



GUÍA DOCENTE 2015-2016

Validación y Pruebas «Validating and testing»

1. Denominación de la asignatura:

Validación y Pruebas «Validating and testing»

Titulación

Grado en Ingeniería Informática «Degree in computer science engineering»

Código

6376

2. Materia o módulo a la que pertenece la asignatura:

Mención Ingeniería del Software «Software Engineering mention»

3. Departamento(s) responsable(s) de la asignatura:

Ingeniería Civil, LSI «Civil Engineering, Computer languages and systems»

4.a Profesor que imparte la docencia (Si fuese impartida por mas de uno/a incluir todos/as) :

Pedro Renedo Fernandez

4.b Coordinador de la asignatura

Carlos Pardo Aguilar

5. Curso y semestre en el que se imparte la asignatura:

4º Curso - 7º Semestre

6. Tipo de la asignatura: (Básica, obligatoria u optativa)

Optativa



7. Requisitos de formación previos para cursar la asignatura:

Requisitos recomendados: Obligatorias de la materia, Diseño y Mantenimiento del Software

8. Número de créditos ECTS de la asignatura:

6

9. Competencias que debe adquirir el alumno/a al cursar la asignatura

Específica. 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 del Software.

Específica. IS3. Capacidad de dar solución a problemas de integración en función de las estrategias, estándares y tecnologías disponibles.

Específica. 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.

General. 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.

General. 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.

General. 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.

General. 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.



10. Programa de la asignatura

10.1- Objetivos docentes
Conocer los conceptos fundamentales de la teoría de validación del software.
Saber aplicar las técnicas básicas para la prueba del software.
Aplicar procedimientos de pruebas a software de distinta granularidad.
Probar sistemas software utilizando herramientas adecuadas.
10.2- Unidades docentes (Bloques de contenidos)
Unidad A. Teoría general de validación y pruebas Tema 1. Introducción a la validación y pruebas del Software
Unidad B. Técnicas y estrategias de pruebas Tema 2. Pruebas de caja negra Tema 3. Pruebas de caja blanca Tema 4. Pruebas de integración, sistema y aceptación Tema 5. Desarrollo dirigido por pruebas
Unidad C. Herramientas de pruebas Tema 6. Frameworks de Automatización de Pruebas Tema 7. Bibliotecas para Pruebas del Software
Unidad D. Pruebas en diferentes paradigmas Tema 8. Técnicas dependientes del modelo de programación



10.3- Bibliografía

BIBLIOGRAFÍA BÁSICA

Glenn Myers, (1984) El arte de probar el software, Primera edición, El Ateneo,
Kaner, C.; Bach, J. & Pettichord, B., (1999) Lessons Learned in Software Testing. A
Context-Driven Approach, 1st Edition, Publishing, W. C. (ed.) John Wiley & Sons,
Inc,

Robert Binder, (2000) Testing Object-Oriented System. Models, Patterns and Tools,
1st Edition, Addison-Wesley,

BIBLIOGRAFÍA COMPLEMENTARIA

Beck, K, (1999) Extreme Programming Explained: Embrace Change, 1st Edition,
Addison-Wesley Professional,

Bolaños, D.; Alonso, A. S. & Rodríguez, M. I. A., (2008) Pruebas de Software y JUnit.
Un análisis en profundidad y ejemplos prácticos, Primera Edición, Pearson Educación,

Link, J., (2002) Unit Testing in Java. How Tests Drive the Code, 1st Edition, Morgan
Kaufmann Publishers Inc,

Rainsberger, J. B., (2005) JUnit Recipes, 1st Edition, 2005,

**11. Metodología de enseñanza y aprendizaje y su relación con las competencias
que debe adquirir el estudiante:**

Metodología	Competencia relacionada	Horas presenciales	Horas de trabajo	Total de horas
Clases teóricas (grupo grande)		20	0	20
Preparación de trabajos, su exposición y debate		4	16	20
Estudio personal		0	35	35
Clases prácticas en laboratorio (grupo pequeño)		24	0	24
Implementación de programas y documentación asociada		0	45	45
Pruebas de evaluación		6	0	6
Total		54	96	150



12. Sistemas de evaluación:

Es necesario sacar un mínimo de un 40% en cada procedimiento

La calificación de trabajo exposición y debate no es recuperable

Procedimiento	Peso primera convocatoria	Peso segunda convocatoria
Desarrollo de un proyecto en equipo utilizando pruebas.	25 %	25 %
Desarrollo de un proyecto dirigido por pruebas.	25 %	25 %
Trabajo, exposición y debate.	25 %	25 %
Prueba escrita de evaluación.	25 %	25 %
Total	100 %	100 %

Evaluación excepcional:

En caso de evaluación excepcional se sustituirán las pruebas por otras equivalentes

13. Calendarios y horarios:

<http://www.ubu.es/informatica>

El calendario académico de aprobado por Junta de la Escuela Politécnica Superior, los horarios y el calendario de exámenes se puede consultar en la página de la titulación
«The academic calendar approved by the Council of Polytechnic School, timetables and exam schedule can be found on the title page»

14. Idioma en que se imparte:

Español (con alguna bibliografía en inglés)