Guia docente				
Datos Identificativos				2013/14
Asignatura (*)	Metodologías de Desarrollo		Código	614G01224
Titulación	Grao en Enxeñaría Informática			
Descriptores				
Ciclo	Periodo	Curso	Tipo	Créditos
Grado	1º cuatrimestre	Curso adap. Ing.. Téc. Informática	Obligatoria	6
Idioma	Gallego			
Prerrequisitos				
Departamento	Computación			
Coordinador/a	Casanova Crespo, Jose Maria	Correo electrónico	jose.casanova.crespo@udc.es	
Profesorado	Casanova Crespo, Jose Maria	Correo electrónico	jose.casanova.crespo@udc.es	
Web	campusvirtual.udc.es			
Descripción general	Metodoloxías Clásicas O proceso unificado de desenvolvemento Metodoloxías axiles de desenvolvemento Programación Extrema (XP) Desenvolvemento colaborativo Evolución e mantemento do software Aspectos sociais, legais e éticos no desenvolvemento software			

Competencias de la titulación	
Código	Competencias de la titulación
A6	Conocimiento adecuado del concepto de empresa, marco institucional y jurídico de la empresa. Organización y gestión de empresas.
A7	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.
A8	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.
A9	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.
A22	Conocimiento y aplicación de los principios, metodologías y ciclos de vida de la ingeniería de software.
A24	Conocimiento de la normativa y la regulación de la informática en los ámbitos nacional, europeo e internacional.
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.
A27	Capacidad de dar solución a problemas de integración en función de las estrategias, estándares y tecnologías disponibles.
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.
A29	Capacidad de identificar, evaluar y gestionar los riesgos potenciales asociados que pudieran presentarse.
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.
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
B8	Capacidad de trabajar en un equipo interdisciplinar

C1	Expresarse correctamente, tanto de forma oral como escrita, en las lenguas oficiales de la comunidad autónoma.
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.
C4	Desarrollarse para el ejercicio de una ciudadanía abierta, culta, crítica, comprometida, democrática y solidaria, capaz de analizar la realidad, diagnosticar problemas, formular e implantar soluciones basadas en el conocimiento y orientadas al bien común.
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			
Competencias de materia (Resultados de aprendizaje)	Competencias de la titulación		
Conocer los diferentes tipos de metodologías de desarrollo de software y sus fundamentos.	A8 A22 A25 A29	B3 B8	C1 C4 C6 C7
Ser capaz de seleccionar la metodología de desarrollo más adecuada y adaptarla a las necesidades del proyecto software y a la organización de lo desarrolla.	A6 A8 A9 A22 A24 A25 A28	B1 B4 B6	C1 C3 C4 C6 C8
Utilizar herramientas metodológicas para el desarrollo en entornos colaborativos.	A7 A22 A30	B1 B2 B5 B8	C1 C2 C3 C6 C8
Conocer metodologías y técnicas para la reutilización, evolución y mantenimiento de proyectos.	A22 A27 A29 A30	B1 B2 B3 B8	C1 C3 C6 C7
Adaptar las metodologías a los requsitos éticos, sociales y legales.	A6 A9 A22 A27	B6 B8	C1 C2 C4 C7

Contenidos	
Tema	Subtema
1. Introducción	* Metodología vs método. * Metodologías de desarrollo de software. * El ciclo de vida del software.
2. Metodologías Clásicas	* Cascada * Prototipado * Espiral * Incremental * Desarrollo rápido de aplicaciones

3. El proceso unificado de desarrollo	* Características * El Lenguaje Unificado de Modelado (UML) * Casos de uso * La importancia da arquitectura * Ciclo de vida
4. Metodologías agiles de desarrollo	* El manifiesto Agile * Características de las metodologías ágiles. * La metodología SCRUM * Otras metodologías ágiles
5. Programación Extrema (XP)	* Valores y Principios * Técnicas * Equipo de desarrollo * Aplicaciones prácticas
6. Desarrollo colaborativo	* Desarrollo en equipo * Herramientas de trabajo colaborativo. * Desarrollo en aberto. * Gestión de una comunidad de software libre. * Estándares abiertos. * Colaborando con proveedores.
7. Evolución y mantenimiento del software	* Legacy como software de éxito. * Mantenimiento y evolución del software. * Control de versiones. * Gestión de errores.
8. Aspectos sociales, legales y éticos en el desarrollo software.	* Licenciamiento y propiedad intelectual en el Software. * Software libre y Software privativo. * Interoperabilidad y uso de estándares. * Seguridad y confianza.

Planificación			
Metodologías / pruebas	Horas presenciales	Horas no presenciales / trabajo autónomo	Horas totales
Sesión magistral	21	42	63
Prácticas de laboratorio	14	28	42
Seminario	5	10	15
Trabajos tutelados	2	16	18
Prueba objetiva	3	6	9
Atención personalizada	3	0	3
(*)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 expositivas de presentación de los conocimientos teóricosutilizando diferentes recursos: pizarra, proyección de material de formato electrónico, apuntes en formato electrónico y los recursos facilitados por el equipo docente de la asignatura en la facultad virtual.
Prácticas de laboratorio	Prácticas diseñadas por el equipo docente de la asignatura basadas en los conocimientos que cada estudiante va adquiriendo. Estos trabajos serán desarrollados en grupo. Se trabajará en el uso de herramientas de apoyo a la implantación de una metodología en un proyecto de desarrollo de Software.

