



Guía docente				
Datos Identificativos				2019/20
Asignatura (*)	Metodologías de Desarrollo	Código	614G01051	
Titulación	Grao en Enxeñaría Informática			
Descritores				
Ciclo	Periodo	Curso	Tipo	Créditos
Grado	2º cuatrimestre	Cuarto	Optativa	6
Idioma	Castellano			
Modalidad docente	Presencial			
Prerrequisitos				
Departamento	Ciencias da Computación e Tecnoloxías da InformaciónComputación			
Coordinador/a	Rodríguez Rubio, Miguel Jose	Correo electrónico	miguel.rodriguez.rubio@udc.es	
Profesorado	Martín Rodilla, Patricia	Correo electrónico	patricia.martin.rodilla@udc.es	
	Rodríguez Rubio, Miguel Jose		miguel.rodriguez.rubio@udc.es	
Web				
Descripción general	<p>Metodologías existentes para distintos ciclos de vida (ágil, clásica, cascada, espiral,...).</p> <p>Metodologías clásicas. Proceso unificado de desarrollo. Metodologías ágiles de desarrollo.</p> <p>Programación extrema. Desarrollo colaborativo. Evolución y mantenimiento del software (integración, sistemas legacy,...). Metodología y aspectos éticos/sociales/legales característicos en distintos dominios de aplicación.</p>			

Competencias / Resultados del título	
Código	Competencias / Resultados del título
A25	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.
A26	Capacidad para valorar las necesidades del cliente y especificar los requisitos software para satisfacer estas necesidades, reconciliando objetivos en conflicto mediante la búsqueda de compromisos aceptables dentro de las limitaciones derivadas del coste, del tiempo, de la existencia de sistemas ya desarrollados y de las propias organizaciones.
A27	Capacidad de dar solución a problemas de integración en función de las estrategias, estándares y tecnologías disponibles.
A30	Capacidad para diseñar soluciones apropiadas en uno o más dominios de aplicación utilizando métodos de la ingeniería del software que integren aspectos éticos, sociales, legales y económicos.
A54	Capacidad para emplear metodologías centradas en el usuario y la organización para el desarrollo, evaluación y gestión de aplicaciones y sistemas basados en tecnologías de la información que aseguren la accesibilidad, ergonomía y usabilidad de los sistemas.
B1	Capacidad de resolución de problemas
B2	Trabajo en equipo
B3	Capacidad de análisis y síntesis
B4	Capacidad para organizar y planificar
B5	Habilidades de gestión de la información
B6	Toma de decisiones
B7	Preocupación por la calidad
B8	Capacidad de trabajar en un equipo interdisciplinar
B9	Capacidad para generar nuevas ideas (creatividad)
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.

Resultados de aprendizaje



Resultados de aprendizaje	Competencias / Resultados del título		
	A25	B1	C3
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 accesibles de desarrollar y mantener, y cumplan normas de calidad, aplicando las teorías, principios, métodos y prácticas de la ingeniería del software. Competencias SE1, SI3 y SE6 en la memoria del Grado.	A26	B2	C6
	A27	B3	C7
	A30	B4	
	A54	B5	
		B6	
		B7	
		B8	
		B9	

Contenidos	
Tema	Subtema
PARTE A. Metodologías y ciclos de vida	Tema 1. Evolución de las metodologías y sus enfoques Tema 2. Metodologías clásicas Tema 3. Enfoque de las Metodologías Actuales
PARTE B. Metodologías para Desarrollo Incremental	Tema 4. Estructura y alcance del Proceso Unificado de Desarrollo de Software Tema 5. Proceso: dirigido por casos de uso; centrado en las arquitecturas; iterativo e incremental Tema 6. Flujos de trabajo: captura de requisitos, análisis, diseño, implementación, validación y prueba
PARTE C. Metodologías ágiles de desarrollo	Tema 7. Introducción a las metodologías ágiles Tema 8. SCRUM: roles, equipos y auto-organización; planificación de reuniones; sprints; gestión y implementación. Tema 9. Programación extrema: principios de XP; conceptualización; planificación; desarrollo; entrega.
PARTE D. Complementos	Tema 10. Desarrollo colaborativo Tema 11. Evolución e mantenimiento del software Tema 12. El impacto del legacy Tema 13. Otros aspectos del desarrollo. Deontología y enfoque legal.

Planificación				
Metodologías / pruebas	Competencias / Resultados	Horas lectivas (presenciales y virtuales)	Horas trabajo autónomo	Horas totales
Sesión magistral	A25 A27 A30 B1 B3 B5 C3 C6 C7	21	52.5	73.5
Prácticas de laboratorio	A26 A54 B2 B4 B8	14	14	28
Solución de problemas	B7 B9	7	14.5	21.5
Prueba objetiva	B6	3	18	21
Atención personalizada		6	0	6

(*) 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 de ordenadores
Solución de problemas	Resolución de trabajos tutelados planteados e resueltos en horario de tutorías de grupos reducidos
Prueba objetiva	Examen escrito

Atención personalizada

Metodologías	Descripción
Sesión magistral Prácticas de laboratorio Solución de problemas Prueba objetiva	Atención personalizada. Solución de dudas de teoría o prácticas, ejercicios, etc.

Evaluación

Metodologías	Competencias / Resultados	Descripción	Calificación
Prácticas de laboratorio	A26 A54 B2 B4 B8	Entrega de trabajos prácticos de laboratorio en plazos establecidos a lo largo del curso. Computa un máximo de 4 puntos sobre el total de 10 de la nota final. La entrega no es obligatoria.	40
Solución de problemas	B7 B9	Planteamiento de pequeños problemas e resolución en grupos de tutorías reducidos. Computa un máximo de 2 punto sobre el total de 10 de la nota final. La entrega no es obligatoria. El día del examen se podrá hacer y entregar un ejercicio para subir la nota del TGR.	20
Prueba objetiva	B6	Realización de un examen escrito sobre los contenidos del temario de la materia. Computa un máximo de 4 puntos sobre el total de la nota final. Para aprobar la asignatura es preciso obtener un mínimo de 2 puntos en la prueba escrita.	40

Observaciones evaluación

--

Fuentes de información

Básica	<ul style="list-style-type: none"> - Ivar Jacobson, Grady Booch and James Rumbaugh (). El Proceso Unificado de Desarrollo Software. Addison-Wesley - James Rumbaugh, Ivar Jacobson and Grady Booch (). El Lenguaje Unificado de Modelado. Manual de Referencia. Addison-Wesley - Craig Larman (). Agile & Iterative Development. Addison-Wesley - Robert C. Martin (). Agile Software Development. Prentice Hall - Ken Schwaber and Mike Beedle (). Agile Software Development with Scrum. Prentice Hall - David Astels, Geanville Miller and Miroslav Novak (). A Practical Guide to Extreme Programming. Prentice Hall
Complementaria	

Recomendaciones

Asignaturas que se recomienda haber cursado previamente

Aseguramiento de la Calidad/614G01028

Asignaturas que se recomienda cursar simultáneamente

Marcos de Desarrollo/614G01052

Validación y Verificación del Software/614G01053



Asignaturas que continúan el temario

Proyectos de Desarrollo Software/614G01087
--

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