Guia docente				
Datos Identificativos				2019/20
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	CastellanoGallego			
Modalidad docente	Presencial			
Prerrequisitos				
Departamento	Computación			
Coordinador/a		Correo electrónico		
Profesorado		Correo electrónico		
Web	guiadocente.udc.es/guia_docent/index.php?centre=614&ensenyament=614G01&assignatura=614G01051&any_academic=2017_18&am			
Descripción general	Los contenidos de la materia de Metodologías de Desarrollo se centran el los aspectos metodológicos de la Ingeniería del Software. La asignatura trata de acercar a los estudiantes las diferentes aproximaciones para afrontar la problemática de como organizar un equipo que tiene como objetivo desarrolar un proyecto Software. Las herramientas metodológicas nos acercan prácticas y estrategias que nos ayudarán a definir la forma de organizar el trabajo para conseguir productos software que funcionen y cumplan las necesidades de los usuarios. La asignatura se centra en las nuevas metodologías de desarrollo de Software basadas en el paradigma Agile, pero realiza un breve recorrido por las metodologías clásicas como el desarrollo en Cascada o en Espiral. Dentro de las metodologías Agile tendrán especial relevancia Scrum y eXtreme Programming (XP), sin olvidar la importancia del proceso unificado de desarrollo. Se analizarán los paradigmas de desarrollo colaborativo que se basan en las prácticas del Open Source y se abordarán las implicaciones de la evolución y mantenimiento de proyectos software. La asignatura tentrá en cuenta los diferentes aspectos sociales, legales y éticos en el desarrollo software, y tocará temas como el Software Libre y el código ético de la Ingeniería del Software de la ACM. Durante el curso se realizarán diferentes lecturas de artículos clásicos relacionados con los contenidos de la asignatura e se desarrollará un proyecto colaboarativo de desarrollo de Software aplicando las diferentes herramientas metodológicas descritas en el curso. La guía docente de esta asignatura está diseñada y adaptada para el Curso de Adaptación o Grado de las Ingenierías Técnicas de Informática.			

Competencias del título	
Código	Competencias del título
A22	Conocimiento y aplicación de los principios, metodologías y ciclos de vida de la ingeniería de software.
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.

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
B7	Preocupación por la calidad
B8	Capacidad de trabajar en un equipo interdisciplinar
B9	Capacidad para generar nuevas ideas (creatividad)
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			
Resultados de aprendizaje	Competencias del título		
Conocer los diferentes tipos de metodologías de desarrollo de software y sus fundamentos.	A22 A25 A30	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.	A22 A27	B1 B4	C1 C3 C4 C6 C8
Utilizar herramientas metodológicas para el desarrollo en entornos colaborativos.	A27	B1 B2 B7 B8 B9	C1 C2 C3 C6 C8
Conocer metodologías y técnicas para la reutilización, evolución y mantenimiento de proyectos.	A22 A30	B1 B2 B3 B8	C1 C3 C6 C7
Adaptar las metodologías a los requisitos éticos, sociales y legales.	A27	B8	C1 C2 C4 C7 C8

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. Metodologías agiles de desarrollo	* El manifiesto Agile * Principios Agile * Características de las metodologías ágiles.
4. Scrum	* Fundamentos y características de Scrum * Roles: Product owner, ScrumMaster, el Equipo * Reuniónes: Planificación de Sprint , Revisión del Sprint, Retrospectiva del Sprint, Reunión diaria de Scrum * Artefactos: Product backlog, Sprint backlog, gráficas Burndown * Prácticas de gestión de proyectos Scrum * Errores comunesen la aplicación de Scrum
5. Programación Extrema (XP)	* Fundamentos y características de XP * Valores y Principios * Prácticas XP: Prácticas Principales y Corolario. * Equipo de desarrollo XP
6. El proceso unificado de desarrollo	* Fundamentos y características del Proceso Unificado * El Lenguaje Unificado de Modelado (UML) * Casos de uso * Centrado en la arquitectura * El Ciclo de vida y las fases del proceso Unificado: Inicio, Elaboración, Construcción y Transición.
7. Desarrollo colaborativo	* Desarrollo en equipo * Desarrollo en abierto. * Gestión de una comunidad de software libre. * Herramientas de trabajo colaborativo.
8. Evolución y mantenimiento del software	* Mantenimiento y evolución del software. * Control de versiones. * Gestión de errores.
9. Aspectos sociales, legales y éticos en el desarrollo software.	* El código ético y de práctica profesional en la Ingeniería del Software de ACS/IEEE-CS * Software libre y Software privativo. * Licenciamiento y propiedad intelectual en el Software. * Interoperabilidad y uso de estándares. * Seguridad y confianza.

