



Guía docente				
Datos Identificativos				2019/20
Asignatura (*)	Validación y Verificación del Software		Código	614G01225
Titulación	Grao en Enxeñaría Informática			
Descriptorios				
Ciclo	Periodo	Curso	Tipo	Créditos
Grado	2º cuatrimestre	Curso adap. Ing. Téc. Informática	Obligatoria	6
Idioma	Castellano			
Modalidad docente	Presencial			
Prerrequisitos				
Departamento	Computación			
Coordinador/a		Correo electrónico		
Profesorado		Correo electrónico		
Web	guiadocente.udc.es/guia_docent/index.php?centre=614&ensenyament=614G01&assignatura=614G01053&any_academic=2017_18&			
Descripción general	Esta asignatura busca dominar las alternativas actuales de la ingeniería de software para la validación y verificación del software, mediante: - el conocimiento de técnicas y herramientas de validación de software funcionales y no funcionales a todos los niveles (unidad, integración, sistema); - el conocimiento de técnicas y herramientas de razonamiento automático; y - el conocimiento de técnicas y herramientas de verificación formal.			

Competencias del título	
Código	Competencias del título
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
B3	Capacidad de análisis y síntesis
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.
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			
Resultados de aprendizaje		Competencias del título	
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 B3 C2 C3 C6 C7 C8

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 qué 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	Competencias	Horas presenciales	Horas no presenciales / trabajo autónomo	Horas totales
Sesión magistral	B3 C2 C7 C8	21	26.25	47.25
Prácticas de laboratorio	A28 B1 B3 C2 C3 C6	14	35	49
Trabajos tutelados	A28 B1 B3 C2 C3 C6	7	7	14
Prueba objetiva	B1 B3 C6	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
Sesión magistral Prácticas de laboratorio Trabajos tutelados Prueba objetiva	Resolución de dudas de teoría y/o prácticas, trabajos tutelados, etc. en horario de tutorías de cada profesor/a.

Evaluación			
Metodologías	Competencias	Descripción	Calificación
Prácticas de laboratorio	A28 B1 B3 C2 C3 C6	Entrega y defensa de trabajos prácticos de laboratorio. Computa hasta un máximo de 4 puntos de la nota final. Su realización no es obligatoria para superar la asignatura.	40
Trabajos tutelados	A28 B1 B3 C2 C3 C6	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



Prueba objetiva	B1 B3 C6	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
-----------------	----------	---	----

Observaciones evaluación

En caso de no alcanzar el mínimo en la prueba objetiva, la nota final será la obtenida en la prueba objetiva.

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

De acuerdo con la normativa de la UDC en relación al estudiantado matriculado a tempo parcial, el régimen de asistencia a clase no afectará negativamente al proceso de evaluación, admitiéndose en esta asignatura la dispensa académica para la asistencia, solicitada por las vías institucionales habilitadas al efecto. Sin embargo, esta flexibilidad asistencial no eximirá de la entrega de trabajos tutelados y prácticas de laboratorio en los mismos plazos fijados para el estudiantado a tiempo completo.

Fuentes de información

Básica	- Mordechai Ben-Ari (2012). Mathematical Logic for Computer Science. Springer - Ron Patton (2001). Software testing. Sams - Peter Farrell-Vinay (2008). Manage software testing. Auerbach - Kent Beck (2002). Test Driven Development (By Example). Addison-Wesley - Gerard J. Holzmann (2003). The SPIN model checker: primer and reference manual. Addison-Wesley - Mordechai Ben-Ari (2001). Mathematical Logic for Computer Science. Springer - Zohar Manna and Amir Pnueli (1991). The Temporal Logic of Reactive and Concurrent Systems. Specification. Springer - Zohar Manna and Amir Pnueli (1995). The Temporal Logic of Reactive and Concurrent Systems. Safety. Springer
Complementaria	

Recomendaciones

Asignaturas que se recomienda haber cursado previamente

Diseño Software/614G01015

Concurrencia y Paralelismo/614G01018

Proceso Software/614G01019

Arquitectura del Software/614G01221

Ingeniería de Requisitos/614G01222

Aseguramiento de la Calidad/614G01223

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

Proyectos de Desarrollo Software/614G01226

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