



Guía docente				
Datos Identificativos				2014/15
Asignatura (*)	Validación y Verificación del Software	Código	614G01053	
Titulación	Grao en Enxeñaría Informática			
Descritores				
Ciclo	Periodo	Curso	Tipo	Créditos
Grado	1º cuatrimestre	Cuarto	Obligatoria	6
Idioma	Castellano			
Prerrequisitos				
Departamento	Computación			
Coordinador/a	Castro Souto, Laura Milagros	Correo electrónico	laura.milagros.castro.souto@udc.es	
Profesorado	Cabalar Fernandez, Jose Pedro	Correo electrónico	pedro.cabalar@udc.es	
	Castro Souto, Laura Milagros		laura.milagros.castro.souto@udc.es	
	Perez Vega, Gilberto		gilberto.pvega@udc.es	
Web	moodle.udc.es			
Descripción general	<p>Esta asignatura busca dominar as alternativas actuais da enxeñaría de software para a validación e verificación do software, mediante o:</p> <ul style="list-style-type: none"> <li>- coñecemento de técnicas e ferramentas de validación de software funcionais e non funcionais a todos os niveles (unidade, integración, sistema);</li> <li>- coñecemento de técnicas e ferramentas de razonamiento automático; e</li> <li>- o coñecemento de técnicas e ferramentas de verificación formal.</li> </ul>			

Competencias de la titulación	
Código	Competencias de la titulación
A28	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.
B1	Capacidad de resolución de problemas
B2	Trabajo en equipo
B3	Capacidad de análisis y síntesis
B4	Capacidad para organizar y planificar
C2	Dominar la expresión y la comprensión de forma oral y escrita de un idioma extranjero.
C3	Utilizar las herramientas básicas de las tecnologías de la información y las comunicaciones (TIC) necesarias para el ejercicio de su profesión y para el aprendizaje a lo largo de su vida.
C4	Desarrollarse para el ejercicio de una ciudadanía abierta, culta, crítica, comprometida, democrática y solidaria, capaz de analizar la realidad, diagnosticar problemas, formular e implantar soluciones basadas en el conocimiento y orientadas al bien común.
C6	Valorar críticamente el conocimiento, la tecnología y la información disponible para resolver los problemas con los que deben enfrentarse.
C7	Asumir como profesional y ciudadano la importancia del aprendizaje a lo largo de la vida.
C8	Valorar la importancia que tiene la investigación, la innovación y el desarrollo tecnológico en el avance socioeconómico y cultural de la sociedad.

Resultados de aprendizaje			
Competencias de materia (Resultados de aprendizaje)	Competencias de la titulación		
	Capacidad de identificar y analizar problemas y diseñar, desarrollar, implementar, verificar y documentar soluciones software sobre la base de un conocimiento apropiado de las teorías, modelos y técnicas actuales.	A28	B1 B2 B3 B4



Contenidos	
Tema	Subtema
Bloque I: Validación de software	I.1 Especificación, diseño y ejecución de pruebas I.1.1. Niveles y tipos de pruebas I.1.2. Propiedades y trazabilidad de requisitos I.2 Gestión de pruebas: planificación, evaluación, métricas y revisiones
Bloque II: Métodos formales y razonamiento automático	II.1 Introducción: Deducción natural y cálculo de secuentes II.2 Prueba automática utilizando PVS II.3 ¿Qué es y para que se utiliza un probador de teoremas? II.4 Lenguaje de especificación de PVS: Tipos, expresiones, teorías, subtipado II.5 Probador PVS: tácticas, recursión, razonamiento ecuacional
Bloque III: Comprobación por modelos (model checking)	III.1 Introducción a las lógicas modales temporales III.2 Especificación de propiedades: deadlocks, safety, liveness, fairness III.3 Funcionamiento de un comprobador por modelos III.4 Introducción al manejo de una herramienta de model checking

Planificación			
Metodologías / pruebas	Horas presenciales	Horas no presenciales / trabajo autónomo	Horas totales
Sesión magistral	21	26.25	47.25
Prácticas de laboratorio	14	35	49
Trabajos tutelados	7	7	14
Prueba objetiva	3	31.5	34.5
Atención personalizada	5.25	0	5.25

(\*) Los datos que aparecen en la tabla de planificación són de carácter orientativo, considerando la heterogeneidad de los alumnos

Metodologías	
Metodologías	Descripción
Sesión magistral	Clases de teoría donde se imparten los contenidos del temario.
Prácticas de laboratorio	Elaboración de trabajos prácticos en el laboratorio.
Trabajos tutelados	Resolución de trabajos tutelados planteados y resueltos en horario de tutorías de grupos reducidos.
Prueba objetiva	Examen escrito.

Atención personalizada	
Metodologías	Descripción
Prueba objetiva Trabajos tutelados Sesión magistral Prácticas de laboratorio	Resolución de dudas de teoría y/o prácticas, trabajos tutelados, etc. en horario de tutorías de cada profesor.

Evaluación		
Metodologías	Descripción	Calificación
Prueba objetiva	Examen escrito que computa un máximo de 4 puntos sobre la nota final. Se requiere un mínimo de 2 puntos para superar la asignatura.	40
Trabajos tutelados	Resolución y participación en trabajos tutelados en horario de tutorías de grupos reducidos. Computa un máximo de 2 puntos de la nota final. Su realización no es obligatoria para superar la asignatura.	20



Prácticas de laboratorio	Entrega e defensa de traballos prácticos de laboratorio. Comuta hasta un máximo de 4 puntos de la nota final. Su realización no es obligatoria para superar la asignatura.	40
--------------------------	--	----

### Observaciones evaluación

En la convocatoria de segunda oportunidad, la prueba objetiva irá acompañada de una actividad de evaluación de las prácticas de laboratorio.

### Fuentes de información

<b>Básica</b>	<ul style="list-style-type: none"><li>- Peter Farrell-Vinay (2008). Manage software testing. Auerbach</li><li>- Mordechai Ben-Ari (2012). Mathematical Logic for Computer Science. Springer</li><li>- Mordechai Ben-Ari (2001). Mathematical Logic for Computer Science. Springer</li><li>- Ron Patton (2001). Software testing. Sams</li><li>- Kent Beck (2002). Test Driven Development (By Example). Addison-Wesley</li><li>- Gerard J. Holzmann (2003). The SPIN model checker: primer and reference manual. Addison-Wesley</li><li>- Zohar Manna and Amir Pnueli (1995). The Temporal Logic of Reactive and Concurrent Systems. Safety. Springer</li><li>- Zohar Manna and Amir Pnueli (1991). The Temporal Logic of Reactive and Concurrent Systems. Specification. Springer</li></ul>
<b>Complementaria</b>	

### Recomendaciones

#### Asignaturas que se recomienda haber cursado previamente

Proyectos de Desarrollo Software/614G01226

#### Asignaturas que se recomienda cursar simultáneamente

Representación del Conocimiento y Razonamiento Automático/614G01036

Teoría de la computación/614G01039

Metodologías de Desarrollo/614G01051

#### Asignaturas que continúan el temario

Diseño Software/614G01015

Concurrencia y Paralelismo/614G01018

Proceso Software/614G01019

Arquitectura del Software/614G01221

Ingeniería de Requisitos/614G01222

Aseguramiento de la Calidad/614G01223

#### Otros comentarios

(\*) La Guía Docente es el documento donde se visualiza la propuesta académica de la UDC. Este documento es público y no se puede modificar, salvo cosas excepcionales bajo la revisión del órgano competente de acuerdo a la normativa vigente que establece el proceso de elaboración de guías