



DATOS IDENTIFICATIVOS

Desarrollo ágil de aplicaciones

Asignatura	Desarrollo ágil de aplicaciones			
Código	O06G150V01944			
Titulación	Grado en Ingeniería Informática			
Descriptores	Creditos ECTS	Seleccione	Curso	Cuatrimestre
	6	OP	4	2c
Lengua Impartición	Castellano Gallego			
Departamento	Informática			
Coordinador/a	Reboiro Jato, Miguel			
Profesorado	Reboiro Jato, Miguel			
Correo-e	mrjato@uvigo.es			
Web	http://fatic.uvigo.es			
Descripción general	En esta asignatura el alumno conocerá las bases del desarrollo ágil de software, así como las distintas metodologías enmarcadas en este campo de desarrollo de software. Además, el alumno deberá aplicar las dichas metodologías en el desarrollo de un producto software.			

Competencias

Código	
A2	Que los estudiantes sepan aplicar sus conocimientos a su trabajo o vocación de una forma profesional y posean las competencias que suelen demostrarse por medio de la elaboración y defensa de argumentos y la resolución de problemas dentro de su área de estudio.
A3	Que los estudiantes tengan la capacidad de reunir e interpretar datos relevantes (normalmente dentro de su área de estudio) para emitir juicios que incluyan una reflexión sobre temas relevantes de índole social, científica o ética.
A4	Que los estudiantes puedan transmitir información, ideas, problemas y soluciones a un público tanto especializado como no especializado.
A5	Que los estudiantes hayan desarrollado aquellas habilidades de aprendizaje necesarias para emprender estudios posteriores con un alto grado de autonomía.
B1	Capacidad para concebir, redactar, organizar, planificar, desarrollar y firmar proyectos en el ámbito de la ingeniería en informática que tengan por objeto, de acuerdo con los conocimientos adquiridos, la concepción, el desarrollo o la explotación de sistemas, servicios y aplicaciones informáticas.
B2	Capacidad para dirigir las actividades objeto de los proyectos del ámbito de la informática de acuerdo con los conocimientos adquiridos.
B5	Capacidad para concebir, desarrollar y mantener sistemas, servicios y aplicaciones informáticas empleando los métodos de la ingeniería del software como instrumento para el aseguramiento de su calidad, de acuerdo con los conocimientos adquiridos.
B9	Capacidad para resolver problemas con iniciativa, toma de decisiones, autonomía y creatividad. Capacidad para saber comunicar y transmitir los conocimientos, habilidades y destrezas de la profesión de Ingeniero Técnico en Informática.
B12	Conocimiento y aplicación de elementos básicos de economía y de gestión de recursos humanos, organización y planificación de proyectos, así como la legislación, regulación y normalización en el ámbito de los proyectos informáticos, de acuerdo con los conocimientos adquiridos.
C7	Capacidad para diseñar, desarrollar, seleccionar y evaluar aplicaciones y sistemas informáticos, asegurando su fiabilidad, seguridad y calidad, conforme a principios éticos y a la legislación y normativa vigente
C8	Capacidad para planificar, concebir, desplegar y dirigir proyectos, servicios y sistemas informáticos en todos los ámbitos, liderando su puesta en marcha y su mejora continua y valorando su impacto económico y social
C9	Capacidad para comprender la importancia de la negociación, los hábitos de trabajo efectivos, el liderazgo y las habilidades de comunicación en todos los entornos de desarrollo de software
C12	Conocimiento y aplicación de los procedimientos algorítmicos básicos de las tecnologías informáticas para diseñar soluciones a problemas, analizando la idoneidad y complejidad de los algoritmos propuestos
C22	Conocimiento y aplicación de los principios, metodologías y ciclos de vida de la ingeniería de software

