Guia docente				
Datos Identificativos				2019/20
Asignatura (*)	Diseño de sistemas de información		Código	614502007
Titulación	Mestrado Universitario en Enxeñaría Informática (plan 2012)			
Descriptores				
Ciclo	Periodo	Curso	Tipo	Créditos
Máster Oficial	1º cuatrimestre	Primero	Obligatoria	6
Idioma	GallegoInglés			
Modalidad docente	Presencial			
Prerrequisitos				
Departamento	Ciencias da Computación e Tecnoloxías da InformaciónComputación			
Coordinador/a	Sanchez Penas, Juan Jose	Correo electrónico	juan.jose.sanchez.penas@udc.es	
Profesorado	Sanchez Penas, Juan Jose	Correo electrónico	juan.jose.sanchez.penas@udc.es	
Web	https://moodle.udc.es/course/view.php?id=32116			
Descripción general	Revisaremos conceptos avanzados relacionados con todos los aspectos del diseño software, incluyendo patrones de diseño y arquitecutra, diseño orientado a componentes, calidad en el diseño, evolución del software, métricas y complejidad software, o accesibilidad. El objetivo será consolidar esos conceptos estudiando proyectos del mundo real desde unha perspectiva profesional. El idioma principal de la asignatura será el inglés.			

Competencias / Resultados del título	
Código	Competencias / Resultados del título
A4	Capacidad para modelar, diseñar, definir la arquitectura, implantar, gestionar, operar, administrar y mantener aplicaciones, redes, sistemas, servicios y contenidos informáticos.
A14	Capacidad para conceptualizar, diseñar, desarrollar y evaluar la interacción persona?ordenador de productos, sistemas, aplicaciones y servicios informáticos.
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).
B10	Capacidad para proyectar, calcular y diseñar productos, procesos e instalaciones en todos los ámbitos de la ingeniería informática
B13	Capacidad para el modelado matemático, cálculo y simulación en centros tecnológicos y de ingeniería de empresa, particularmente en tareas de investigación, desarrollo e innovación en todos los ámbitos relacionados con la Ingeniería en Informática
B14	Capacidad para la elaboración, planificación estratégica, dirección, coordinación y gestión técnica y económica de proyectos en todos los ámbitos de la Ingeniería en Informática siguiendo criterios de calidad y medioambientales
B17	Capacidad para la aplicación de los conocimientos adquiridos y de resolver problemas en entornos nuevos o poco conocidos dentro de contextos más amplios y multidisciplinares, siendo capaces de integrar estos conocimientos
B21	Poseer y comprender conocimientos que aporten una base u oportunidad de ser originales en el desarrollo y/o aplicación de ideas, a menudo en un contexto de investigación
B22	Que los estudiantes sepan aplicar los conocimientos adquiridos y su capacidad de resolución de problemas en entornos nuevos o poco conocidos dentro de contextos más amplios (o multidisciplinares) relacionados con su área de estudio
B23	Que los estudiantes sean capaces de integrar conocimientos y enfrentarse a la complejidad de formular juicios a partir de una información que, siendo incompleta o limitada, incluya reflexiones sobre las responsabilidades sociales y éticas vinculadas a la aplicación de sus conocimientos y juicios
B24	Que los estudiantes sepan comunicar sus conclusiones, y los conocimientos y razones últimas que las sustentan, a públicos especializados y no especializados de un modo claro y sin ambigüedades

B25	Que los estudiantes posean las habilidades de aprendizaje que les permitan continuar estudiando de un modo que habrá de ser en gran medida autodirigido o autónomo
C1	Expresarse correctamente, tanto de forma oral como escrita, en las lenguas oficiales de la comunidad autónoma.
C6	Valorar críticamente el conocimiento, la tecnología y la información disponible para resolver los problemas con los que deben enfrentarse

Resultados de aprendizaje			
Resultados de aprendizaje	Competencias / Resultados del título		
Comprender y saber diseñar sistemas de información mediante patrones y siguiendo pautas de calidad.	AP4 AP14	BP1 BP2 BP3 BP4 BP5 BP6 BP7 BP8 BP9 BP10 BP13 BP14 BP17 BM1 BM2 BM3 BM4 BM5	CP1 CP6

