



## FICHA IDENTIFICATIVA

### DATOS DE LA ASIGNATURA

**Código:** 34662

**Nombre:** Ingeniería del software II

**Ciclo:** Grado

**Créditos ECTS:** 6

**Curso académico:** 2025-26

### TITULACIONES

Titulación	Centro	Curso	Periodo
1400 - Grado en Ingeniería Informática	Escola Tècnica Superior d'Enginyeria	3	Primer cuatrimestre

### MATERIAS

Titulación	Materia	Carácter
1400 - Grado en Ingeniería Informática	Ingeniería del Software y Gestión de Proyectos	OBLIGATORIA

### COORDINACIÓN

GIL PASCUAL MIRIAM

## RESUMEN

La asignatura Ingeniería del Software II forma parte de la materia Ingeniería del Software y Gestión de Proyectos. El objetivo general de esta asignatura es ampliar los contenidos ampliados en la asignatura Ingeniería del Software I. En concreto, enmarcándolos dentro de la metodología MÉTRICA versión 3, e introduciendo las metodologías ágiles, las pruebas de software, los procesos de mantenimiento y reingeniería de sistemas de información, y describiendo una serie de actividades de tipo organizativo o de soporte al proceso de desarrollo y a los productos, que en el caso de existir en la organización se deberán aplicar para enriquecer o influir en la ejecución de las actividades de los procesos principales de la metodología de desarrollo de software y que si no existen habrá que realizar para complementar y garantizar el éxito del proyecto desarrollado (Gestión de la configuración del software, Gestión y aseguramiento de la calidad).

En términos generales, los objetivos de la asignatura son:

- Dar a conocer al alumnado la visión general del proceso de desarrollo de sistemas de información, explicando las características propias de los procesos de mantenimiento y reingeniería de sistemas de información.



- Presentar las actividades de tipo organizativo o de soporte asociadas al proceso de desarrollo de software que deben aplicarse para enriquecer o influir en la ejecución de las actividades de los procesos principales de la metodología de desarrollo.
- Dar a conocer al alumnado las metodologías ágiles y cómo se implementan en el desarrollo de software versus las metodologías tradicionales.
- Dar a conocer la necesidad de mantener la integridad de los productos que se obtienen a lo largo del desarrollo de los sistemas, garantizando que no se realizan cambios incontrolados y que todos los participantes disponen de la versión adecuada de los productos que manejan.
- Proporcionar al alumnado un marco común de referencia para la definición y puesta en marcha de planes específicos de aseguramiento de calidad aplicables a proyectos concretos.
- Concienciar al alumnado de la necesidad de incorporar el proceso de pruebas de software para garantizar la calidad y fiabilidad de los productos software en el desarrollo de proyectos.
- Presentar al alumnado las técnicas más frecuentes para la medición y la estimación en proyectos software durante su desarrollo y mantenimiento.
- Dar a conocer la importancia de las arquitecturas software como estructuras fundamentales que proporcionan un marco para el diseño y desarrollo de sistemas software.

Desde el punto de vista docente, la asignatura tiene un planteamiento fundamentalmente práctico y está enfocada al desarrollo de habilidades prácticas para el ingeniero que deberá utilizar en su desarrollo profesional como jefe de proyectos, o formando parte del equipo de proyecto. Para ello deberá adquirir una serie de habilidades relacionadas con la gestión, tanto de recursos materiales como humanos, y con la descomposición de tareas.

## CONOCIMIENTOS PREVIOS

## RELACIÓN CON OTRAS ASIGNATURAS DE LA MISMA TITULACIÓN

No se han especificado restricciones de matrícula con otras asignaturas del plan de estudios.

## OTROS TIPOS DE REQUISITOS

Como su nombre indica, la asignatura Ingeniería del Software II tiene una conexión muy directa con los conocimientos que se imparten en la Ingeniería del Software I, siendo recomendable haber cursado previamente dicha asignatura. Más concretamente, la asignatura se apoya en los siguientes conceptos que se supondrán ya conocidos:

- Especificación y modelización de requisitos del software



- Análisis del sistema (Modelado de casos de uso, clases, etc)
- Diseño del sistema (Modelado de diagrama de clases de diseño, diagramas de secuencia, diagramas de estado, etc.)
- Implementación del sistema a partir de los modelos de diseño.

## COMPETENCIAS / RESULTADOS DE APRENDIZAJE

-

G10 - 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 las competencias específicas establecidas.

G2 - Capacidad para dirigir las actividades objeto de los proyectos del ámbito de la informática de acuerdo con los conocimientos adquiridos según las competencias específicas establecidas.

G3 - 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.

G5 - 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 las competencias específicas establecidas.