C25	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
C26	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
C28	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
C29	Capacidad de identificar, evaluar y gestionar los riesgos potenciales asociados que pudieran presentarse
D1	I1: Capacidad de análisis, síntesis y evaluación
D2	I2: Capacidad de organización y planificación
D3	I3: Comunicación oral y escrita en la lengua nativa
D5	I5: Capacidad de abstracción: capacidad de crear y utilizar modelos que reflejen situaciones reales
D6	I6: Capacidad de diseñar y realizar experimentos sencillos y analizar e interpretar sus resultados
D7	I7: Capacidad de buscar, relacionar y estructurar información proveniente de diversas fuentes y de integrar ideas y conocimientos
D8	I8: Resolución de problemas
D9	I9: Capacidad de tomar decisiones
D10	I10: Capacidad para argumentar y justificar lógicamente las decisiones tomadas y las opiniones
D11	P1: Capacidad de actuar autónomamente
D13	P3: Capacidad de integrarse rápidamente y trabajar eficientemente en equipos unidisciplinarios y de colaborar en un entorno multidisciplinar
D15	P5: Capacidad de relación interpersonal
D16	S1: Razonamiento crítico
D18	S3: Aprendizaje autónomo
D20	S5: Creatividad
D21	S6: Liderazgo
D22	S7: Tener iniciativa y ser resolutivo
D24	S9: Tener motivación por la calidad y la mejora continua

Resultados de aprendizaje

Resultados previstos en la materia	Resultados de Formación y Aprendizaje			
RA1: Conocer las bases del desarrollo ágil de software	A2	B1	C8	D1
	A3	B2	C9	D2
	A4		C22	D3
	A5			D5
				D6
				D8
				D9
				D10
				D11
				D13
				D15
				D16
				D18
				D20
				D21
				D22
				D24
RA2: Conocer las distintas metodologías ágiles existentes para el desarrollo de software	A3	B5	C8	D1
	A4		C9	D3
	A5		C22	D7
			C25	D8
				D10
				D11
				D15
				D16
				D18
				D22

RA3: Saber aplicar las principales metodologías de desarrollo ágil de software

A2	B1	C7	D1
A3	B2	C8	D2
A4	B5	C9	D3
A5	B9	C12	D5
	B12	C22	D6
		C25	D8
		C26	D9
		C28	D10
		C29	D11
			D13
			D15
			D16
			D20
			D21
			D22
			D24

RA4: Conocer y saber aplicar métodos ágiles de gestión de proyectos

A2	B1	C7	D1
A3	B2	C8	D2
A4	B5	C9	D3
A5	B9	C12	D5
	B12	C22	D6
		C25	D7
		C26	D8
		C28	D9
		C29	D10
			D11
			D13
			D15
			D16
			D18
			D20
			D21
			D22
			D24

RA5: Identificar y saber valorar las ventajas y desventajas de las metodologías ágiles frente a otras metodologías de desarrollo de software

A3	B5	C8	D1
A4	B12	C22	D3
A5		C25	D7
		C28	D8
			D10
			D11
			D15
			D16
			D22

RA6: Emplear las pruebas de software como un parte clave del desarrollo de software

A2	B1	C7	D1
A3	B5	C8	D2
A4	B9	C12	D5
A5		C22	D6
		C25	D7
		C26	D8
		C28	D9
		C29	D10
			D11
			D13
			D15
			D16
			D20
			D24

Contenidos

Tema

Bases del Desarrollo Ágil de Aplicaciones (DAA)	Introducción a las bases del DAA y presentación de las principales metodologías.
Buenas Prácticas en el DAA	Estudio de las prácticas básicas y más importantes en el DAA.
Gestión de Proyectos en DAA	Introducción y aprendizaje de las principales metodologías ágiles para la gestión de proyectos, como, por ejemplo, Scrum.
Metodologías de DAA	Introducción y aprendizaje de las principales metodologías ágiles para el desarrollo de aplicaciones, como, por ejemplo, la programación extrema.

Planificación			
	Horas en clase	Horas fuera de clase	Horas totales
Lección magistral	16.5	23.5	40
Prácticas de laboratorio	10	15	25
Resolución de problemas y/o ejercicios	3	12	15
Presentación	3	7	10
Proyecto	20	40	60

*Los datos que aparecen en la tabla de planificación son de carácter orientativo, considerando la heterogeneidad de alumnado

Metodologías	
	Descripción
Lección magistral	Exposición de los contenidos teóricos de la materia. Con el fin de facilitar la comprensión de la materia y aumentar el interés del alumno, se incluirán diversos ejemplos en los que se puede requerir la participación activa del alumno.
Prácticas de laboratorio	Estudio teórico y práctico de las tecnologías necesarias para el desarrollo del proyecto y de una propuesta de arquitectura para lo mismo.

