



Guía Docente				
Datos Identificativos				2013/14
Asignatura (*)	Metodoloxías de Desenvolvemento		Código	614G01224
Titulación	Grao en Enxeñaría Informática			
Descritores				
Ciclo	Período	Curso	Tipo	Créditos
Grao	1º cuatrimestre	Curso de Adaptación Enxeñeiros Téc. en Informática	Obrigatoria	6
Idioma	Galego			
Prerrequisitos				
Departamento	Computación			
Coordinación	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			
Descrición xeral	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 da titulación	
Código	Competencias da titulación
A6	Coñecemento adecuado do concepto de empresa, marco institucional e xurídico da empresa. Organización e xestión de empresas.
A7	Capacidade para deseñar, desenvolver, seleccionar e avaliar aplicacións e sistemas informáticos que aseguren a súa fiabilidade, seguranza e calidade, conforme a principios éticos e á lexislación e normativa vixente.
A8	Capacidade para planificar, concibir, despregar e dirixir proxectos, servizos e sistemas informáticos en todos os ámbitos, liderando a súa posta en marcha e a súa mellora continua e valorando o seu impacto económico e social.
A9	Capacidade para comprender a importancia da negociación, os hábitos de traballo efectivos, o liderado e as habilidades de comunicación en todos os contornos de desenvolvemento de sóftware
A22	Coñecemento e aplicación dos principios, metodoloxías e ciclos de vida da enxeñaría do sóftware.
A24	Coñecemento da normativa e a regulación da informática nos ámbitos nacional, europeo e internacional.
A25	Capacidade para desenvolver, manter e avaliar servizos e sistemas sóftware que satisfagan todos os requisitos do usuario e se comporten de forma fiable e eficiente, sexan accesibles de desenvolver e manter, e cumpran normas de calidade, aplicando as teorías, principios, métodos e prácticas da enxeñaría do sóftware.
A27	Capacidade de dar solución a problemas de integración en función das estratexias, estándares e tecnoloxías dispoñibles.
A28	Capacidade de identificar e analizar problemas, e deseñar, desenvolver, implementar, verificar e documentar solucións sóftware sobre a base dun coñecemento adecuado das teorías, modelos e técnicas actuais.
A29	Capacidade de identificar, avaliar e xestionar os riscos potenciais asociados que se puideren presentar.
A30	Capacidade para deseñar solucións apropiadas nun ou máis dominios de aplicacións utilizando métodos da enxeñaría do sóftware que integren aspectos éticos, sociais, legais e económicos.
B1	Capacidade de resolución de problemas
B2	Traballo en equipo
B3	Capacidade de análise e síntese
B4	Capacidade para organizar e planificar
B5	Habilidades de xestión da información
B6	Toma de decisións



B8	Capacidade de traballar nun equipo interdisciplinar
C1	Expresarse correctamente, tanto de forma oral coma escrita, nas linguas oficiais da comunidade autónoma.
C2	Dominar a expresión e a comprensión de forma oral e escrita dun idioma estranxeiro.
C3	Utilizar as ferramentas básicas das tecnoloxías da información e as comunicacións (TIC) necesarias para o exercicio da súa profesión e para a aprendizaxe ao longo da súa vida.
C4	Desenvolverse para o exercicio dunha cidadanía aberta, culta, crítica, comprometida, democrática e solidaria, capaz de analizar a realidade, diagnosticar problemas, formular e implantar solucións baseadas no coñecemento e orientadas ao ben común.
C6	Valorar criticamente o coñecemento, a tecnoloxía e a información dispoñible para resolver os problemas cos que deben enfrontarse.
C7	Asumir como profesional e cidadán a importancia da aprendizaxe ao longo da vida.
C8	Valorar a importancia que ten a investigación, a innovación e o desenvolvemento tecnolóxico no avance socioeconómico e cultural da sociedade.