G6 - Capacidad para concebir y desarrollar sistemas o arquitecturas informáticas centralizadas o distribuidas integrando hardware, software y redes de acuerdo con los conocimientos adquiridos según las competencias específicas establecidas.

G9 - 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.

R16 - Conocimiento y aplicación de los principios, metodologías y ciclos de vida de la ingeniería de software.

R18 - Conocimiento de la normativa y la regulación de la informática en los ámbitos nacional, europeo e internacional.

R1 - 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.

R2 - Capacidad para planificar, concebir, desplegar y dirigir proyectos, servicios y sistemas informáticos en todos los ámbitos, liderando su puesta en marcha y su mejora continua y valorando su impacto económico y social.

R3 - Capacidad para comprender la importancia de la negociación, los hábitos de trabajo efectivos, el



liderazgo y las habilidades de comunicación en todos los entornos de desarrollo de software.

SI3 - Capacidad para participar activamente en la especificación, diseño, implementación y mantenimiento de los sistemas de información y comunicación.

TI2 - Capacidad para seleccionar, diseñar, desplegar, integrar, evaluar, construir, gestionar, explotar y mantener las tecnologías de hardware, software y redes, dentro de los parámetros de coste y calidad adecuados.

## DESCRIPCIÓN DE CONTENIDOS

### 1. Metodologías de software: MÉTRICA versión 3

- 1.1. Introducción a MÉTRICA versión 3
- 1.2. Procesos principales
- 1.3. Actividades de los procesos
- 1.4. Interfaces del desarrollo de software
- 1.5. Participantes

### 2. Metodologías ágiles: Historias de usuario

- 2.1. Introducción a SCRUM
- 2.2. Definición de requisitos: Historias de usuario
  - 2.2.1. Métodos para la captación y representación
  - 2.2.2. Información necesaria
  - 2.2.3. Descripción de la historia
  - 2.2.4. Estimación y valoración de una historia
  - 2.2.5. Priorización de las historias
  - 2.2.6. Criterios de validación y calidad
  - 2.2.7. Historias técnicas

### 3. Gestión de la configuración del software

- 3.1. Introducción
- 3.2. Elementos de la gestión de configuraciones
- 3.3. Actividades de la gestión de configuraciones del SW
  - 3.3.1. Control de cambios
  - 3.3.2. Control de versiones
  - 3.3.3. Construcción del sistema
  - 3.3.4. Gestión de entregas
  - 3.3.5. Gestión de incidencias y problemas
- 3.4. Herramientas

### 4. Arquitecturas software

- 4.1. Introducción a las Arquitecturas Software
- 4.2. Estilos Arquitectónicos
- 4.3. Principios de Diseño en Arquitecturas Software
- 4.4. Herramientas y Técnicas de Modelado



## 5. Pruebas del software

- 5.1. Introducción
- 5.2. Técnicas de pruebas
- 5.3. Estrategias de prueba del software
  - 5.3.1. Pruebas unitarias
  - 5.3.2. Pruebas de integración
  - 5.3.3. Pruebas del sistema
  - 5.3.4. Pruebas de aceptación
- 5.4. Depuración del software

## 6. Medición del software

- 6.1. Medición de software
- 6.2. Medidas del producto
- 6.3. Medidas del proceso y de recursos
- 6.4. Metodologías y estándares para la medición

## 7. Evolución del software

- 7.1. Evolución del software
- 7.2. Dinámica de la evolución
- 7.3. Mantenimiento de software
- 7.4. Tipos de mantenimiento
- 7.5. Costes del mantenimiento
- 7.6. Predicción del mantenimiento
- 7.7. Métricas en el mantenimiento
- 7.8. Soluciones al mantenimiento
  - 7.8.1. Actividades principales y costes
  - 7.8.2. Ingeniería inversa, reingeniería y refactorización

## 8. Calidad del software

- 8.1. Introducción
- 8.2. Estándares de calidad
- 8.3. Calidad del producto
- 8.4. Calidad del proceso
  - 8.4.1. Aseguramiento de la calidad
  - 8.4.2. El modelo CMMI
  - 8.4.3. Modelo SPICE: estándar ISO/IEC 15504
  - 8.4.4. ISO 9000
- 8.5. Inspecciones y auditorías

## VOLUMEN DE TRABAJO (HORAS)

### ACTIVIDADES PRESENCIALES

Actividad	Horas
Teoría	30,00
Prácticas en aula	10,00
Laboratorio	20,00
<b>Total horas</b>	<b>60,00</b>

**ACTIVIDADES NO PRESENCIALES**

Actividad	Horas
Asistencia a otras actividades	0,00
Elaboración de trabajos individuales o en grupo	30,00
Estudio y trabajo autónomo	20,00
Preparación de clases	25,00
Preparación de actividades de evaluación	15,00
Resolución de casos prácticos	0,00
<b>Total horas</b>	<b>90,00</b>

**METODOLOGÍA DOCENTE**

El desarrollo de la asignatura se estructura en torno a los siguientes ejes: lecciones teóricas, sesiones de problemas y ejercicios prácticos, sesiones de laboratorio y actividades y trabajos de carácter práctico a realizar por parte del alumnado de forma individual y en pequeños grupos con objeto de plasmar los conocimientos adquiridos durante las lecciones de teoría y problemas.