Atención personalizada	
Metodologías	Descripción
Prácticas de laboratorio	Atención a las preguntas y dudas de los alumnos que puedan surgir a lo largo del trabajo a realizar en las clases.
Pruebas	Descripción
Resolución de problemas y/o ejercicios	Seguimiento semanal del trabajo en el proyecto grupal y resolución de las dudas de carácter técnico que puedan surgir relacionadas con él.
Proyecto	Seguimiento semanal del trabajo en el proyecto grupal y resolución de las dudas de carácter técnico que puedan surgir relacionadas con él.

Evaluación				
	Descripción	Calificación	Resultados de Formación y Aprendizaje	
Resolución de problemas y/o ejercicios	Realización de un proyecto grupal en el que se aplicarán de forma práctica metodologías ágiles simulando un entorno real. Cada grupo de trabajo deberá hacer varias entregas en las que cada alumno será evaluado por el trabajo individual, el trabajo grupal y por una prueba escrita en la que se evaluarán sus conocimientos del proyecto y de la metodología empleada. RESULTADOS DE APRENDIZAJE: RA1, RA3, RA4 y RA6.	50	A2	B1 C7 D1
			A3	B2 C8 D2
			A4	B5 C9 D3
			A5	B9 C12 D5
				B12 C22 D6
				C25 D7
				C26 D8
				C28 D9
				C29 D10
				D11
				D13
				D15
				D16
				D20
Presentación	Preparación y presentación en pequeños grupos de un tema relacionado con la materia. En la evaluación del trabajo se tendrá en cuenta el contenido del trabajo, la presentación oral del mismo y la contextualización dentro de la materia. Además, se evaluará la capacidad del alumnado para calificar los trabajos presentados por los compañeros. RESULTADOS DE APRENDIZAJE: RA2, RA4 y RA5.	10	A3	B9 D1
			A4	D3
			A5	D5
				D7
				D9
				D10
				D11
				D15
				D16
				D18
				D22

Proyecto	Realización de un proyecto grupal en el que se aplicarán de forma práctica metodologías ágiles simulando un entorno real.	40	A2	B1	C7	D1
			A3	B2	C8	D2
Cada grupo de trabajo deberá hacer varias entregas en las que cada alumno será evaluado por el trabajo individual, el trabajo grupal y por una prueba escrita en la que se evaluarán sus conocimientos del proyecto y de la metodología empleada.			A4	B5	C9	D3
			A5	B9	C12	D5
				B12	C22	D6
					C25	D7
					C26	D8
					C28	D9
					C29	D10
						D11
						D13
						D15
RESULTADOS DE APRENDIZAJE: RA1, RA3, RA4 y RA6.						D16
						D20
						D21
						D22
						D24

Otros comentarios sobre la Evaluación

MÉTODO DE EVALUACIÓN

En el inicio de la asignatura, los alumnos deberán cubrir un formulario en el que indiquen el método por el que quieren ser evaluados (asistentes o no asistentes). Aquellos alumnos que no entreguen este formulario serán evaluados por el método de no asistentes.

CRITERIOS DE EVALUACIÓN PARA LA 1ª EDICIÓN DE LAS ACTAS

[Asistentes]

Calificación final = 0.1 * nota de "Presentación" + 0.4 * nota de los "Proyecto" + 0.5 * nota de la "Resolución de problemas y/o ejercicios"

[No asistentes]

En el caso de los no asistentes el trabajo de "Presentación" se realizará de forma individual y solo se presentará ante el profesorado de la materia. Los resultados de aprendizaje y competencias de esta actividad son los mismos que los del trabajo de "Presentación".