Resultados da aprendizaxe			
Competencias de materia (Resultados de aprendizaxe)	Competencias da titulación		
Coñecer os diferentes tipos de metodoloxías de desenvolvemento de software e os seus fundamentos.	A8 A22 A25 A29	B3 B8	C1 C4 C6 C7
Ser capaz de seleccionar a metodoloxía de desenvolvemento máis adecuada e adaptala ás necesidades do proxecto Software e a organización que o desenvolve.	A6 A8 A9 A22 A24 A25 A28	B1 B4 B6	C1 C3 C4 C6 C8
Utilizar ferramentas metodolóxicas para o desenvolvemento en entornos colaborativos.	A7 A22 A30	B1 B2 B5 B8	C1 C2 C3 C6 C8
Coñecer metodoloxías e técnicas para a reutilización, evolución e mantemento de proxectos.	A22 A27 A29 A30	B1 B2 B3 B8	C1 C3 C6 C7
Adaptar as metodoloxías aos requisitos éticos, sociais e legais.	A6 A9 A22 A27	B6 B8	C1 C2 C4 C7

Contidos	
Temas	Subtemas
1. Introducción	<ul style="list-style-type: none"> <li>* Metodoloxía vs método.</li> <li>* Metodoloxías de desenvolvemento de software.</li> <li>* O ciclo de vida do software.</li> </ul>



2. Metodoloxías Clásicas	<ul style="list-style-type: none"> <li>* Cascada</li> <li>* Prototipado</li> <li>* Espiral</li> <li>* Incremental</li> <li>* Desenvolvemento rápido de aplicacións</li> </ul>
3. O proceso unificado de desenvolvemento	<ul style="list-style-type: none"> <li>* Características</li> <li>* A Linguaxe Unificada de Modelado (UML)</li> <li>* Casos de uso</li> <li>* A importancia da arquitectura</li> <li>* Ciclo de vida</li> </ul>
4. Metodoloxías axiles de desenvolvemento	<ul style="list-style-type: none"> <li>* O Manifesto Agile</li> <li>* Características das metodoloxías áxiles.</li> <li>* A metodoloxía SCRUM</li> <li>* Outras metodoloxías áxiles</li> </ul>
5. Programación Extrema (XP)	<ul style="list-style-type: none"> <li>* Valores e Principios</li> <li>* Técnicas</li> <li>* Equipo de desenvolvemento</li> <li>* Aplicacións prácticas</li> </ul>
6. Desenvolvemento colaborativo	<ul style="list-style-type: none"> <li>* Desenvolvemento en equipo</li> <li>* Ferramentas de traballo colaborativo.</li> <li>* Desenvolvemento en aberto.</li> <li>* Xestión dunha comunidade de software libre.</li> <li>* Estándares abertos.</li> <li>* Colaborando con provedores.</li> </ul>
7. Evolución e mantemento do software	<ul style="list-style-type: none"> <li>* Legacy coma software de éxito.</li> <li>* Mantemento e evolución do software.</li> <li>* Control de versións.</li> <li>* Xestión de erros.</li> </ul>
8. Aspectos sociais, legais e éticos no desenvolvemento software	<ul style="list-style-type: none"> <li>* Licenzamento e propiedade intelectual no Software.</li> <li>* Software libre e Software privativo.</li> <li>* Interoperabilidade e uso de estándares.</li> <li>* Seguridade e confianza.</li> </ul>

### Planificación

Metodoloxías / probas	Horas presenciais	Horas non presenciais / traballo autónomo	Horas totais
Sesión maxistral	21	42	63
Prácticas de laboratorio	14	28	42
Seminario	5	10	15
Traballos tutelados	2	16	18
Proba obxectiva	3	6	9
Atención personalizada	3	0	3

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

### Metodoloxías

Metodoloxías	Descrición
--------------	------------



Sesión maxistral	Clases expositivas de presentación dos coñecementos teóricos empregando diferentes recursos: pizarra, proxección de material en formato electrónico, apuntes en formato electrónico e os recursos facilitados pola equipa docente da asignatura na facultade virtual.
Prácticas de laboratorio	Prácticas deseñadas pola equipa docente da asignatura baseadas nos coñecementos que cada estudante vai adquirindo. Estes traballos serán desenvolvidos en grupo. Traballarase no uso de ferramentas de apoio a implantación dunha metodoloxía un proxecto de desenvolvemento Software.
Seminario	Titorías de grupos reducidos (TGRs) nos que se proporán actividades relacionadas cos coñecementos adquiridos en teoría ou práctica
Traballos tutelados	Os alumnos deberán realizar un informe que analice e avalíe de forma crítica unha ferramenta de apoio para a implementación dunha metodoloxía.  Os traballos publicaranse na facultade virtual e serán compartidos entre os alumnos para axudarlles na selección de ferramentas software. Potenciarase a realización de traballos en inglés.
Proba obxectiva	Proba escrita mediante a que se valoran os coñecementos adquiridos polo estudiantado. Cada estudante deberá aplicar tanto os seus coñecementos tanto a nivel teórico coma a nivel práctico.

