

viu  
.es



# Guía didáctica

## Fundamentos de Ingeniería del Software

Título: Grado en Ingeniería Informática

Módulo: 2

Créditos: 6

ECTS Código:

18GIIN

Curso: 2018-

2019

**viu** | **Universidad**  
Internacional  
de Valencia

**Título:** Fundamentos de Ingeniería del Software

**Descripción:**

Esta asignatura aporta una visión general de la Ingeniería del Software y el papel que juega en el contexto de la Ingeniería Informática. Para ello se presentan los conceptos básicos y las principales características y técnicas a aplicar en las etapas de desarrollo de software: requisitos, análisis, diseño, implementación y pruebas. Además, una vez construido el software y desplegado es necesario también mantenerlo para completar el ciclo de vida del software, que va desde la definición hasta su retirada. Como resultado se espera capacitar al alumno con las habilidades y conocimientos básicos necesarios para trabajar como Ingeniero Software. Estos conocimientos estarán referenciados en el entorno de los estándares ISO que atañen al ciclo de vida del software, lo que dotará al alumno de una mayor visión global, y le proporcionará facilidades para trabajar en cualquier organización.

**Carácter:** Obligatoria

**Créditos ECTS:** 6

**Contextualización:**

Al igual que para la construcción o desarrollo en cualquier otra ingeniería, el desarrollo del software debe tratarse como un proyecto que habrá que gestionar adecuadamente. Por lo que la planificación y gestión de proyectos software es crucial. Es muy importante realizar un proceso de desarrollo controlado, pero no es suficiente. Una vez construido el software y desplegado es necesario mantenerlo para completar su ciclo de vida, que va desde la definición del mismo hasta su retirada.

**Modalidad de impartición:** Online

**Equipo docente:**

Profesora: Dr. / D. Ana Isabel Gómez Carretero  
Correo electrónico: [Anaisabel.Gomez@campusviu.es](mailto:Anaisabel.Gomez@campusviu.es)

**Temario:**

Concepto de Ingeniería del Software

El producto Software, propiedades y ciclo de vida

El proceso de desarrollo

Ingeniería de requisitos

Diseño e implementación de software

Planificación y gestión de proyectos

Validación y verificación de software

Mantenimiento de software

**Competencias:**

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 la resolución.

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.

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.

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.

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.

R4 - Capacidad para elaborar el pliego de condiciones técnicas de una instalación informática que cumpla los estándares y normativas vigentes.

R5 - Conocimiento, administración y mantenimiento sistemas, servicios y aplicaciones informáticas.

R8 - Capacidad para analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados.

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

## RESULTADOS DE APRENDIZAJE

Al finalizar esta asignatura se espera que el estudiante sea capaz de describir:

RA.1.- Describir el proceso de desarrollo de software y explicar las principales actividades realizadas y los productos obtenidos en el mismo.

RA.2.- Identificar los principales métodos de desarrollo de software y utilizar alguno de estos métodos.

RA.3.- Describir las técnicas concretas de especificación de requisitos y determinar la importancia de los requisitos como base del desarrollo de software.

RA.4.- Diferenciar entre planificación y gestión de proyectos y realizar la planificación inicial de supuestos prácticos.

## Actividades Formativas:

Actividad Formativa	Horas	Presencialidad
Clases expositivas	15	60
Resolución de ejercicios prácticos	20	30
Prácticas de laboratorios virtuales	25	20
Tutorías	20	0
Trabajo Autónomo	70	0

**Metodologías docentes:**

Clases teóricas impartidas como lecciones magistrales o exposiciones, en las que además de presentar el contenido de la asignatura, se explican los conceptos fundamentales y se desarrolla el contenido teórico.
Colección de tareas que el alumnado llevará a cabo a lo largo de toda la asignatura, entre las que podemos encontrar: análisis de casos, resolución de problemas, prácticas de laboratorios, comentarios críticos de textos, análisis de lecturas, etc.
Sesiones periódicas entre el profesorado y el alumnado para la resolución de dudas, orientación, supervisión, etc.
Trabajo tanto individual como grupal para la lectura crítica de la bibliografía, estudio sistemático de los temas, reflexión sobre problemas planteados, resolución de actividades propuestas, búsqueda, análisis y elaboración de información, investigación e indagación, así como trabajo colaborativo basado en principios constructivistas.

**Sistema de Evaluación:**

Sistemas de evaluación	Ponderación mínima	Ponderación máxima
Entrega de informes de problemas y ejercicios	10.0	10.0
Informes o memorias de prácticas de laboratorio	15.0	15.0
Trabajos o proyectos desarrollados en grupo o de forma individual	10.0	10.0
Participación activa en los debates, foros y otros medios	5.0	5.0
Evaluación final: se podrán realizar exámenes finales o parciales (que incluyan ítems de alternativas, de asociación, multi-ítems, interpretativos, preguntas de desarrollo breve o extenso), supuestos prácticos y/o análisis de casos, sobre el desarrollo y los resultados de las actividades propuestas.	60.0	60.0

**Bibliografía:**

Pressman, R. S. (2005) Ingeniería del Software: Un enfoque práctico. McGraw-Hill. ISBN: 9789701054734

Larson, E. W., & Gray, C. F. (2015). A Guide to the Project Management Body of Knowledge: PMBOK (®) Guide. Project Management Institute.

Garzías Parra, J., Juan A. Enríquez de S., & Irrazábal, E. (2012). Gestión ágil de proyectos software. Kybele Consulting.

Gutierrez de Mesa, J. A. (2008). Planificación y gestión de proyectos informáticos (No. 658: 681.3 658.05). e-libro, Corp.

Booch, G., Rumbaugh, J., Jacobson, I., (2006) El lenguaje Unificado de Modelado. Addison Wesley  
Schach, S.R. (2004) Classical and Object-Oriented Software Engineering. McGraw Hill

Larman, C. (2003) UML y patrones. Una introducción al análisis y diseño orientado a objetos y al proceso unificado. Prentice Hall.

Arlow, J. and Neustadt, I. (2006). UML 2 and the Unified Process. Practical Object Oriented Analysis and Design 2nd Edition. Addison-Wesley.