

## Guía Docente: Ingeniería del Software

DATOS GENERALES	
<b>Facultad</b>	Facultad de Ciencias y Tecnología
<b>Titulación</b>	Grado en Ingeniería Informática
<b>Plan de estudios</b>	2012
<b>Materia</b>	Ingeniería del Software, Sistemas de Información y Sistemas Inteligentes
<b>Carácter</b>	Obligatorio
<b>Período de impartición</b>	Primer Trimestre
<b>Curso</b>	Tercero
<b>Nivel/Ciclo</b>	Grado
<b>Créditos ECTS</b>	6
<b>Lengua en la que se imparte</b>	Castellano
<b>Prerrequisitos</b>	No se precisa

### DATOS DEL PROFESORADO

<b>Profesor Responsable</b>	Juan Agustín Fraile Nieto	<b>Correo electrónico</b>	juanagustin.fraile@ui1.es
<b>Área</b>	Tecnología Electrónica	<b>Facultad</b>	Facultad de Ciencias y Tecnología
<b>Perfil Profesional 2.0</b>	<p>Doctor en Informática con amplia experiencia como docente de asignaturas relacionadas con las TIC y como tutor de proyectos de grado/máster. He colaborado en proyectos de investigación del ámbito de los sistemas inteligentes, las tecnologías móviles (NFC) y la gestión de proyectos. Además, realizo tareas como consultor en procesos de implantación de software.</p> <p>Emprendedor, con gran iniciativa y voluntad de abordar proyectos formativos innovadores.</p> <p>Acreditado como profesor contratado doctor y profesor de universidad privada por la Agencia Nacional de Evaluación de la Calidad y Acreditación (ANECA).</p> <p><a href="#">About.me</a></p> <p><a href="#">LinKedin</a></p> <p><a href="#">Twitter</a></p>		

### CONTEXTUALIZACIÓN Y JUSTIFICACIÓN DE LA ASIGNATURA

<b>Asignaturas de la materia</b>	<ul style="list-style-type: none"> <li>• Bases de Datos</li> <li>• Dirección de Proyectos de Desarrollo de Software</li> <li>• Ingeniería del Software</li> <li>• Inteligencia Artificial</li> <li>• Interfaces de Usuario</li> <li>• Minería de Datos e IA Corporativa</li> </ul>
<b>Contexto y sentido de la asignatura en la titulación y perfil profesional</b>	<p>La ingeniería del software es una asignatura fundamental por su relación con un buen número de asignaturas y materias del grado. En esta asignatura se pretende mostrar al alumnado las distintas fases del ciclo de vida de un proyecto software, sus principios, técnicas y diferentes modelos. Se inicia al alumnado en las técnicas de análisis de requisitos de los clientes, el diseño de soluciones software de calidad, el diseño y realización de pruebas que verifiquen el correcto funcionamiento de las aplicaciones, ajustándose a los requisitos de análisis y diseño. Se muestran técnicas de mantenimiento del software así como métodos de planificación y control de proyectos software.</p> <p>Los contenidos de esta asignatura están relacionados con otras como la dirección de proyectos software, donde el conocimiento de estos contenidos es primordial. En distintas fases del proyecto la aplicación de técnicas y métodos vistos en otras asignaturas puede facilitar su comprensión, en ambos sentidos, como ocurre con las asignaturas de bases de datos o interfaces de usuario.</p>

