

PLAN DOCENTE DE LA ASIGNATURA: Metodología y Desarrollo de Programas

CÓDIGO: **501309**

CURSO ACADÉMICO: **2022/2023**

PLAN DOCENTE DE LA ASIGNATURA

Curso académico: 2022/2023

Identificación y características de la asignatura			
Código	501309		Créditos ECTS 6
Denominación (español)	Metodología y desarrollo de programas		
Denominación (inglés)	Methodology for software development		
Titulaciones	Grado en Ingeniería Informática en Tecnologías de la Información		
Centro	Centro Universitario de Mérida		
Semestre	3º	Carácter	Obligatoria
Módulo	Común a la rama de Informática		
Materia	Programación		
Profesor/es			
Nombre	Despacho	Correo-e	Página web
Luis Arévalo Rosado	9	ljarevalo@unex.es	http://campusvirtual.unex.es/portal/
Área de conocimiento	Lenguajes y Sistemas Informáticos		
Departamento	Ingeniería de Sistemas Informáticos y Telemáticos		
Profesor coordinador (si hay más de uno)			

Competencias	
Competencias básicas	
x	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
x	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
x	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
x	CB4 - Que los estudiantes puedan transmitir información, ideas, problemas y soluciones a un público tanto especializado como no especializado
x	CB5 - Que los estudiantes hayan desarrollado aquellas habilidades de aprendizaje necesarias para emprender estudios posteriores con un alto grado de autonomía
Competencias generales	
x	CG5 - 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 anexo-2 de la Resolución de 8 de junio de 2009 de la Secretaría General de Universidades (BOE de 4 de Agosto de 2009) en el ámbito de las Tecnologías de la Información.
	CG6 - Capacidad para concebir y desarrollar sistemas o arquitecturas informáticas centralizadas o distribuidas integrando hardware, software y redes de acuerdo con los

	conocimientos adquiridos según lo establecido en el anexo-2 de la Resolución de 8 de junio de 2009 de la Secretaría General de Universidades (BOE de 4 de Agosto de 2009) en el ámbito de las Tecnologías de la Información.
	CG8 - 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.
	CG9 - 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.
	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 anexo-2 de la Resolución de 8 de junio de 2009 de la Secretaría General de Universidades (BOE de 4 de Agosto de 2009) en el ámbito de las Tecnologías de la Información.
Competencias específicas	
	CE3 - 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.
	CE12 - Conocimiento y aplicación de los procedimientos algorítmicos básicos de las tecnologías informáticas para diseñar soluciones a problemas, analizando la idoneidad y complejidad de los algoritmos propuestos.
X	CE13 - Conocimiento, diseño y utilización de forma eficiente los tipos y estructuras de datos más adecuados a la resolución de un problema.
X	CE14 - Capacidad para analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados.
	CE20 - Conocimiento y aplicación de los principios fundamentales y técnicas básicas de la programación paralela, concurrente, distribuida y de tiempo real.
	CE21 - Conocimiento y aplicación de los principios fundamentales y técnicas básicas de los sistemas inteligentes y su aplicación práctica.
Competencias transversales	
	CT4. Resolución de problemas
X	CT5. Toma de decisiones
X	CT6. Orientación al aprendizaje
	CT11. Comunicación en lengua extranjera
	CT12. Diversidad e interculturalidad
	CT13. Resistencia y adaptación al entorno
	CT15. Comunicación interpersonal

Contenidos	
Breve descripción del contenido	
Diseño y uso de estructuras de datos avanzadas para la resolución de problemas. Lenguajes y paradigmas de programación para resolver problemas	
Temario de la asignatura	
Temario de la parte teórica	
Denominación del tema 1: Desarrollo de Software	
Contenidos del tema 1:	
<ul style="list-style-type: none"> • Metodología del software • Calidad del software 	