Planificación				
Metodologías / pruebas	Competéncias	Horas presenciales	Horas no presenciales / trabajo autónomo	Horas totales
Sesión magistral		21	42	63

Prácticas de laboratorio		14	28	42
Seminario		5	5	10
Prueba de ensayo/desarrollo		2	20	22
Prueba objetiva		3	6	9
Atención personalizada		4	0	4
(*)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 teóricos utilizando 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. El material de las sesiones estará disponible en la plataforma de teleformación, por lo que los estudiantes con dedicación a tiempo parcial podrán seguir la materia. También estará disponible la bibliografía recomendada para cada uno de los temas tratados en la materia.
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. Las prácticas de laboratorio serán presenciales en el aula. Se desarrollará un proyecto conjunto en equipos de 4-8 personas con colaboración entre los diferentes grupos de la clase. Para los alumnos con dedicación a tiempo parcial, se podrán realizar adaptaciones específicas que permitan la realización de las prácticas. Si es posible, participando a través de Internet con un grupo del aula o definiendo un plan de trabajo individual que encaje con su disponibilidad.
Seminario	A lo largo del curso se organizarán seminarios en los que se comentarán a modo de foro de debate las lecturas con sus correspondientes ensayos críticos elaborados por los alumnos. Se tratará de llegar a conclusiones que relacionen los artículos con los objetivos y temarios de la materia, para adquirir conocimientos mediante la puesta en común del trabajo individual realizado por los alumnos del grupo. La participación en los seminarios podrá ser substituida en los alumnos con dedicación a tiempo parcial con la participación en el foro de la asignatura comentando las diferentes lecturas.
Prueba de ensayo/desarrollo	Los alumnos deberán elaborar ensayos que analice de forma crítica varios artículos propuestos por el profesor preferiblemente en su versión original en Inglés sobre los diferentes contenidos de la asignatura. Por ejemplo: artículo sobre metodologías, prácticas, aspectos éticos y legales, etc... Los ensayos se subiran a la facultad virtual y si la herramienta moodle lo permite serán compartidos entre los alumnos una vez haya finalizado el plazo de entrega de cada uno de los trabajos. Los alumnos a tiempo parcial tendrán la información disponible para desarrollar los trabajos en el moodle y siendo los trabajos de redacción individuales no precisa adaptación. Las consultas podrán trasladarse a través del moodle o directamente al profesor a través del correo electrónico o mediante tutorías.
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 Prueba de ensayo/desarrollo	* 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 ensayos críticos, de las lecturas. La atención individualizada será especialmente importante para los alumnos que por causas de una dedicación a tiempo parcial no puedan asistir periodicamente a clase.

Evaluación			
Metodologías	Competéncias	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
Prueba de ensayo/desarrollo		Los alumnos deberán realizar ensayos críticos que analicen de forma crítica un artículo científico o divulgativo sobre los contenidos de la materia o una implementación de unha metodología. Por lo menos la mitad de los artículos estarán disponibles en Inglés para su lectura. 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 se mantiene idéntica a la obtenida en la primera oportunidad al no poder repetirse la evaluación continua y el trabajo en equipos en la segunda oportunidad.La nota de los ensayos críticos de las lecturas se mantiene idéntica a la obtenida en la primera oportunidad, sin embargo existirá la posibilidad de volver a entregar los ensayos críticos de las lecturas. En ese caso se deberá indicar las diferencias y mejoras con la entrega de la primera oportunidad.La nota de la prueba objetiva 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 proba 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 - Beck, Kent ; Andres, Cynthia (2005). Extreme Programming Explained (2nd ed.). Addison Wesley - Jacobson, Ivar ; Booch, Grady; Rumbaugh, James (2000). El proceso unificado de desarrollo de software. Addison Wesley - Fogel, Karl (2010). Creando Software Libre. Edizer, GHANDALF - Schwaber, Ken (2004). Agile Project Management with Scrum. Microsoft Press
Complementária	- Fox, Armando; Patterson, David (2012). Engineering Long-Lasting Software. Strawberry Canyon LLC - Derby, Esther; Larsen Diana (2006). Agile Retrospectives. Making Good Teams Great. Pragmatic Bookshelf - 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 - P. Brooks, Erick (1995). The Mythical Man-Month (Capítulo 16- No Silver Bullet). Addison Wesley Logman Inc.

Recomendaciones
Asignaturas que se recomienda haber cursado previamente
Asignaturas que se recomienda cursar simultáneamente
Ingeniería de Requisitos/614G01222 Aseguramiento de la Calidad/614G01223
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