**COMPETENCIAS QUE ADQUIERE EL ESTUDIANTE Y RESULTADOS DE APRENDIZAJE**

<p><b>Competencias de la asignatura</b></p>	<ul style="list-style-type: none"> <li>• CR02: 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.</li> <li>• CR16: Conocimiento y aplicación de los principios, metodologías y ciclos de vida de la ingeniería de software.</li> <li>• CE9: 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.</li> <li>• CT-01: Capacidad de análisis y síntesis: encontrar, analizar, criticar (razonamiento crítico), relacionar, estructurar y sintetizar información proveniente de diversas fuentes, así como integrar ideas y conocimientos</li> <li>• CT-04: Capacidad para la resolución de problemas</li> <li>• CR17: Capacidad para diseñar y evaluar interfaces persona computador que garanticen la accesibilidad y usabilidad a los sistemas, servicios y aplicaciones informáticas.</li> <li>• CR03: 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.</li> <li>• CR04: Capacidad para elaborar el pliego de condiciones técnicas de una instalación informática que cumpla los estándares y normativas vigentes.</li> </ul>
<p><b>Resultados de aprendizaje de la asignatura</b></p>	<ul style="list-style-type: none"> <li>• Conoce los patrones que se puede aplicar a un diseño, su estructura y clasificación y su importancia como herramienta para mejorar la calidad de un diseño.</li> <li>• Adquiere destreza en la identificación de los patrones aplicables a un determinado problema.</li> <li>• Conoce las técnicas de desarrollo dirigidas por modelos, sus ventajas e inconvenientes.</li> <li>• Distingue los conceptos de validación y verificación de requisitos.</li> <li>• Conoce los niveles de prueba del software, incluyendo las pruebas unitarias y de integración, las pruebas de validación, las de sistema y las de aceptación.</li> <li>• Es capaz de definir, desarrollar e implementar un plan de pruebas.</li> <li>• Conoce los aspectos específicos de las pruebas de software orientado a objetos.</li> <li>• Comprende el concepto de componente y su papel en el proceso de desarrollo de software.</li> <li>• Conoce arquitecturas específicas para el diseño de sistemas de información.</li> <li>• Conoce la importancia del mantenimiento y su integración en el proceso de desarrollo de software.</li> </ul>

**PROGRAMACION DE CONTENIDOS**

<p><b>Breve descripción de la asignatura</b></p>	<p>Esta asignatura, que constituye una de los grandes áreas conceptuales del Grado, describirá los patrones de diseño como base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces. Se hará un recorrido por las arquitecturas orientadas a componentes y servicios, así como por las técnicas de desarrollo dirigidas por modelos y el modelado de negocio.</p> <p>Será igualmente importante para construir un software de calidad que el alumnos conozca las distintas técnicas de verificación y validación del software y pruebas.</p>
<p><b>Contenidos</b></p>	<p><b>UD1: Conceptos básicos, modelos de ciclo de vida y gestión de proyectos</b></p> <ul style="list-style-type: none"> <li>• Software vs. Ingeniería del software</li> <li>• El ciclo de vida del software</li> <li>• Planificación de proyectos <i>software</i></li> <li>• Calendarización del proyecto</li> <li>• Técnica PERT (redes de precedencia)</li> <li>• Estimación de riesgos</li> </ul> <p><b>UD2: Análisis de requisitos de software</b></p> <ul style="list-style-type: none"> <li>• Obtención de requisitos software</li> <li>• Modelado de requisitos</li> <li>• Tipos de modelado</li> </ul> <p><b>UD3: Diseño de software</b></p> <ul style="list-style-type: none"> <li>• Principios de diseño</li> <li>• Diseño de datos</li> <li>• Diseño arquitectónico</li> <li>• Diseño de la interfaz</li> <li>• Diseño de componentes</li> </ul> <p><b>UD4: Calidad. Pruebas</b></p> <ul style="list-style-type: none"> <li>• Requisitos de calidad</li> <li>• Técnicas de prueba</li> <li>• Estrategia de pruebas de software</li> <li>• Diseño de casos de prueba</li> </ul> <p><b>UD5: Mantenimiento del software</b></p> <ul style="list-style-type: none"> <li>• El proceso de mantenimiento del software</li> <li>• Tipos de mantenimiento</li> <li>• Costes de mantenimiento</li> <li>• Problemas del mantenimiento del software</li> <li>• La reingeniería del <i>software</i></li> <li>• Ingeniería inversa</li> <li>• Reestructuración</li> <li>• Ingeniería hacia adelante</li> </ul> <p><b>UD6: Metodologías Ágiles: Scrum</b></p> <ul style="list-style-type: none"> <li>• Ciclo de desarrollo de una metodología ágil</li> <li>• Tipos de metodologías ágiles</li> <li>• Scrum framework</li> <li>• DevOps</li> </ul>

