

Guía docente de la asignatura

Asignatura	TECNOLOGIAS PARA EL DESARROLLO DE SOFTWARE		
Materia	INGENIERIA DEL SOFTWARE		
Módulo	Tecnologías Específicas		
Titulación	GRADO EN INGENIERÍA INFORMÁTICA (545)		
Plan	545	Código	46928
Periodo de impartición	1 ^{er} . CUATRIMESTRE	Tipo/Carácter	Optativa-1 (Mención IS)
Nivel/Ciclo	GRADO	Curso	3º
Créditos ECTS	6 ECTS		
Lengua en que se imparte	CASTELLANO		
Profesor/es responsable/s	Yania Crespo		
Datos de contacto (E-mail, teléfono...)	TELÉFONO: 983 423000 ext. 5695 E-MAIL: yania@infor.uva.es		
Horario de tutorías	Véase www.uva.es → Grados → Grado en Ingeniería Informática → Tutorías , y buscar el nombre del profesor de la asignatura.		
Departamento	DEPARTAMENTO DE INFORMÁTICA		

1. Situación / Sentido de la Asignatura

1.1 Contextualización

En la actualidad, la demanda de software a gran escala, la transformación de espacios de producción en equipos de trabajo geográficamente distribuidos, la necesidad de reducir el tiempo de salida al mercado del producto desarrollado y la satisfacción del cliente han ido generando una multitud de tecnologías hoy día imprescindibles para el desarrollo de software. El objetivo de esta asignatura es dotar al alumno de técnicas y herramientas que permiten agilizar la producción de software con garantías de calidad.

1.2 Relación con otras materias

La asignatura se relaciona con todas las materias en las que el objetivo sea el desarrollo de software de grandes dimensiones, la actividad en entornos de producción ágiles y el trabajo en equipos distribuidos. En particular en esta titulación la asignatura se enmarca en la materia Tecnologías Específicas de la mención de Ingeniería del Software y se relaciona directamente con asignaturas de la propia materia y de las materias Fundamentos Básicos de Informática y Entorno Software tales como Fundamentos de Programación, Programación Orientada a Objetos, Fundamentos de Ingeniería del Software, Modelado de Sistemas Software, Diseño de Software, Desarrollo basado en Componentes y Servicios, Planificación y Gestión de Proyectos y Calidad del Software

1.3 Prerrequisitos

Aunque no se han establecido prerrequisitos, es recomendable que el alumno posea conocimientos básicos de programación, en particular haber cursado con aprovechamiento las asignaturas de Fundamentos de Programación y Programación Orientada a Objetos.

2. Competencias

2.1 Generales

De las Competencias Generales se hace énfasis en esta asignatura en CG3, CG5, CG8, CG9

Código	Descripción
CG3	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.
CG5	Capacidad para concebir, desarrollar y mantener sistemas, servicios y aplicaciones informáticas empleando los métodos de la ingeniería de software como instrumento para el aseguramiento de su calidad.
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

No se añaden las competencias transversales.

2.2 Específicas

Específicas de Mención de Ingeniería del Software

Código	Descripción
IS1	Capacidad para 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 de Software
IS3	Capacidad de dar solución a problemas de integración en función de las estrategias, estándares y tecnologías disponibles.
IS4	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.

3. Objetivos

- Conocer y saber utilizar sistemas de gestión de configuración de software.
- Conocer y saber utilizar técnicas y herramientas de generación incremental y automática de documentación.
- Conocer y saber utilizar entornos de desarrollo para sistemas hardware/software, middleware y distribuidos.

- Conocer y saber utilizar técnicas y herramientas para el diseño y ejecución de pruebas.

Estos objetivos fundamentales se acompañan de otros colaterales según el enfoque se sigue la asignatura:

- Conocer y aplicar técnicas de desarrollo basado en frameworks.
- Conocer y aplicar prácticas ágiles de desarrollo de software.
- Ser capaz de aplicar las técnicas básicas del desarrollo de software dirigido por pruebas (Test Driven Development TDD).
- Ser capaz de aplicar técnicas para la Integración y entrega continua
- Conocer y aplicar las operaciones más habituales de refactoring

4. Tabla de dedicación del estudiante a la asignatura

ACTIVIDADES PRESENCIALES	HORAS	ACTIVIDADES NO PRESENCIALES	HORAS
Clases teórico-prácticas (T/M)	25	Estudio y trabajo autónomo individual	45
Clases prácticas de aula (A)		Estudio y trabajo autónomo grupal	45
Laboratorios (L)	28		
Prácticas externas, clínicas o de campo			
Seminarios (S)			
Tutorías grupales (TG)	5		
Evaluación (fuera del periodo oficial de exámenes)	2		
Total presencial	60	Total no presencial	90

5. Bloques temáticos

Bloque 1: Bloque único

Carga de trabajo en créditos ECTS:

a. Contextualización y justificación

En esta primera implantación de la asignatura se realizará como un único bloque temático

b. Objetivos de aprendizaje

Todos los objetivos de la asignatura

c. Contenidos

Los contenidos generales

1. Tecnologías de construcción en la ingeniería de software.
2. Tecnologías para sistemas hardware-software, middleware y distribuidos.
3. Entornos de desarrollo.

