



Guía Docente				
Datos Identificativos				2024/25
Asignatura (*)	Validación y Verificación del Software	Código	614G01053	
Titulación				
Descritores				
Ciclo	Período	Curso	Tipo	Créditos
Grao	1º cuatrimestre	Cuarto	Optativa	6
Idioma	CastelánGalego			
Modalidade docente	Presencial			
Prerrequisitos				
Departamento	Ciencias da Computación e Tecnoloxías da InformaciónComputación			
Coordinación	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	
	Paris Fernandez, Javier		javier.paris@udc.es	
	Perez Vega, Gilberto		gilberto.pvega@udc.es	
Web	campusvirtual.udc.es			
Descrición xeral	<p>Esta materia 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"> - coñecemento de técnicas e ferramentas de validación de software funcionais e non funcionais a todos os niveles (unidade, integración, sistema); - coñecemento de técnicas e ferramentas de razonamiento automático; e - coñecemento de técnicas e ferramentas de verificación formal. 			

Competencias / Resultados do título	
Código	Competencias / Resultados do título

Resultados da aprendizaxe			
Resultados de aprendizaxe			Competencias / Resultados do título
Capacidade de identificar e analizar problemas e deseñar, desenvolver, implementar, verificar e documentar solucións software sobre a base dun coñecemento axeitado das teorías, modelos e técnicas actuais.	A28	B1	C2
		B3	C3
			C6
			C7
			C8

Contidos	
Temas	Subtemas
Bloque I: Validación de software	I.1 Especificación, deseño e execución de probas I1.1. Niveis e tipos de probas I1.2. Propiedades e trazabilidade de requirimentos I1.3. Automatización I.2 Xestión de probas: planificación, avaliación, métricas e revisións
Bloque II: Métodos formais e razonamiento automático	II.1 Introducción: Dedución natural e cálculo de secuentes II.2 Proba automática utilizando PVS II.3 Que é e para que se utiliza un probador de teoremas? II.4 Linguaxe de especificación de PVS: Tipos, expresións, teorías, subtipado II.5 Probador PVS: tácticas, recursión, razoamento ecuacional



Bloque III: Comprobación por modelos (model checking)	III.1 Introducción ás lóxicas modais temporais III.2 Especificación de propiedades: deadlocks, safety, liveness, fairness III.3 Funcionamento dun comprobador por modelos III.4 Introducción ao manexo dunha ferramenta de model checking
---	--

Planificación				
Metodoloxías / probas	Competencias / Resultados	Horas lectivas (presenciais e virtuais)	Horas traballo autónomo	Horas totais
Sesión maxistral	B3 C2 C7 C8	21	26.25	47.25
Prácticas de laboratorio	A28 B1 B3 C2 C3 C6	21	42	63
Proba obxectiva	B1 B3 C6	3	31.5	34.5
Atención personalizada		5.25	0	5.25

*Os datos que aparecen na táboa de planificación son de carácter orientativo, considerando a heteroxeneidade do alumnado

Metodoloxías	
Metodoloxías	Descrición
Sesión maxistral	Clases de teoría onde se imparten os contidos do temario.
Prácticas de laboratorio	Elaboración de traballos prácticos no laboratorio.
Proba obxectiva	Exame escrito.

Atención personalizada	
Metodoloxías	Descrición
Proba obxectiva Sesión maxistral Prácticas de laboratorio	Resolución de dúbidas de teoría e/ou prácticas, traballos tutelados, etc. en horario de titorías de cada profesor/a. O seguimento da materia non debe presentar problemas ao estudantado con matrícula a tempo parcial, xa que non se esixe nin se puntúa a asistencia. Porén, este alumnado é responsable de estar ao corrente dos materiais colgados no Moodle, así como das tarefas que por se propoñan para entrega. Estas entregas, de non ser telemáticas, serán acordadas co estudantado a tempo parcial de xeito que se compatibilice na medida do posible coa súa dispoñibilidade e a do profesorado.

