilo para el desarrollo ágil de software. 2012
- Joshua Bloch. Effective Java. 3rd ed. 2017.

Otros recursos y materiales docentes complementarios

Medios materiales utilizados:

- Teoría: aula virtual, aula física, cañón de vídeo, pizarra y dispositivos portátiles de los estudiantes.
- Práctica: aula virtual, laboratorio de ordenadores con todas las herramientas software de la asignatura correctamente instaladas, cañón de vídeo y pizarra.

Todo el material y recursos utilizados en la asignatura están disponibles en el aula virtual de la misma:

- Transparencias para cada tema de teoría.
- Guiones de las sesiones de laboratorio.
- Videotutoriales
- Planificación del curso, etc.

Además, en la **biblioteca** del centro existen ejemplares de los **libros aconsejados en la bibliografía**. Los manuales y enlaces digitales podrán ser consultados y/o descargados durante las sesiones prácticas, en las cuales se dispone de acceso a internet.