## METODOLOGÍA

### Actividades formativas

La evaluación continua de la asignatura **Ingeniería del software** se articula sobre cuatro tipos básicos de actividades: estudios de caso, actividades de contenidos teóricos, foros de debate, cuestionarios y trabajos colaborativos. En el desarrollo de cada actividad, en el aula, se establecen las características específicas de entrega, plazos, puntuación y cualquier otra información útil para su realización.

**Foros de debate:** actividad en la que se discutirá y argumentará acerca de diferentes temas relacionados con la asignatura y que servirán para guiar el proceso de descubrimiento inducido.

**Estudios de caso:** actividad en la que el alumno podrá llevar a cabo un aprendizaje contextualizado que le permitirá realizar un primer acercamiento a los diferentes temas de estudio.

**Cuestionarios:** cuestionario evaluable que servirán para poner a prueba los conocimientos adquiridos.

**Actividades de contenidos:** Al igual que el cuestionario, pone a prueba los conocimientos adquiridos mediante la resolución de ejercicios prácticos.

**Trabajo colaborativo:** en esta tarea se deberá reflexionar o investigar sobre alguno de los temas planteados y entablar un diálogo y debate con el resto de estudiantes para presentar un trabajo conjunto.

**Videotutorías:** sesiones en directo, que pueden visualizarse en diferido, donde se expone la resolución de las dudas presentadas al profesor previamente.

**Lectura crítica, análisis e investigación:** se trata de actividades en las que el alumno se acerca a los diferentes campos de estudio con una mirada crítica que le permite un acercamiento a la investigación.

### Prueba de Evaluación por Competencias (PEC)

En el caso de optar por la opción 2 de evaluación (PEC+ examen final), el estudiante tendrá que realizar la PEC. Esta prueba se define como una actividad integradora a través de la cual el estudiante deberá demostrar la adquisición de competencias propuestas en la asignatura, vinculadas principalmente al «saber hacer». Para ello hará entrega de un conjunto de evidencias en respuesta a los retos propuestos en esta prueba. La entrega se realizará antes de finalizar la asignatura.

## EVALUACIÓN

### Sistema evaluativo

El sistema de evaluación se basará en una selección de las pruebas de evaluación más adecuadas para el tipo de competencias que se trabajen. El sistema de calificaciones estará acorde con la legislación vigente (*Real Decreto 1125/2003, de 5 de septiembre, por el que se establece el sistema europeo de créditos y el sistema de calificaciones en las titulaciones universitarias de carácter oficial y de validez en todo el territorio nacional*).

El sistema de evaluación de la Universidad Isabel I queda configurado de la siguiente manera:

## Sistema de evaluación convocatoria ordinaria

### Opción 1. Evaluación continua

Los estudiantes que opten por esta vía de evaluación deberán realizar el **seguimiento de la evaluación continua (EC)** y podrán obtener hasta un **60 %** de la calificación final a través de las actividades que se plantean en la evaluación continua.

Además, deberán realizar un **examen final presencial (EX)** que supondrá el **40 %** restante. Esta prueba tiene una parte dedicada al control de la identidad de los estudiantes que consiste en la verificación del trabajo realizado durante la evaluación continua y otra parte en la que realizan diferentes pruebas teórico-prácticas para evaluar las competencias previstas en cada asignatura.

Para la aplicación de los porcentajes correspondientes, el estudiante debe haber obtenido una nota mínima de un 4 en cada una de las partes de las que consta el sistema de evaluación continua.

Se considerará que el estudiante supera la asignatura en la convocatoria ordinaria por el sistema de evaluación continua, siempre y cuando al aplicar los porcentajes correspondientes se alcance una calificación mínima de un 5.

