

Guía docente de la asignatura

## Metodologías de Desarrollo Ágil (Especialidad Ingeniería del Software) (296114D)



Fecha de aprobación: 25/06/2024

<b>Grado</b>	Grado en Ingeniería Informática	<b>Rama</b>	Ingeniería y Arquitectura				
<b>Módulo</b>	Formación de Especialidad 2: Ingeniería del Software	<b>Materia</b>	Desarrollo y Gestión de Proyectos				
<b>Curso</b>	4º	<b>Semestre</b>	1º	<b>Créditos</b>	6	<b>Tipo</b>	Obligatoria

### PRERREQUISITOS Y/O RECOMENDACIONES

No es necesario que los alumnos tengan aprobadas asignaturas, materias o módulos previos como requisito indispensable para cursar este módulo. No obstante se recomienda la superación de los contenidos y adquisición de competencias de las materias de formación básica y de rama, teniendo especial importancia la superación de las materias “Fundamentos de Ingeniería del Software”, “Programación y Diseño Orientado a Objetos” y “Diseño de Interfaces de Usuario”.

### BREVE DESCRIPCIÓN DE CONTENIDOS (Según memoria de verificación del Grado)

- Desarrollo centrado en el usuario.
- Metodologías de desarrollo ágiles.
- El software libre y la empresa.
- Metodologías de desarrollo usadas en el software libre.
- Técnicas de evaluación de la calidad.
- Desarrollo dirigido por las pruebas.

### COMPETENCIAS ASOCIADAS A MATERIA/ASIGNATURA

#### 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, 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.
- 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.

- 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 Técnico en Informática.

### COMPETENCIAS TRANSVERSALES

- CT02 - Capacidad para tomar decisiones basadas en criterios objetivos (datos experimentales, científicos o de simulación disponibles) así como capacidad de argumentar y justificar lógicamente dichas decisiones, sabiendo aceptar otros puntos de vista.

### RESULTADOS DE APRENDIZAJE (Objetivos)

- Conocer los principales métodos de obtención de requisitos y las situaciones en las que pueden ser usados.
- Comprender el papel que juega el usuario en la ingeniería de requisitos y su posterior participación en el desarrollo del software.
- Conocer las técnicas de desarrollo centradas en el usuario y su aportación a la ingeniería del software.
- Entender la naturaleza de la filosofía del desarrollo ágil y los métodos asociados al mismo.
- Identificar y estudiar los diferentes esquemas de desarrollo ágiles y las técnicas y herramientas que proponen.
- Identificar las formas de estructurar un proyecto y las técnicas de gestión más adecuadas para el control de un proyecto de forma ágil.
- Analizar la importancia del capital humano en relación a las técnicas ágiles y su incorporación a un equipo de desarrollo.
- Conocer el software libre, sus tipos y características junto con su integración en la empresa.
- Conocer la estructura y los procesos de desarrollo en ámbitos del software libre y las actividades de emprender, implantar y liderar proyectos en este ámbito.
- Conocer las propiedades de calidad que afectan a todo producto y proceso de desarrollo software y las técnicas que podemos usar para su evaluación.
- Ser capaz de aplicar un proceso básico de desarrollo de software dirigido por las pruebas.
- Ser capaz de desarrollar, mantener y evaluar servicios y sistemas software que satisfagan todos los requisitos del usuario y se comporten de forma fiable y eficiente, sean asequibles de desarrollar y mantener y cumplan normas de calidad, aplicando las teorías, principios, métodos y prácticas de la Ingeniería del Software.
- Ser capaz de valorar las necesidades del cliente y especificar los requisitos software para satisfacer estas necesidades, reconciliando objetivos en conflicto mediante la búsqueda de compromisos aceptables dentro de las limitaciones derivadas del coste, del tiempo, de la existencia de sistemas ya desarrollados y de las propias organizaciones.
- Capacidad de identificar y analizar problemas y diseñar, desarrollar, implementar, verificar y documentar soluciones software sobre la base de un conocimiento adecuado de las teorías, modelos y técnicas actuales.