Calificación final = 0.1 * nota del "Trabajo teórico" + 0.4 * nota de los "Proyecto" + 0.5 * nota de la "Resolución de problemas y/o ejercicios"

CRITERIOS DE EVALUACIÓN PARA LA 2ª EDICIÓN DE LAS ACTAS Y FIN DE CARRERA

En el caso de las ediciones de las actas segunda y fin de carrera, el trabajo de "Presentaciones/exposiciones" se realizará de forma individual y solo se presentará ante el profesorado de la materia. La evaluación será a misma para asistentes y no asistentes. Los resultados de aprendizaje y competencias de esta actividad son los mismos que los del trabajo de "Presentación".

Respeto al trabajo de "Proyecto", en caso de que el alumno trabajara en un proyecto durante lo curso, deberá continuar con el dicho proyecto. En caso contrario, se podrá incorporar un proyecto existente o iniciar un nuevo proyecto, según el criterio del profesorado.

Calificación final = 0.1 * nota de "Trabajo teórico" + 0.4 * nota de los "Proyecto" + 0.5 * nota de la "Resolución de problemas y/o ejercicios"

PROCESO DE CALIFICACIÓN DE LAS ACTAS

En cualquiera de las convocatorias, el alumno deberá superar cada una de las metodologías de evaluación y de las pruebas parciales de las que se componen para superar la materia. Se considerará que una metodología de evaluación está superada cuando se obtenga una puntuación igual o superior al 50% de la nota máxima de la dicha metodología. Además, se considerará que una prueba está superada cuando se obtenga una puntuación igual o superior al 40% de la nota máxima de la dicha prueba. En caso de que un alumno no supere alguna de las metodologías y/o pruebas, se asignará un máximo de 4,9 puntos como nota final de la materia.

FECHAS DE EVALUACIÓN

El calendario de pruebas de evaluación aprobado oficialmente por la Xunta de Centro de la ESEI se encuentra publicado en la página web <http://www.esei.uvigo.es>.

Fuentes de información

Bibliografía Básica

Peter Tahchiev, Felipe Leme, Vincent Massol y Gary Gregory, **JUnit In Action**, 2ª, Manning, 2011
Mike Cohn, **User Stories Applied: for Agile Software Development**, 1ª, Addison-Wesley, 2004
Kent Beck, **Test-Driven Development: by example**, 1ª, Addison-Wesley, 2003
Kent Beck, **Extreme Programming: Explained**, 2ª, Addison-Wesley, 2005
Chris Sims y Hillary Louise Johnson, **The Elements of Scrum**, 1ª, Dymaxicon, 2011
Mike Cohn, **Succeeding with Agile: Software Development Using Scrum**, 1ª, Addison-Wesley, 2010
David J. Anderson, **Kanban: successful evolutionary change for your technology business**, 1ª, Blue Hole Press, 2010
Mike Cohn, **Agile Estimating and Planning**, 1ª, Prentice Hall Professional Technical Reference, 2012
Glenford J. Myers, Tom Badgett, Corey Sandler, **The Art of Software Testing**, 3ª, John Wiley & Sons, 2012

Bibliografía Complementaria

Steve Freeman y Nat Pryce, **Growing Object-Oriented Software, Guided By Tests**, 1ª, Addison Wesley, 2010
Paul M. Duvall, **Continuous integration: improving software quality and reducing risk / Paul M. Duvall with Steve Matyas and Andrew Glover**, 1ª, Addison-Wesley, 2007
Martin Fowler, **Refactoring: Improving the Design of Existing Code**, 1ª, Addison-Wesley, 2000
Henrik Kniberg, **Scrum y XP desde las Trincheras**, 1ª, InfoQ, 2007
Carlos Blé Jurado, Juan Gutiérrez Plaza, Fran Reyes Perdomo y Gregorio Mena, **Diseño Ágil con TDD**, 1ª, Lulu, 2010
Kenneth S. Rubin, **Essential Scrum: A Practical Guide to the Most Popular Agile Process**, 1ª, Addison-Wesley, 2013
George Meszaros, **xUnit Test Patterns: Refactoring Test Code**, 5ª, Addison-Wesley, 2012

Recomendaciones

Asignaturas que se recomienda haber cursado previamente

Algoritmos y estructuras de datos II/O06G150V01302
Ingeniería del software I/O06G150V01304
Ingeniería del software II/O06G150V01403
Diseño de arquitecturas de grandes sistemas software/O06G150V01947