### Opción 2. Prueba de evaluación de competencias

Los estudiantes que opten por esta vía de evaluación deberán realizar una **prueba de evaluación de competencias (PEC)** y un **examen final presencial (EX)**.

La **PEC** se propone como una prueba que el docente plantea con el objetivo de evaluar en qué medida el estudiante adquiere las competencias definidas en su asignatura. Dicha prueba podrá ser de diversa tipología, ajustándose a las características de la asignatura y garantizando la evaluación de los resultados de aprendizaje definidos. Esta prueba supone el 50 % de la calificación final.

El **examen final presencial**, supondrá el **50 %** de la calificación final. Esta prueba tiene una parte dedicada al control de la identidad de los estudiantes que consiste en la verificación del seguimiento de las actividades formativas desarrolladas en el aula virtual y otra parte en la que realizan diferentes pruebas teórico-prácticas para evaluar las competencias previstas en cada asignatura.

Al igual que con el sistema de evaluación anterior, para la aplicación de los porcentajes correspondientes el estudiante debe haber obtenido una puntuación mínima de un 4 en cada una de las partes de las que consta la opción de prueba de evaluación de competencias.

Se considerará que el estudiante supera la asignatura en la convocatoria ordinaria por el sistema de la prueba de evaluación de competencias siempre y cuando al aplicar los porcentajes correspondientes se alcance una calificación mínima de un 5.

### Características de los exámenes

Los exámenes constarán de 30 ítems compuestos por un enunciado y cuatro opciones de respuesta, de las cuales solo una será la correcta. Tendrán una duración de 90 minutos y

la calificación resultará de otorgar 1 punto a cada respuesta correcta, descontar 0,33 puntos por cada respuesta incorrecta y no puntuar las no contestadas. Después, con el resultado total, se establece una relación de proporcionalidad en una escala de 10.

### **Sistema de evaluación convocatoria extraordinaria**

Todos los estudiantes, independientemente de la opción seleccionada, que no superen las pruebas evaluativas en la convocatoria ordinaria tendrán derecho a una convocatoria extraordinaria.

La convocatoria extraordinaria completa consistirá en la realización de una **prueba de evaluación de competencias** que supondrá el **50 %** de la calificación final y un **examen final presencial** cuya calificación será el **50 %** de la calificación final.

Para la aplicación de los porcentajes correspondientes, el estudiante debe haber obtenido una nota mínima de un 4 en cada una de las partes de las que consta el sistema de evaluación de la convocatoria extraordinaria.

Los estudiantes que hayan suspendido todas las pruebas evaluativas en convocatoria ordinaria (evaluación continua o prueba de evaluación de competencias y examen final) o no se hayan presentado deberán realizar la convocatoria extraordinaria completa, como se recoge en el párrafo anterior.

En caso de que hayan alcanzado una puntuación mínima de un 4 en alguna de las pruebas evaluativas de la convocatoria ordinaria (evaluación continua o prueba de evaluación de competencias y examen final), se considerará su calificación para la convocatoria extraordinaria, debiendo el estudiante presentarse a la prueba que no haya alcanzado dicha puntuación o que no haya realizado.

En el caso de que el alumno obtenga una puntuación que oscile entre el 4 y el 4,9 en las dos partes de que se compone la convocatoria ordinaria (EC o PEC y examen), solo se considerará para la convocatoria extraordinaria la nota obtenida en la evaluación continua o prueba de evaluación de competencias ordinaria (en función del sistema de evaluación elegido), debiendo el alumno realizar el examen extraordinario para poder superar la asignatura.

Al igual que en la convocatoria ordinaria, se entenderá que el alumno ha superado la materia en convocatoria extraordinaria si, aplicando los porcentajes correspondientes, se alcanza una calificación mínima de un 5.

## **BIBLIOGRAFÍA Y OTROS RECURSOS**

### **Bibliografía básica**

- R.S. Pressma. *Ingeniería del software. Un enfoque práctico*. (7ª ed.). México: McGraw-Hill. 2010

Se trata de un libro de referencia básico en muchas Universidades y centros de formación, así como entre los profesionales. Su autor es internacionalmente reconocido en el ámbito de la Ingeniería del Software, y sus métodos utilizados en compañías de desarrollo de software de todo el mundo.