Se desarrollan en los siguientes Temas desglosados:



Tema 1 Desarrollo basado en frameworks

Tipos, técnicas y ejemplos prácticos

frameworks de aplicación vs frameworks de dominio

frameworks de caja blanca vs frameworks de caja negra. Caja gris

Técnicas:

método plantilla

anotaciones

lenguajes específicos de dominio

XML vs JSON

Ejemplos prácticos

JAXP: The Java API for XML Processing

JSR: The Java API for JSON Processing

JUnit

Tema 2 Prácticas Ágiles y TDD. El ciclo Red-Green-Refactor

Programación por pares.

El ciclo Red-Green-Refactor

Red: TDD (Test Driven Development)

Pruebas de Software

Pruebas unitarias con JUnit

Pruebas en aislamiento mediante Mock Objects

Otros casos especiales: Pruebas de aplicaciones que acceden a BBDD, pruebas de documentos XML

Tema 3 Green: Desarrollo y Depuración.

Gestión de la Configuración

Autodocumentación - javadoc

Repositorios

Control de Versiones

Configuración de Proyectos y Gestión de Dependencias: ant vs maven

Integración y entrega continua

Cobertura y monitorización

Tema 4. Refactor:

Las Ws de Refactoring (What? When? Where? How?)

Operaciones de Refactoring más utilizadas

Encapsulate field

Renames

Moves

Extract Method

Extract Interface

Extract Superclass

Use supertype where possible

Eclipse Refactoring Scripts

Las prácticas se realizarán con Eclipse y otras herramientas como plug-ins de Eclipse, bibliotecas, etc.

Temario de prácticas se resume en las herramientas a utilizar para cada parte de la teoría

Eclipse



Java

JAXP, JSR

JUnit

EasyMock/mockito

git

bitbucket/github

ant vs maven

Hudson/Jenkins

log4j

EclEmma vs jcoverage

Eclipse Refactor Menu

d. Métodos docentes

Ver Anexo en el punto 8 de esta guía

e. Plan de trabajo

Ver cronograma en el punto 9 de esta guía.

f. Evaluación

Ver tabla y criterios en el punto 7 de esta guía.

g. Bibliografía básica

- Daniel Bolaños Alonso Almudena Sierra Alonso, **Pruebas de Software y JUnit**, Pearson Educación, 2008. ISBN 978-84-8322-354-3
- Beck, K., **Test driven development by example**, Addison-Wesley 2002, ISBN 978-0321146533
- Duvall, P.; Matyas, S.; Glover, A., **Continuos integration: Improving Software Quality and Reducing Risk**, Addison Wesley Signature Series by Martin Fowler, 2007, 978-0321336385
- Humble, J.; Farley, D., **Continuos delivery: Reliable Software Releases through Build, Test, and Deployment Automation**, Addison Wesley Signature Series by Martin Fowler, 2010, ISBN 978-0321601919

- Fowler, M. et al, **Refactoring: improving the design of existing code**, Addison-Wesley, 1999, ISBN 978-0201485677

h. Bibliografía complementaria

- Ble, C et al, **Diseño ágil con TDD**, 2009, Creative Common License. ISBN: 978-1-4452-6471-4, on line: <http://www.dirigidoportests.com/el-libro/>
- Moser, M. and O'Brien, T. **The Hudson book**. 2011. Oracle inc. wiki.eclipse.org/The_Hudson_Book
- Martin, R., **Clean Code. A handbook of Agile Software Craftsmanship**. Prentice Hall, 2008, ISBN: 978-0-13-235088-4
- Hunt, A., Thomas, D., **The Pragmatic Programmer**. Addison Wesley. 2010. ISBN 0-201-61622-X. <https://progprog.com/book/tpp/the-pragmatic-programmer>
- Swicegood, T., **Pragmatic Version Control using Git**. The Pragmatic Starter Kit - Volume I, 2008 The Pragmatic Programmers ISBN: 978-1-93435-615-9. <https://progprog.com/book/tsgit/pragmatic-version-control-using-git>
- Hunt, A., Thomas, D., **Pragmatic Unit Testing. In Java with JUnit**. The Pragmatic Starter Kit - Volume II, 2003 The Pragmatic Programmers. ISBN 0-9745140-1-2. <https://progprog.com/book/utj/pragmatic-unit-testing-in-java-with-junit>
- Clark, M., **Pragmatic Project Automation. How to build, deploy and monitor Java Applications**. The Pragmatic Starter Kit - Volume III. 2004 The Pragmatic Programmers. ISBN 0-9745140-3-9. <https://progprog.com/book/auto/pragmatic-project-automation>