<ul style="list-style-type: none"> • Parámetros • Ciclo de vida
<p>Denominación del tema 2: Análisis y Diseño OO mediante UML. Contenidos del tema 2:</p> <ul style="list-style-type: none"> • Ciclo de vida • Planificación <ol style="list-style-type: none"> 1. Identificación. Título, descripción, versión, fecha,.. 2. Documento de análisis 3. Especificación de requisitos o requerimientos 4. Diagramas de casos de uso 5. Escenarios y sub-escenarios • Análisis y Diseño <ol style="list-style-type: none"> 1. Diagrama de secuencia 2. Diagrama de clase 3. Otros diagramas • Implementación y Pruebas • Ejercicios
<p>Denominación del tema 3: Patrones de diseño OO Contenidos del tema 3:</p> <ul style="list-style-type: none"> • Introducción a los patrones • Patrones de creación • Patrones estructurales • Patrones de comportamiento • Ejercicios
<p>Denominación del tema 4: Prueba y Documentación Contenidos del tema 4:</p> <ul style="list-style-type: none"> • Verificación • Pruebas del software <ol style="list-style-type: none"> 1. Objetivos, importancia y Principios de la prueba. 2. El proceso de prueba. 3. Métodos de diseño de casos de prueba. 4. Enfoque estructural y funcional. 5. Estrategias de prueba del software. • Documentación del Software
<p>Denominación del tema 5: Flujos Contenidos del tema 5:</p> <ul style="list-style-type: none"> • Entrada Orientada a Bytes • Salida Orientada a Bytes • Entrada Orientada a Caracteres • Salida Orientada a Caracteres • Entrada / Salida por Consola • Class File • Serialización • Lectura de un Archivo desde un Servidor de Internet • Ejercicios
<p>Denominación del tema 6: Programación Orientada a Eventos Contenidos del tema 6:</p> <ul style="list-style-type: none"> • Introducción • Problemática • GUI's/ Interfaces Gráficas de Usuarios <p>Herramientas visuales de desarrollo: JavaFX</p>
<p>TEMARIO DE LA PARTE PRÁCTICA</p> <ul style="list-style-type: none"> • Práctica 1: Introducción a la metodología de prácticas de la asignatura • Práctica 2: Programación orientada a Objetos • Práctica 3: Relaciones en la Programación Orientada a Objetos

- Práctica 4: Uso de Colecciones
- Práctica 5: Editores gráficos de UML
- Práctica 6: Patrones
- Práctica 7: Pruebas y documentación
- Práctica 8: Flujos
- Práctica 9: Interfaces Gráficas I (Componentes básicos)
- Práctica 10: Interfaces Gráficas II (Componentes avanzados)
- Práctica 11: Internacionalización de aplicaciones

Actividades formativas*								
Horas de trabajo del alumno por tema		Horas teóricas	Actividades prácticas				Actividad de seguimiento	No presencial
Tema	Total	GG	PCH	LAB	ORD	SEM	TP	EP
1	12	2			8		-	2
2	37	10			4		2	21
3	18	6			4		-	8
4	19	5			2		1	11
5	14	4			2		-	8
6	22	5			5		-	12
Evaluación^{1**}	28	3			0		-	25
TOTAL	150	35			25		3	87

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.

Metodología docentes
<ul style="list-style-type: none"> • Clases expositivas de teoría y problemas: Presentación de los contenidos de la asignatura y planificación de la participación de todos los estudiantes en las distintas tareas. Discusión de aspectos teóricos. Adicionalmente se realizarán charlas divulgativas realizadas por expertos y/o empresas de la materia. Esta metodología se podrá aplicar bien con el alumnado físicamente en aula o a distancia mediante videoconferencia • Enseñanza participativa: Trabajos prácticos en grupos medianos o pequeños. Esta metodología se podrá aplicar bien con el alumnado físicamente en aula o a distancia mediante videoconferencia • Tutorización: Actividad de seguimiento para tutela de trabajos dirigidos, consultas de dudas y asesoría en grupos pequeños o individuales. Esta metodología se podrá aplicar bien con el alumnado físicamente en aula o a distancia mediante videoconferencia

1 **Indicar el número total de horas de evaluación de esta asignatura.

- Aprendizaje autónomo mediante el análisis de documentos escritos, la elaboración de memorias, el estudio de la materia impartida y desarrollo de los supuestos prácticos planteados.
- Aprendizaje virtual. Uso de herramientas virtuales de comunicación entre profesor y estudiante e incluso entre los estudiantes entre sí.

Resultados de aprendizaje

- Justificar la utilización de distintos paradigmas de programación y plataformas de desarrollo de software en un determinado contexto.
- Buscar, analizar, sintetizar y criticar 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.
- Colaborar con otros en la toma de decisiones grupales de calidad. (CT5, 2do nivel de dominio)
- Incorporar los aprendizajes propuestos por los expertos y mostrar una actitud activa para su asimilación. (CT6, 1er nivel de dominio)

Sistemas de evaluación

Modalidad de Evaluación Continua

Se valorarán los conocimientos teóricos-prácticos adquiridos por el alumno o alumna en relación con las competencias y objetivos de la asignatura, así como su grado de participación en el proceso educativo. La evaluación se realiza de acuerdo a la memoria verificada en los siguientes puntos:

Sistemas de evaluación	Porcentaje
Examen de certificación.	40,00 %
Exposición oral de trabajos realizados.	0,00 %
Realización de trabajos dirigidos (informes, casos prácticos, ejercicios y problemas).	60,00 %

- Actividades de sesión de laboratorio (AS)(15%). (Recuperable) (Trabajos dirigidos)

Se valorará la entrega de las actividades solicitadas en laboratorio.