Contenidos	
Tema	Subtema
Introducción al diseño de software avanzado	Importancia del diseño software Metodologías y procesos de diseño y desarrollo software Patrones de diseño y arquitectura, diseño orientado a componentes Evolución del software, calidad del diseño, métricas y complejidad del software Accesibilidad del software Ejemplos del mundo real de diseño software complejo
Conceptos avanzados de diseño software	Lenguajes y herramientas usadas para el diseño software Patrones de diseño Patrones de arquitectura Patrones de interfaz de usuario y experiencia de usuario Introducción a la refactorización y la evolución del software
Conceptos avanzados de calidad en el diseño software	Software y calidad en el diseño Métricas y complejidad del software Evaluación y verificación de sistemas software
Conceptos avanzados de accesibilidad del software	Importancia de la accesibilidad del software Accesibilidad del software y diseño software Standards de accesibilidad en el software Herramientas y tecnologías para la accesibilidad del software Casos de estudio de accesibilidad del software

Casos de estudio del mundo real	Revisión de algunos sistemas software populares y complejos Diseño software en proyectos de software libre utilizados en la industria Análisis en profundidad del diseño, las herramientas, la calidad y la accesibilidad en varios proyectos de software libre (por ejemplo WebKit, GNOME&KDE, Linux, MeeGo/Tizen, etc.)

Planificación				
Metodologías / pruebas	Competencias / Resultados	Horas lectivas (presenciales y virtuales)	Horas trabajo autónomo	Horas totales
Sesión magistral	A4 B7 B10 B14 B17	10	15	25
Estudio de casos	A14 B2 B5 B6 B13	10	20	30
Prueba objetiva	B1 B3	5	0	5
Taller	B21 C6	10	20	30
Lecturas	B24 B25	0	10	10
Prácticas de laboratorio	B4 B8 B9	10	20	30
Eventos científicos y/o divulgativos	B23	0	8	8
Foro virtual	B22 C1	0	10	10
Atención personalizada		2	0	2
(*)Los datos que aparecen en la tabla de planificación són de carácter orientativo, considerando la heterogeneidad de los alumnos				

Metodologías	
Metodologías	Descripción
Sesión magistral	Invitaremos ingenieros y managers relevantes de la industria de las TIC, con el objetivo de impartir sesiones magistrales que complementen los contenidos formativos de la asignatura.
Estudio de casos	Revisaremos proyectos reales y discutiremos la forma en la el contenido teórico estudiado en la asignatura es aplicado en ellos. Nos enfocaremos principalmente en proyectos de software libre, ya que tenemos acceso a todo el código fuente y material de diseño.
Prueba objetiva	Examen escrito, en el que el estudiante tendrá que demostrar tanto los conocimientos teóricos adquiridos como la capacidad para resolver problemas prácticos.
Taller	Sesiones de análisis, diseño y discusión práctica, con los estudiantes organizados en grupos, supervisados por el profesor.
Lecturas	El profesor proporcionará a los estudiantes artículos y capítulos de libros relevantes, relacionados con el contenido teórico del curso, y el estudiante tendrá que hacer una lectura crítica de los mismos y preparar un resumen que será revisado por el profesor o por toda la clase, dependiendo del caso.
Prácticas de laboratorio	Ejercicios prácticos de diseño y desarrollo, con los estudiantes organizados en grupos, supervisados por el profesor.
Eventos científicos y/o divulgativos	Como complemento de las clases teóricas y prácticas, se recomendará a los alumnos la asistencia (en persona o en remoto) a conferencias relacionadas con el diseño y desarrollo de software.
Foro virtual	Todos los temas estudiados en las clases, talleres y tiempo práctico de laboratorio tendrán su continuidad en los foros online. Se tratará de estimular la conversación en ellos, y de abrir nuevos temas de debate proponiendo enlaces extra que complementen el conocimiento de los alumnos en temas colaterales que puedan ser de su interés.

Atención personalizada	
Metodologías	Descripción

Foro virtual Lecturas Prácticas de laboratorio	La atención personal al estudiante incluye, en este caso, no sólo el clásico tiempo de tutorías, o el apoyo virtual usando los recursos online, sino las siguientes acciones: - Se seguirá constantemente el trabajo del estudiante en las tareas supervisadas que serán propuestas a lo largo de la duración de la asignatura. - Evaluación crítica de los resultados obtenidos en los trabajos prácticos desarrollados por el estudiante. - Comunicación constante con el objetivo de resolver los problemas encontrados por el estudiante para comprender los contenidos expuestos en las clases o las dificultades de las tareas propuestas por el profesor.