## Atención personalizada

Metodoloxías	Descrición
Prácticas de laboratorio	* A atención personalizada consistirá na realización de titorías presenciais ou a través do campus virtual para resolver dúbidas sobre os contidos da asignatura.
Seminario	* Realizarase o seguemento personalizado das prácticas de laboratorio.
Traballos tutelados	* Así mesmo revisaranse persoalmente cos alumnos os resultados dos traballos dos seminarios así como na selección e avaliación do traballo tutelado.

## Avaliación

Metodoloxías	Descrición	Cualificación
Proba obxectiva	Proba escrita realizada ao final do curso sobre contidos teórico-prácticos.  A proba obxectiva é obrigatoria para aprobar a materia. É obrigatorio obter unha nota mínima de 3,5 sobre 10 para poder facer media cos outros elementos avaliábles. En caso de non chegar á nota mínima implicará que non se poida obter máis dun 4,5 na nota final da materia	40
Prácticas de laboratorio	Avaliación continua das prácticas propostas ao longo do curso Entre os aspectos a considerar á hora de valorar as prácticas, atópanse: - Rigor no acadamento dos obxectivos perseguidos na práctica empregando as técnicas propostas na asignatura. - Asimilación dos conceptos perseguidos pola práctica. - Orixinalidade nas propostas acometidas durante a realización da práctica. - Responsabilidade na entrega das prácticas en tempo e forma, así coma no uso axeitado dos recursos habilitados para esa fin. - Valoración do traballo en grupo	40



Traballos tutelados	Os alumnos deberán realizar un informe que analice e avalíe de forma crítica unha ferramenta de apoio para a implementación dunha metodoloxía.  Os traballos publicaranse na facultade virtual e serán compartidos entre os alumnos para axudarlles na selección de ferramentas software. Potenciarase a realización de traballos en inglés.  Valorarase a calidade do traballo, a expresión, o rigor académico. Valorarase adicionalmente o uso do inglés.	20
---------------------	---	----

### Observacións avaliación

Aspectos a ter en conta na avaliación da segunda oportunidade de Xullo:

\* A nota de prácticas de laboratorio e do traballo tutelado mantense idéntica á obtida na primeira oportunidade, existirá a posibilidade de voltar a entregar o traballo tutelado. Nese caso deberase indicar as diferencias e melloras coa entrega da primeira oportunidade.\* A nota da proba obxectiva só se mantén no caso de que sexa igual ou superior a cinco na primeira oportunidade. No caso de non superar o cinco na primeira oportunidade será obrigatorio repetir a proba obxectiva.Considerarase que un alumno preséntase á segunda oportunidade se volve entregar o traballo tutelado ou presentase á proba obxectiva.

### Fontes de información

<b>Bibliografía básica</b>	<ul style="list-style-type: none"><li>- Larman, Craig (2004). Agile &amp; Iterative Development. Addison Wesley</li><li>- Fogel, Karl (2010). Creando Software Libre. Edizer, GHANDALF</li><li>- Jacobson, Ivar ; Booch, Grady; Rumbaugh, James (2000). El proceso unificado de desarrollo de software. Addison Wesley</li><li>- Beck, Kent ; Andres, Cynthia (2005). Extreme Programming Explained (2nd ed.). Addison Wesley</li><li>- Pitone, Dan; Miles, Russ (2008). Head First Software Development. O'Reilly</li><li>- Fowler, Martin (2000). Refactoring: Improving the Design of Existing Code. Addison Wesley</li></ul>
<b>Bibliografía complementaria</b>	<ul style="list-style-type: none"><li>- Martin, Robert C (2008). Clean Code: A Handbook of Agile Software Craftsmanship. Prentice Hall</li><li>- Fox, Armando; Patterson, David (2012). Engineering Long-Lasting Software. Strawberry Canyon LLC</li><li>- Stallman, Richard M (2010). Free Software, Free Society: Selected Essays of Richard M. Stallman. Free Software Foundation</li><li>- Raymond, Eric S. (2000). The Cathedral and the Bazaar. O'Reilly</li></ul>

### Recomendacións

#### Materias que se recomenda ter cursado previamente

Proxectos de Desenvolvemento Software/614G01226

#### Materias que se recomenda cursar simultaneamente

Enxeñaría de Requisitos/614G01222

Aseguramento da Calidade/614G01223

#### Materias que continúan o temario

### 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