Seminario	Tutorías de grupos reducidos (TGRs) en los que se propondrán actividades relacionadas con los conocimientos adquiridos en teoría o en práctica.
Trabajos tutelados	Los alumnos deben realizar un informe que analice y evalue de forma crítica una herramienta de apoyo para la implementación de una metodología. Los trabajos se publicarán en la facultad virtual y serán compartidos entre los alumnos para ayudarles en la selección de herramientas software. Se potenciará la realización de trabajos en inglés.
Prueba objetiva	Prueba escrita medianta la que se valoran los conocimientos adquiridos por el estudiantado. Cada estudiante deberá aplicar tanto sus conocimientos a nivel teórico como a nivel práctico.

Atención personalizada	
Metodologías	**Descripción**
Prácticas de laboratorio Seminario Trabajos tutelados	* La atención personalizada consistirá en la realización de tutorías presenciales o a través del campus virtual para resolver dudas sobre los contenidos de la asignatura. * Se realizará el seguimiento personalizado de las prácticas de laboratorio. * De la misma forma, se revisará personalmente con los alumnos los resultados de los trabajos de los seminarios, así como la realización del trabajo tutelado.

Evaluación		
Metodologías	**Descripción**	**Calificación**
Prueba objetiva	Prueba escrita realizada al final del curso sobre contenidos teórico-prácticos. La prueba objetiva es obligatoria para superar la asignatura. Es obligatorio obtener una nota mínima de 3,5 sobre 10 para poder hacer media con otros elementos evaluables. En caso de no llegar a la nota mínima implicará que no se puede obtener más de un 4,5 en la nota final de la asignatura.	40
Prácticas de laboratorio	Evaluación continua de las prácticas propuestas a lo largo del curso. Entre los aspectos a considerar a la hora de valorar las prácticas, se encuentran: - Rigor en la consecución de los objetivos perseguidos en la práctica utilizando las técnicas propuestas en la asignatura. - Asimilación de los conceptos perseguidos por la práctica. - Originalidad en las propuestas acometidas durante la realización de la práctica. - Responsabilidad en la entrega de las prácticas en tiempo y forma, así como el uso adecuado de los recursos habilitados para este fin. - Valoración del trabajo en grupo	40
Trabajos tutelados	Los alumnos deberan realizar un informe que analice y evalue de forma crítica una herramienta de apyo para la implementación de una metodología. Los trabajos se publicarán en la facultad virtual y serán compartidos entre los alumnos para ayudarles a la selección de herramientas software. Se valorará la calidad del trabajo, la expresión, el rigor académico. También se valorará adicionalmente el uso del inglés.	20

Observaciones evaluación

Aspectos a tener en cuenta en la evaluación de la segunda oportunidad de Julio: * La nota de prácticas de laboratorio y del trabajo tutelado se mantiene idéntica a la de la primera oportunidad, existirá la posibilidad de volver a entregar el trabajo tutelado. En ese caso se deberá indicar las diferencias y mejoras con la entrega de la primera oportunidad. * La nota de la prueba objectiva sólo se mantiene en el caso de que sea igual o superior a cinco en la primera oportunidad. En el caso de no superar el cinco en la primera oportunidad será obligatorio repetir la prueba objetiva.Se considerara que un alumno se presenta a la segunda oportunidad si vuelve a entregar el trabajo tutelado o se presenta a la prueba objetiva.

Fuentes de información	
Básica	- Larman, Craig (2004). Agile & Iterative Development. Addison Wesley - Fogel, Karl (2010). Creando Software Libre. Edizer, GHANDALF - Jacobson, Ivar ; Booch, Grady; Rumbaugh, James (2000). El proceso unificado de desarrollo de software. Addison Wesley - Beck, Kent ; Andres, Cynthia (2005). Extreme Programming Explained (2nd ed.). Addison Wesley - Pilone, Dan; Miles, Russ (2008). Head First Software Development. O'Reilly - Fowler, Martin (2000). Refactoring: Improving the Design of Existing Code. Addison Wesley
Complementária	- Martin, Robert C (2008). Clean Code: A Handbook of Agile Software Craftsmanship. Prentice Hall - Fox, Armando; Patterson, David (2012). Engineering Long-Lasting Software. Strawberry Canyon LLC - Stallman, Richard M (2010). Free Software, Free Society: Selected Essays of Richard M. Stallman. Free Software Foundation - Raymond, Eric S. (2000). The Cathedral and the Bazaar. O'Reilly

Recomendaciones
Asignaturas que se recomienda haber cursado previamente
Proyectos de Desarrollo Software/614G01226
Asignaturas que se recomienda cursar simultáneamente
Ingeniería de Requisitos/614G01222 Aseguramiento de la Calidad/614G01223
Asignaturas que continúan el temario
Otros comentarios