- T. Dimes,. *Conceptos Básicos de Scrum: Desarrollo de software Agile y manejo de proyectos Agile*. Babelcube Inc.. 2015

Se trata de un libro utilizado como referencia bibliográfica en el Grado de Informática de

muchas Universidades. Está dirigido tanto a estudiantes como a ingenieros que desarrollan su actividad en el ámbito del desarrollo software.

### Bibliografía complementaria

- D. Bolaños, A. Sierra y I. Alarcón. *Pruebas de software. Un análisis en profundidad y ejemplos prácticos*. Madrid: Pearson Education. 2007.
- G. Booch. *El Lenguaje Unificado de Modelado. Guía del Usuario*. (2ªed.). Madrid: Pearson Educación. 2006
- N. Fento S. Pfleeger. *Software Metrics*. Thomson Learning. 2002
- E. Hull, K. Jackson y J. Dick. *Requirements Engineering*. London: Springer. 2011
- ISO/IEC/IEEE. Software engineering – software life cycle processes – maintenance. Technical Report International Standard ISO/IEC 14764, IEEE. Std 14764-2006.
- S. Pfleeger. *Ingeniería del Software. Teoría y Práctica*. Prentice Hall. 2002
- R.G. Figueroa, C.J. Solís y A.A. Cabrera. Metodologías tradicionales vs. metodologías ágiles. *Universidad Técnica Particular de Loja, Escuela de Ciencias de la Computación*, 9. 2008
- M.G. Piattini. *Medición y estimación del software. Técnicas y métodos para mejorar la calidad y la productividad*. Madrid: RA-MA. 2008
- K. Pohl. *Requirements Engineering*. London: Springer. 2010
- S. Sánchez, M. Sicilia, y D. Rodríguez. *Ingeniería del software. Un enfoque desde la guía SWEBOK*. Madrid: Garceta. 2011

### Otros recursos

- Definición del estándar UML:  
<http://www.uml.org/>
- Guía SWEBOK:  
<http://www.computer.org/web/swebok>
- Manifiesto por el desarrollo ágil del software:  
<http://agilemanifesto.org/iso/es/principles.html>
- Métrica 3: \_  
[http://administracionelectronica.gob.es/pae\\_Home/pae\\_Documentacion/pae\\_Metodolog/pae\\_Metrica\\_v3.html#.VTDVvC6YGSo](http://administracionelectronica.gob.es/pae_Home/pae_Documentacion/pae_Metodolog/pae_Metrica_v3.html#.VTDVvC6YGSo)
- Software Engineering Code of Ethics: \_  
<http://www.computer.org/web/education/code-of-ethics>
- Software Engineering Resources:  
<http://www.rsqa.com/spi/index.html>
- Recursos Scrum:  
<https://www.scrumalliance.org/>
- Estándares IEEE:  
<http://standards.ieee.org/index.html>
- Estándares ISO:



<http://www.iso.org/iso/home/standards.htm>

### COMENTARIOS ADICIONALES

Definir la ingeniería del *software* es una tarea difícil, ya que no existe un consenso ni en la industria del desarrollo del *software* ni en el ámbito académico. Esto se debe a que es, por un lado, una rama nueva de la ingeniería que busca sistematizar de una manera racional las prácticas que aparecen en la industria del *software*.

La Ingeniería del *Software* no es una “bala de plata”, una herramienta milagrosa que hace que todo funcione bien pero sí que te ayudará a resolver todos esos problemas anteriores y quizá alguno que aún no te hayas imaginado.

Afronta esta materia con mente abierta y espíritu crítico. A partir de los conceptos, herramientas y definiciones básicas que asimiles debes ser capaz de realizar una planificación de un proyecto de desarrollo de *software* óptima que te ayude en la implementación de un producto de calidad.