- **Lecciones teóricas.** En las clases teóricas se desarrollarán los contenidos fundamentales de la asignatura proporcionando una visión global e integradora, analizando con mayor detalle los aspectos clave y de mayor complejidad. Para ello se utilizarán medios audiovisuales (como presentaciones, transparencia, pizarra y otros) fomentando en todo momento la participación del alumnado.

- **Ejercicios prácticos y problemas.** Las clases teóricas se complementarán con actividades prácticas y resolución de problemas en el aula con objeto de verificar y reforzar los conceptos básicos expuestos con anterioridad. Se planificarán varios tipos de actividades y problemas:

- **Actividades individuales:** Que serán realizadas de forma individual por el alumnado con objeto de valorar la comprensión individual de la materia. Se fomentará la participación del alumnado que se encargará de su explicación y corrección en la pizarra y será responsable de resolver las dudas que tengan el resto de compañeras y compañeros.
- **Actividades en pequeños grupos:** Los problemas planteados serán resueltos por grupos de alumnas y alumnos de forma conjunta con objeto de fomentar el trabajo en grupo y el análisis colectivo de diferentes planteamientos.

- **Sesiones de laboratorio.** Consistirán en la resolución de problemas relacionados con los contenidos teóricos por medio de programas informáticos con objeto de que el alumno conozca de forma práctica alguna de las herramientas CASE que implementa las metodologías de desarrollo y mantenimiento de software. Estas sesiones de laboratorio estarán organizadas en torno a pequeños grupos de trabajo formados como máximo por cinco personas.

- **Trabajo personal del estudiantado.** Consistirán en la realización de trabajos, cuestiones y problemas propuestos por el profesorado para fuera del aula, así como la preparación de clases con antelación, la lectura de textos recomendados y la preparación de exámenes (estudio). Estas actividades se realizarán de manera individual e intentan potenciar el trabajo autónomo. Parte de estas actividades tendrán carácter voluntario como formación complementaria de la asignatura.



- **Trabajo en pequeños grupos.** Al comienzo de la asignatura se establecerán grupos reducidos de 3 o 4 personas que se mantendrán fijos durante toda la asignatura. Estos grupos tendrán que realizar trabajos o resolver problemas fuera del aula, con objeto de complementar el trabajo individual y fomentar la capacidad de integración en grupos de trabajo. La naturaleza de estos trabajos será diversa, pero incluirá la presentación de una memoria del trabajo realizado, la exposición en clase de dicho trabajo y la participación como evaluadores de los trabajos presentados por otros grupos.

- **Tutorías:** El alumnado dispondrá de un horario de tutorías cuya finalidad es la de resolver problemas, dudas, orientación en trabajos, etc. El horario de dichas tutorías se indicará al inicio del curso académico. Además tendrán la oportunidad de aclarar algunas dudas mediante correo electrónico o foros de discusión mediante el uso de la herramienta "Aula Virtual" que proporciona la Universitat de València.

## EVALUACIÓN

La evaluación de conocimientos se realizará de dos formas:

### 1) EVALUACIÓN CONTINUA

Método recomendado para el alumnado. Se valoran los siguientes factores para obtener la nota final:

- 60% de conocimientos teóricos y problemas (TEO).
- 25% del laboratorio (LAB)
- 15% de los trabajos adicionales (TRA)

Será necesario que la nota final sea igual o superior a 5 para aprobar.

#### a) Conocimientos teóricos y problemas (TEO).

La nota de conocimientos teóricos y problemas se valoran en función de los siguientes factores:

- 80 % DE PRUEBAS INDIVIDUALES OBJETIVAS. Durante el curso se realizarán diferentes pruebas escritas sobre conocimientos teóricos y problemas. Será necesario obtener una nota superior o igual a 5 en cada prueba para que pueda compensar. En el examen final de la primera convocatoria tendrán que recuperarse aquellas partes que no se hayan superado en las pruebas parciales.
- 20 % DE PROBLEMAS. Se evaluarán los diferentes problemas que se propongan al alumnado, ya sea para realizar en clase o en casa. Esta actividad no es recuperable.