### PROGRAMA DE CONTENIDOS TEÓRICOS Y PRÁCTICOS

#### TEÓRICO



### Tema 1. Metodologías de desarrollo ágiles.

1. Etapas del desarrollo del software. Enfoque tradicional vs ágil.
2. Metodologías y enfoques ágiles. XP y SCRUM.
3. Herramientas ágiles.

### Tema 2. Desarrollo centrado en el usuario.

1. Ingeniería de requisitos.
2. Técnicas de elicitación de requisitos.
3. Metodologías centradas en el usuario.
4. Herramientas de apoyo.

### Tema 3. El software libre y su desarrollo.

1. El software libre y las tecnologías de desarrollo.
2. El software libre y la empresa. Modelo de negocio.

### Tema 4. Calidad, validación y verificación del Software.

1. Evaluación de la calidad y mejora del desarrollo.
2. Desarrollo dirigido por las pruebas. Técnicas de refactoring.

## PRÁCTICO

### Seminarios/Talleres:

- Taller: Introducción a las herramientas y técnicas ágiles a usar en prácticas.

### Prácticas de Laboratorio:

- Práctica 1. Desarrollo de una aplicación usando las técnicas descritas en clase de teoría. Desde su concepción inicial hasta su distribución.
- Práctica 2. Planificación, definición de negocio y estudio de viabilidad.

## BIBLIOGRAFÍA

### BIBLIOGRAFÍA FUNDAMENTAL

- Craig Larman. Agile and Iterative Development: A Manager's Guide. The Agile Software Development Series. Addison-Wesley Professional, 2004
- Alistair Cockburn. Agile Software Development. Addison-Wesley Professional, 2001
- Ken Schwaber. Agile, Project Management with SCRUM. Microsoft Press, 2009
- Carmen Lasa, et.al. Métodos Ágiles. Scrum, Kanban, Lean. Anaya Multimedia, 2017
- Mike Cohn. User Stories Applied. For Agile Software Development. Addison-Wesley, 2013
- Mike Cohn. Agile Estimating and Planning. Prentice Hall, 2006
- Kent Beck. Una explicación de la Programación Extrema: Aceptar el Cambio. Addison-Wesley Iberoamericana España, 2002
- Rober C. Martin. Desarrollo ágil esencial. Vuelta a las raíces. Anaya, 2020

### BIBLIOGRAFÍA COMPLEMENTARIA

- Andreas Rueping. Agile Documentation: A Pattern Guide to Producing Lightweight



- Documents for Software Projects. Wiley Software Patterns Series, 2003
- Stephen R. Palmer , John Felsing, A Practical Guide to Feature-Driven Development, Prentice Hall, 2002
  - Kieran Healy and Alan Schussman. The ecology of open source software development, 2003
  - Mike Cohn, Succeeding with Agile: Software Development Using Scrum, Addison-Wesley Professional, 2009
  - Tito Linz. Testing in Scrum: a guide for software quality assurance in the agile world. Rocky Nook Computing, 2014
  - Ryan Ripley, Todd Miller. Fixing your Scrum: Practical solutions to common Scrum problems. The Pragmatic Programmers, 2020
  - Jeff Gothelf, Josh Seiden. Lean UX: designing great products with agile teams. O'Reilly Media, 2016
  - Susheela Hooda (Ed). Agile software development: trends, changeless and applications. John Wiley & Sons, 2023.
  - Karl Fogel and Moshe Bar, Open Source Development with CVS, 3rd Edition; Paraglyph Press, 2003
  - Joseph Feller, Brian Fitzgerald , Understanding Open Source Software development, Addison-Wesley, 2002
  - Ian Sommerville. Ingeniería del Software. Editorial Addison Wesley, 2016
  - Sutcliffe, A. User-Centred Requirements Engineering. Theory and Practice., Springer, 2002
  - K.L., and Harbison, K., Lawrence , User-centered requirements: The Scenarios-Based Engineering Process, McGraw, : New Jersey, 1997
  - Gojko Adzir. Specification by Example: How Successful Teams Deliver the Right Software. Manning Publications Co, 2011
  - David, J. Anderson. Kanban, Cambio Evolutivo Exitoso para su Negocio de Tecnología. Blue Hola Press, 2010
  - Eric S. Raymond. The Cathedral and the Bazaar, Musings on Linux and Open Source by an Accidental Revolutionary. O'Reilly & Associates, 2001
  - Sommerville, I. Sawyer, P. Requirements Engineering. A good practice guide. Wiley, 2004
  - Fowler, M. et al. Refactoring: Improving the Design of Existing Code. Addison-Wesley, 2018
  - William C Wake. Extreme Programming Explored. Addison Wesley, 2002

