



Guía docente				
Datos Identificativos				2023/24
Asignatura (*)	Validación y Verificación del Software	Código	614G01053	
Titulación	Grao en Enxeñaría Informática			
Descritores				
Ciclo	Periodo	Curso	Tipo	Créditos
Grado	1º cuatrimestre	Cuarto	Optativa	6
Idioma	CastellanoGallego			
Modalidad docente	Presencial			
Prerrequisitos				
Departamento	Ciencias da Computación e Tecnoloxías da InformaciónComputación			
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	
	Muñiz Castro, Brais		brais.mcastro@udc.es	
	Perez Vega, Gilberto		gilberto.pvega@udc.es	
Web	campusvirtual.udc.es			
Descripción general	<p>Esta asignatura busca dominar las alternativas actuales de la ingeniería de software para la validación y verificación del software, mediante:</p> <ul style="list-style-type: none"> <li>- el conocimiento de técnicas y herramientas de validación de software funcionales y no funcionales a todos los niveles (unidad, integración, sistema);</li> <li>- el conocimiento de técnicas y herramientas de razonamiento automático; y</li> <li>- el conocimiento de técnicas y herramientas de verificación formal.</li> </ul>			

Competencias / Resultados del título	
Código	Competencias / Resultados 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 / Resultados 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	C2
		B3	C3
			C6
			C7
			C8

Contenidos	
Tema	Subtema



Bloque I: Validación de software	<p>I.1 Especificación, diseño y ejecución de pruebas</p> <p>I1.1. Niveles y tipos de pruebas</p> <p>I1.2. Propiedades y trazabilidad de requisitos</p> <p>I1.3. Automatización</p> <p>I.2 Gestión de pruebas: planificación, evaluación, métricas y revisiones</p>
Bloque II: Métodos formales y razonamiento automático	<p>II.1 Introducción: Deducción natural y cálculo de secuentes</p> <p>II.2 Prueba automática utilizando PVS</p> <p>II.3 ¿Qué es y para qué se utiliza un probador de teoremas?</p> <p>II.4 Lenguaje de especificación de PVS: Tipos, expresiones, teorías, subtipado</p> <p>II.5 Probador PVS: tácticas, recursión, razonamiento ecuacional</p>
Bloque III: Comprobación por modelos (model checking)	<p>III.1 Introducción a las lógicas modales temporales</p> <p>III.2 Especificación de propiedades: deadlocks, safety, liveness, fairness</p> <p>III.3 Funcionamiento de un comprobador por modelos</p> <p>III.4 Introducción al manejo de una herramienta de model checking</p>

Planificación				
Metodologías / pruebas	Competencias / Resultados	Horas lectivas (presenciales y virtuales)	Horas trabajo autónomo	Horas totales
Sesión magistral	B3 C2 C7 C8	21	26.25	47.25
Prácticas de laboratorio	A28 B1 B3 C2 C3 C6	21	42	63
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.
Prueba objetiva	Examen escrito.

Atención personalizada	
Metodologías	Descripción
Prueba objetiva Sesión magistral Prácticas de laboratorio	<p>Resolución de dudas de teoría y/o prácticas, trabajos tutelados, etc. en horario de tutorías de cada profesor/a.</p> <p>El seguimiento de la asignatura no debe presentar problemas al estudiantado con matrícula a tiempo parcial, ya que no se exige ni se puntúa la asistencia. Sin embargo, este alumnado es responsable de estar al corriente de los materiales colgados en el Moodle, así como de las tareas que se propongan para entrega. Estas entregas, de no ser telemáticas, serán acordadas con el estudiantado a tiempo parcial, de modo que se compatibilice en la medida de lo posible con su disponibilidad y la del profesorado.</p>

Evaluación			
Metodologías	Competencias / Resultados	Descripción	Calificación
Prueba objetiva	B1 B3 C6	Examen escrito que computa 4 puntos de la nota final. Se requiere un mínimo de 2 puntos para superar la asignatura.	40



Prácticas de laboratorio	A28 B1 B3 C2 C3 C6	Entrega y defensa de trabajos prácticos de laboratorio. Computan 6 puntos de la nota final, 2 puntos por cada bloque de contenidos. Su realización no es obligatoria para superar la asignatura.	60
--------------------------	--------------------	--	----

### Observaciones evaluación

En la convocatoria de primera oportunidad, 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 tendrá un peso del 80% en los casos en los que la evaluación continua y de prácticas de laboratorio sume un total inferior

a 1 punto, con el fin de garantizar el derecho de las/los estudiantes a superar la asignatura. De acuerdo con la normativa de la UDC en relación con el 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. Asimismo, según

se recoge en las distintas normativas de aplicación para la docencia universitaria, la incorporación de la perspectiva de género en esta materia

se aplicará empleando lenguaje no sexista y propiciando la intervención durante las sesiones magistrales de alumnas y

alumnos de modo equilibrado. Se trabajará

para identificar y modificar prejuicios y actitudes sexistas, y

se influirá en el entorno para modificarlos y fomentar valores de respeto e

igualdad. Finalmente, en caso de que se detecten situaciones de

discriminación por razón de género, se propondrán acciones y medidas para corregirlas.

### Fuentes de información

<b>Básica</b>	<ul style="list-style-type: none"><li>- Maurício Aniche (2022). Effective software testing. Manning</li><li>- Charity Majors, Liz Fong-Jones, George Miranda (2022). Observability Engineering. O'Reilly Media, Inc.</li><li>- Brian Okken (2022). Python testing with pytest. The Pragmatic Programmers</li><li>- Gayathri Mohan (2022). Full Stack Testing. O'Reilly Media, Inc.</li><li>- Fred Hébert (2019). Property-based testing with PropEr, Erlang, and Elixir : find bugs before your users do. The Pragmatic Bookshelf</li><li>- Mordechai Ben-Ari (2012). Mathematical Logic for Computer Science. Springer</li><li>- Crispin Lisa, Gregory Janet (2008). Agile Testing: A Practical Guide for Testers and Agile Teams. Addison-Wesley</li><li>- Gerard J. Holzmann (2003). The SPIN model checker: primer and reference manual. Addison-Wesley</li><li>- Kent Beck (2002). Test Driven Development (By Example). Addison-Wesley</li><li>- Mordechai Ben-Ari (2001). Mathematical Logic for Computer Science. Springer</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



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