#### b) Laboratorio (LAB).



La nota de laboratorio se obtendrá de promediar la nota obtenida en las N sesiones prácticas. Para poder obtener la nota del laboratorio será necesario haber presentado todas las prácticas y haber asistido a un mínimo del 80% de las clases. Será necesario obtener una nota superior o igual a 4,5 en cada sesión de laboratorio para que esta parte se pueda compensar.

### **c) Trabajos adicionales (TRA).**

La nota de trabajos adicionales se obtendrá de promediar las notas obtenidas en cada uno de los trabajos por el peso asignado a cada trabajo. La nota de cada trabajo se obtendrá en función de la memoria escrita, y opcionalmente se podrá valorar la exposición pública del trabajo realizado.

## **2) SISTEMA DE EVALUACIÓN ÚNICA Y SEGUNDA CONVOCATORIA**

Este método se aplicará a cualquier alumna o alumno que, por un motivo razonado y admitido por el profesor, no pueda asistir con regularidad a las clases y en la segunda convocatoria. Se valoran los siguientes factores para obtener la nota final:

- 60% de conocimientos teóricos y problemas (TEO).

- 25% del laboratorio (LAB)

- 15% de los trabajos adicionales (TRA)

Será necesario que la nota final sea igual o superior a 5 para aprobar.

### **a) Conocimientos teóricos y problemas (TEO).**

La nota de conocimientos teóricos y problemas se valora mediante un único examen, no teniéndose en cuenta otros factores como los ejercicios y/o problemas realizados durante el curso. Será necesario obtener una nota superior o igual a 5 para que pueda compensar.

### **b) Laboratorio (LAB).**

La nota de laboratorio se obtendrá de promediar la nota obtenida en las sesiones prácticas, que deberán de haber sido entregadas, aunque no se haya asistido a las sesiones de laboratorio. Será necesario obtener una nota superior o igual a 4,5 en cada sesión de laboratorio para que esta parte se pueda compensar.

### **c) Trabajos adicionales (TRA).**

La nota de trabajos adicionales se obtendrá de promediar las notas obtenidas en cada uno de los trabajos por el peso asignado a cada trabajo. La nota de cada trabajo se obtendrá en función de la memoria escrita, y opcionalmente se podrá valorar la exposición pública del trabajo realizado.



En cualquier caso, la evaluación de la asignatura se hará de acuerdo con el Reglamento de evaluación y calificación de la Universitat de València para los títulos de grado y master aprobado por Consejo de Gobierno de 30 de mayo de 2017 (ACGUV 108/2017). Asimismo, la copia o plagio manifiesto de cualquier actividad que forma parte de la evaluación supondrá la imposibilidad de superar la asignatura, sometiéndose seguidamente a los procedimientos disciplinarios oportunos indicados en el *PROTOCOLO DE ACTUACIÓN ANTE PRÁCTICAS FRAUDULENTAS EN LA UNIVERSITAT DE VALÈNCIA* ([ACGUV 123/2020](#)).

## BIBLIOGRAFÍA

- Sommerville, I. (2016). Software Engineering (10th ed.). Pearson.
- Project Management Institute. A Guide to the Project Management Body of Knowledge. 4th edition, Project Management Institute (2008), ISBN: 19-33890517
- Sanchez, S; Sicilia, M.A; Rodriguez, D. Ingeniería del Software un enfoque desde la guía SWEBOK. Gaceta grupo editorial; ISBN: 978-8492812400
- Ministerio de Administraciones Públicas. (2001). Métrica versión 3. Madrid: Secretaría de Estado para la Administración Pública.
- Menzinsky, A.; López, G.; Palacio, J.; Scrum Master: Temario troncal 1. Versión 3.052 (junio 2021) Lubaris Info 4 Media S.L (<https://www.scrummanager.net/bok/>)
- Menzinsky, A.; López, G.; Palacio, J.; Historias de usuario: Ingeniería de requisitos ágiles. Versión 3.0 (Septiembre 2020) Lubaris Info 4 Media S.L (<https://www.scrummanager.net/bok/>)
- Bass, L., Clements, P., & Kazman, R. (2012). Software Architecture in Practice (3rd ed.). Addison-Wesley
- Pressman, R. S. (2014). Software Engineering: A Practitioner's Approach (8th Edition). McGraw-Hill Education.
- Piattini, M., García, F., Garzas, J., Genero, M. (2008). Medición y Estimación del Software: Técnicas y Métodos para Mejorar la Calidad y Productividad. Ra-Ma.
- Piattini, M., García, F., Caballero, I. (2006). Calidad de los Sistemas Informáticos. Ra-Ma.