- Actividades de Grupo Grande (AG)(10%). (Recuperable) (Trabajos dirigidos)

Se valorará la entrega de distintas actividades realizadas en clase o fuera de ellas relacionadas con el contenido impartido en GG (Test, cuestionario,..) así como la presentación de trabajos.

- Entregas de Laboratorio (PL)(35%). (Recuperable) (Trabajos dirigidos)

Se realizará una o varias prácticas que requerirán la redacción de una memoria por parte de un alumno.

Nota: En cualquier convocatoria se podrá o bien solicitar una entrega de prácticas adicional o un examen práctico para demostrar la autoría de las prácticas.

- Examen final (EF)(40%). (Recuperable) (Examen de certificación)

Se realizará una prueba final por escrito que recogerá tanto los contenidos teóricos como prácticos de la asignatura.

$$\text{Nota final} = \text{AS} * 0,15 + \text{AG} * 0,1 + \text{PL} * 0,35 + \text{EF} * 0,40$$

Nota:

- En el examen final se debe tener una calificación mínima de 4 sobre 10 para aprobar la asignatura
- Podrá existir un examen de práctica final para demostrar la autoría de la misma. Si es considerado como apto se procederá a corregir la práctica final. La práctica final debe funcionar correctamente para que pueda ser evaluada. Se debe tener aprobada la práctica final (PL) para aprobar la asignatura.
- La competencia transversal (CT6, orientación al aprendizaje) consistirá en la planificación y

elaboración de un trabajo proporcionando al alumnado un detalle pormenorizado o guía del trabajo al tratarse de un nivel de dominio 1. Su evaluación se realizará dentro del apartado “Actividades de Grupo Grande”

- La competencia transversal (CT5, toma de decisiones) consistirá en la elaboración de un trabajo (coincidente con la primera entrega final) en el cual deberán tomar decisiones para la elaboración del diagrama UML resultante. Todos los diagramas se pondrán en común para su debate. Su evaluación se realizará como parte de la primera entrega final.

Modalidad de Evaluación Global

Por cada convocatoria, y dentro del plazo establecido por el CUME, el estudiante deberá notificar al profesorado que opta por esta evaluación, en caso contrario se evaluará por evaluación continua.

- Entregas de Laboratorio (PL)(40%).

Deberá entregar la práctica final funcionando correctamente y realizar una modificación sobre la misma. La evaluación se realizará sobre la entrega realizada.

- Examen final (EF)(60%).

Se realizará una prueba final por escrito que recogerá tanto los contenidos teóricos como prácticos de la asignatura. Este examen podrá ser distinto al examen de la evaluación continua.

Bibliografía (básica y complementaria)

Bibliografía básica:

- Ingeniería del Software
 - Ian Sommerville. “Ingeniería del Software”. Sexta edición. Editorial Addison Wesley, 2002. ISBN. 970-26-0206-8.
- UML y Patrones
 - Craig Larman. “UML y patrones. Introducción al análisis y diseño orientado a objetos”. Editorial Prentice may, 1999. ISBN: 970-17-0261-1.
 - A. Weitzenfeld. "Ingeniería de Software Orientada a Objetos con UML. Java e Internet. Editorial Thomson 2004. ISBN: 970-686-190-4.
- Java
 - Piensa en Java. Bruce Eckel. 4ª Edición. Pearson Prentice Hall. ISBN 13: 9788489660342
 - Programación, Algoritmos y ejercicios resultados en Java. David Camacho Fernández. Pearson Prentice Hall. ISBN 13: 9788420540245

Bibliografía complementaria:

- Ingeniería del Software
 - Bertrand Meyer. “Construcción de Software Orientado a Objetos”. 2ª edición. Editorial Prentice, 1999. ISBN: 84-8322-040-7.
- UML y Patrones
 - Eric T Freeman, Elisabeth Robson, Bert Bates, Kathy Sierra Publisher. Head First Design Patterns. . Editorial O'Reilly Media. October 2004
- Java
 - Core Java 2 Vol I. Horstmann. Fundamentos. Pearson Prentice Hall/Sun. ISBN 13: 9788420548326

Otros recursos y materiales docentes complementarios

Recursos virtuales:

- Aula virtual de la asignatura

Materiales y recursos utilizados:

Los materiales y recursos utilizados estarán disponibles en el espacio reservado para la asignatura en el Campus

Virtual. Concretamente los alumnos dispondrán de:

- Transparencias para cada tema del programa
- Guiones de las sesiones de laboratorio
- Foros de preguntas y respuestas
- Tablón de anuncios de novedades
- Conjunto de referencias web relacionadas con fundamentos de programación
- Tareas virtuales para la entrega de problemas propuestos