Páginas web de interés

- <http://git-scm.com/book>
- <https://confluence.atlassian.com/display/BITBUCKET/Bitbucket+Documentation+Home>
- <http://www.sonatype.com/resources/books/maven-the-complete-reference>
- <http://www.sonatype.com/resources/books/maven-by-example>
- <http://www.eclEmma.org/index.html>
- <http://c2.com/cgi/wiki?CodeCoverageTools>
- <http://logging.apache.org/log4j/2.x/manual/index.html>

i. Recursos necesarios

Las herramientas de programación serán de libre distribución y los utilizarán una máquina virtual proporcionada por el centro en la cual serán los encargados de instalar y configurar las herramientas necesarias para las tareas a desarrollar.

Plataforma Moodle de la ETSII.

6. Temporalización (por bloques temáticos)

BLOQUE TEMÁTICO	CARGA ECTS	PERIODO PREVISTO DE DESARROLLO
-----------------	------------	--------------------------------

Bloque 1: Bloque único	6 ECTS	Semanas 1 a 15
------------------------	--------	----------------

7. Sistema de calificaciones – Tabla resumen

3 tests al final de los Temas 1, 2 y 3 del 10% cada uno

3 Entregas de prácticas por parejas del 15% cada una

Práctica 1 Desarrollo basado en frameworks

Práctica 2 Red

Práctica 3 Green

Opcionalmente a elegir por el alumno

a) o bien examen

b) o bien entrega de una cuarta práctica individual que abarca el ciclo completo Red-Green-Refactor

Las opciones a) o b) harían el 25% restante.

INSTRUMENTO/PROCEDIMIENTO	PESO EN LA NOTA FINAL	OBSERVACIONES
Examen tipo test sobre el tema 1	10%	
Examen tipo test sobre el tema 2	10%	
Examen tipo test sobre el tema 3	10%	
Entrega de la primera práctica	15%	Práctica en parejas
Entrega de la segunda práctica	15%	Práctica en parejas
Entrega de la tercera práctica	15%	Práctica en parejas
Examen final o cuarta entrega	25%	El alumno elegirá entregar una cuarta práctica el día del examen final de la asignatura o realizar un examen de problemas. Cualquiera de las dos opciones representa el 25% de la asignatura. La cuarta entrega de prácticas en caso de elegirse es individual. La fecha de entrega de esta cuarta práctica opcional es la misma fecha del examen ordinario.

CRITERIOS DE CALIFICACIÓN

- **Convocatoria ordinaria:** Suma ponderada de los cuestionarios (30%), prácticas en parejas (45%) y examen o entrega de práctica individual (25%), debiendo obtener una suma igual o mayor a 5.
- **Convocatoria extraordinaria:** Para la convocatoria extraordinaria se mantendrá la ponderación de las calificaciones de la convocatoria ordinaria en las mismas condiciones. En convocatoria extraordinaria podrá optarse a realizar un examen por el 100% de la asignatura en el cual se evaluarán todos los contenidos y supuestos prácticos. El alumno interesado deberá notificarlo al profesor con fecha límite una semana antes del examen en convocatoria extraordinaria

8. Anexo: Métodos docentes

Actividad	Metodología
Clase de teoría	<ul style="list-style-type: none">• Clase magistral participativa• Estudio de casos en aula• Resolución de problemas

Clase práctica	<ul style="list-style-type: none">• Clase magistral participativa• Realización en grupos de dos personas de tres prácticas en las que se utilicen adecuadamente las técnicas presentadas en la asignatura• Opcionalmente realización de una práctica individual
-----------------------	---

9. Anexo: Cronograma de actividades previstas

Semana	Fecha	Teoría	Prácticas	Entrega Trabajos	Evaluación
1		Tema 1			
2		Tema 1			
3		Tema 1			
4		Tema 1			
5		Tema 2			Test 1
6		Tema 2		Práctica 1	
7		Tema 2			
8		Tema 2			
9		Tema 3			Test 2
10		Tema 3		Práctica 2	
11		Tema 3			
12		Tema 3			
13		Tema 4			Test 3
14		Tema 4		Práctica 3	
15		Tema 4			
Día del examen				Examen o entrega de práctica 4	Examen o entrega de práctica 4

Nota: Las fechas indicadas en esta tabla para pruebas y entregas son aproximadas. Tanto pruebas como entregas serán convocadas con la suficiente antelación mediante la plataforma Moodle de la escuela. Por otra parte, el calendario concreto podrá variar de acuerdo a la coordinación entre asignaturas, evitando grandes acumulaciones de carga en semanas puntuales. Todo esto se avisará en la plataforma Moodle de la escuela.