Avaliación			
Metodoloxías	Competencias / Resultados	Descrición	Cualificación
Proba obxectiva	B1 B3 C6	Exame escrito que computa 4 puntos da nota final. Require un mínimo de 2 puntos para superar a materia.	40
Prácticas de laboratorio	A28 B1 B3 C2 C3 C6	Entrega e defensa de traballos prácticos de laboratorio. Computa 6 puntos da nota final, 2 puntos por bloque de contidos da materia. A súa realización non é obrigatoria para superar a materia.	60

Observacións avaliación



Na convocatoria de primeira oportunidade, en caso de non acadar o mínimo na proba obxectiva, a nota final será a obtida na proba obxectiva.

Na convocatoria de segunda oportunidade, a proba obxectiva terá un peso do 80% nos casos nos que a avaliación

continúa e de prácticas de laboratorio suma un total inferior

a 1 punto, a fin de garantir o dereito das/os estudantes a superar a materia.

De acordo coa normativa da UDC en relación ao estudiantado matriculado a tempo parcial, o réxime de asistencia a clase non afectará negativamente ao proceso de avaliación, admitíndose nesta materia a dispensa académica para a asistencia solicitada polas canles institucionais habilitadas ao efecto. Porén, esta flexibilidade asistencial non eximirá da entrega de traballos tutelados e prácticas nos mesmos prazos fixados para o estudiantado a tempo completo.

Así mesmo, segundo

se recolle nas distintas normativas de aplicación para a docencia

universitaria, a incorporación da perspectiva de xénero nesta materia aplicarase empregando unha linguaxe non sexista e propiciando a intervención durante as sesións mixtas de alumnas e

alumnos de xeito equilibrado. Traballarase

para identificar e modificar prexuízos e actitudes sexistas, e

influirase na contorna para modificalos e fomentar valores de respecto e

igualdade. Finalmente, caso de se detectaren situacións de discriminación por razón de xénero, proporáanse accións e medidas para corrixilas.

Fontes de información

Bibliografía básica	<ul style="list-style-type: none"> - Maurício Aniche (2022). Effective software testing. Manning - Charity Majors, Liz Fong-Jones, George Miranda (2022). Observability Engineering. O'Reilly Media, Inc. - Brian Okken (2022). Python testing with pytest. The Pragmatic Programmers - Gayathri Mohan (2022). Full Stack Testing. O'Reilly Media, Inc. - Fred Hébert (2019). Property-based testing with PropEr, Erlang, and Elixir : find bugs before your users do. The Pragmatic Bookshelf - Mordechai Ben-Ari (2012). Mathematical Logic for Computer Science. Springer - Crispin Lisa, Gregory Janet (2008). Agile Testing: A Practical Guide for Testers and Agile Teams. Addison-Wesley - Gerard J. Holzmann (2003). The SPIN model checker: primer and reference manual. Addison-Wesley - Kent Beck (2002). Test Driven Development (By Example). Addison-Wesley - Mordechai Ben-Ari (2001). Mathematical Logic for Computer Science. Springer - Zohar Manna and Amir Pnueli (1995). The Temporal Logic of Reactive and Concurrent Systems. Safety. Springer - Zohar Manna and Amir Pnueli (1991). The Temporal Logic of Reactive and Concurrent Systems. Specification. Springer
Bibliografía complementaria	

Recomendacións

Materias que se recomenda ter cursado previamente

Deseño Software/614G01015
 Concorrencia e Paralelismo/614G01018
 Proceso Software/614G01019
 Arquitectura do Software/614G01221
 Enxeñaría de Requisitos/614G01222
 Aseguramento da Calidade/614G01223

Materias que se recomenda cursar simultaneamente

Representación do Coñecemento e Razoamento Automático/614G01036
 Teoría da computación/614G01039
 Metodoloxías de Desenvolvemento/614G01051



Materias que continúan o temario
Proxectos de Desenvolvemento Software/614G01226
Observacións

(*A Guía docente é o documento onde se visualiza a proposta académica da UDC. Este documento é público e non se pode modificar, salvo casos excepcionais baixo a revisión do órgano competente dacordo coa normativa vixente que establece o proceso de elaboración de guías