## ENLACES RECOMENDADOS

- [Manifiesto por el Desarrollo Ágil de Software](#)
- [SCRUM](#)
- [eXtreme Programming \(XP\)](#)

## METODOLOGÍA DOCENTE

- MD01 - Lección Magistral (Clases Teóricas-Expositivas)
- MD02 - Actividades Prácticas (Resolución de Problemas, Resolución de Casos Prácticos, Desarrollo de Proyectos, Prácticas en Laboratorio, Taller de Programación, Aula de Informática, Prácticas de Campo).
- MD03 - Seminarios (Debates, Demos, Exposición de Trabajos Tutelados, Conferencias, Visitas Guiadas, Monografías).
- MD04 - Actividades no presenciales Individuales.



- MD05 - Actividades no presenciales Grupales.
- MD06 - Tutorías Académicas.

## EVALUACIÓN (instrumentos de evaluación, criterios de evaluación y porcentaje sobre la calificación final)

### EVALUACIÓN ORDINARIA

El sistema que se seguirá para la evaluación de esta asignatura es bajo la modalidad de evaluación continua, tal como se describe a continuación.

El sistema de calificación establece que el 100% de la nota final se reparte en: 50% de teoría y 50% prácticas. Para aprobar la asignatura es necesario tener una calificación numérica superior o igual a 5 puntos (sobre 10). La suma de las calificaciones de ambas partes solo se realizará en caso de tener en cada parte, una puntuación superior o igual a 5 puntos (sobre 10). La calificación final es la suma de las calificaciones obtenidas en teoría y prácticas. Si solo se supera una parte (teoría o prácticas), la calificación de la parte superada se mantiene hasta la siguiente convocatoria extraordinaria del curso académico actual.

Con respecto a la calificación detallada de cada una de las partes, se establece lo siguiente:

Los 5 puntos de la teoría se reparten de la siguiente forma:

- 50% corresponden a pruebas objetivas individuales realizadas en el aula.
- 50% correspondiente a trabajo individual o en grupo para la resolución de ejercicios, problemas, y/o trabajos, así como la participación en clase.

Los 5 puntos de prácticas se obtienen de la evaluación ponderada de cada una de las prácticas realizadas a lo largo del curso, se valorarán las entregas de los informes/memorias realizadas por los alumnos y las sesiones de evaluación.

Todo lo relativo a la evaluación se regirá por la [Normativa de evaluación y calificación de los estudiantes](#) vigente en la Universidad de Granada.

El sistema de calificaciones se expresará mediante calificación numérica de acuerdo con lo establecido en el art. 5 del R. D 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 validez en el territorio nacional.

### EVALUACIÓN EXTRAORDINARIA

La evaluación de los estudiantes en la convocatoria extraordinaria constará de dos pruebas de evaluación, una para la parte teórica y otra para la parte práctica, con las características que se indican a continuación:

- Evaluación de teoría: los estudiantes realizarán una prueba escrita que podrá constar de preguntas de teoría, preguntas tipo test y problemas relacionados con los aspectos teóricos de la asignatura.
- Evaluación de prácticas: los estudiantes realizarán una prueba escrita que constará de ejercicios y casos prácticos sobre los contenidos prácticos impartidos a lo largo del curso académico.

La ponderación de cada parte en la nota final será del 50 %, para aprobar la asignatura es necesario tener una calificación numérica superior o igual a 5 puntos (sobre 10). La suma de las calificaciones de ambas partes solo se realizará en caso de tener en cada parte, una puntuación superior o igual a 5 puntos (sobre 10).

Todo lo relativo a la evaluación se regirá por la [Normativa de evaluación y calificación de los estudiantes](#) vigente en la Universidad de Granada.

El sistema de calificaciones se expresará mediante calificación numérica de acuerdo con lo establecido en el art. 5 del R. D 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 validez en el territorio nacional.

### EVALUACIÓN ÚNICA FINAL

Se realizará una evaluación única final para aquellos estudiantes que no puedan acogerse a la evaluación continua y hayan solicitado dicho examen único final, alegando y acreditando las razones que le asisten para no poder seguir el sistema de evaluación continua; siempre de acuerdo a lo que se especifica en el Artículo 8 de la “Normativa de Evaluación y Calificación de los Estudiantes de la Universidad de Granada”.

La evaluación única final constará de dos pruebas de evaluación, una para la parte teórica y otra para la parte práctica, con las características que se indican a continuación:

- Evaluación de teoría: los estudiantes realizarán una prueba escrita que podrá constar de preguntas de teoría, preguntas tipo test y problemas relacionados con los aspectos teóricos de la asignatura.
- Evaluación de prácticas: los estudiantes realizarán una prueba escrita que constará de ejercicios y casos prácticos sobre los contenidos prácticos impartidos a lo largo del curso académico.

La ponderación de cada parte en la nota final será del 50 %, para aprobar la asignatura es necesario tener una calificación numérica superior o igual a 5 puntos (sobre 10). La suma de las calificaciones de ambas partes solo se realizará en caso de tener en cada parte, una puntuación superior o igual a 5 puntos (sobre 10)

Todo lo relativo a la evaluación se registrará por la [Normativa de evaluación y calificación de los estudiantes](#) vigente en la Universidad de Granada.

El sistema de calificaciones se expresará mediante calificación numérica de acuerdo con lo establecido en el art. 5 del R. D 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 validez en el territorio nacional.

### INFORMACIÓN ADICIONAL

El título de Graduado/a en Ingeniería Informática de la Universidad de Granada ha obtenido, con fecha 5 de junio de 2019, el sello Euro-Inf, otorgado por ANECA en colaboración con el Consejo General de Colegios Profesionales de Ingeniería en Informática (CCII) y con el Consejo General de Colegios Oficiales de Ingeniería Técnica en Informática (CONCITI). Esta acreditación garantiza el cumplimiento de criterios y estándares reconocidos por los empleadores españoles y del resto de Europa, de acuerdo con los principios de calidad, relevancia, transparencia, reconocimiento y movilidad contemplados en el Espacio Europeo de Educación Superior.

Régimen de asistencia a clase:

- La asistencia a clase de teoría no será obligatoria, aunque muy recomendable ya que la participación activa en clase y la entrega de ejercicios planteados por el profesor se tendrá en cuenta dentro del sistema de evaluación continua de la asignatura.
- La asistencia a las clases prácticas será obligatoria, salvo causa justificada.

Información de interés para estudiantado con discapacidad y/o Necesidades Específicas de Apoyo Educativo (NEAE): [Gestión de servicios y apoyos \(https://ve.ugr.es/servicios/atencion-social/estudiantes-con-discapacidad\)](https://ve.ugr.es/servicios/atencion-social/estudiantes-con-discapacidad).