Evaluación			
Metodologías	Competencias / Resultados	Descripción	**Calificación**
Prueba objetiva	B1 B3	Examen por escrito con 3 partes: preguntas teóricas cortas, preguntas más prácticas en las que los estudiantes puedan elaborar con más detenimiento las respuestas a cuestiones planteadas, y un problema real específico de diseño de software.	50
Taller	B21 C6	La evaluación de las tareas prácticas será continua a lo largo del curso, y se basará en una presentación final al profesor. En la evaluación se tendrán en cuenta los siguientes aspectos: - Capacidad para trabajar en grupo. - Capacidad personal para hacer el trabajo y explicarlo. - Capacidad para ajustarse a los objetivos de las tareas. - Capacidad para aplicar conocimiento adquirido durante las clases teóricas. - Pensamiento crítico y capacidad para innovar y encontrar soluciones a problemas. - Capacidad para entregar las tareas a tiempo.	50

Observaciones evaluación
El resumen de la distribución de pesos en las evaluaciones es el siguiente: el 50% de la nota derivará del examen escrito, y el otro 50% de un conjunto de trabajos prácticos que serán realizados a lo largo del curso. Es necesario tener una nota mínima de aprobado tanto en el examen escrito como en el conjunto de trabajos prácticos.Aquellos estudiantes con matrícula a tiempo parcial o cualquier circunstancia que impida la asistencia a las clases, deberán contactar con los docentes para determinar alternativas al seguimiento y la evaluación de la asignatura.

Fuentes de información

Básica	Gamma, E.; Helm, R.; Johnson, R. y Vlissides J. (1996). Design Patterns: Elements of Reusable Object-oriented Software. Addison Wesley Martin Fowler with contributions by Kent Beck, John Brant, William Opdyke and Don Roberts. Refactoring: Improving the Design of Existing Code. Addison-Wesley, 1999. Michael Jackson. Problem Analysis and Structure. In Proceedings of NATO Summer School, Marktoberdorf, August 2000 (in publication). Available here. Michael Jackson. Problem Frames: Analyzing and Structuring Software Development Problems. Addison Wesley, 2001. G. Polya. How to Solve It. 2nd ed., Princeton University Press, 1957. Diomidis Spinellis. Code Quality: The Open Source Perspective. Addison Wesley, Boston, MA, 2006. Stephen H. Kan. Metrics and Models in Software Quality Engineering. Addison-Wesley, Boston, MA, second edition, 2002. Henry, Shawn Lawton. Integrating Accessibility Throughout Design. Lulu.com. February 2007Gamma, E.; Helm, R.; Johnson, R. y Vlissides J. (1996). Design Patterns: Elements of Reusable Object-oriented Software. Addison WesleyMartin Fowler with contributions by Kent Beck, John Brant, William Opdyke and Don Roberts. Refactoring: Improving the Design of Existing Code. Addison-Wesley, 1999.Michael Jackson. Problem Analysis and Structure. In Proceedings of NATO Summer School, Marktoberdorf, August 2000 (in publication). Available here.Michael Jackson. Problem Frames: Analyzing and Structuring Software Development Problems. Addison Wesley, 2001.G. Polya. How to Solve It. 2nd ed., Princeton University Press, 1957.Diomidis Spinellis. Code Quality: The Open Source Perspective. Addison Wesley, Boston, MA, 2006.Stephen H. Kan. Metrics and Models in Software Quality Engineering. Addison-Wesley, Boston, MA, second edition, 2002.Henry, Shawn Lawton. Integrating Accessibility Throughout Design. Lulu.com. February 2007
Complementária	Rumbaugh, J.; Jacobson, I. y Booch, J. (2004). The Unified Modeling Language Reference Manual. Addison Wesley Booch J.; Rumbaugh J. y Jacobson I. (2005). The Unified Modeling Language User Guide. Addison Wesley Page-Jones, M. (2004). Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development. Prentice Hall PTR Cooper J. (2000). Java Design Patterns: A Tutorial. Addison Wesley Stevens, P. y Pooley, R. (1999). Using UML. Software Engineering with Objects and Components. Addison Wesley Ho-Won Jung, Seung-Gweon Kim, and Chang-Sin Chung. Measuring software product quality: A survey of ISO/IEC 9126. IEEE Software, 21(5):10?13, September/October 2004. Omar Alshathry, Helge Janicke, "Optimizing Software Quality Assurance," compsacw, pp. 87?92, 2010 IEEE 34th Annual Computer Software and Applications Conference Workshops, 2010. Robert L. Glass. Building Quality Software. Prentice Hall, Upper Saddle River, NJ, 1992. Roland Petrasch, "The Definition of? Software Quality?: A Practical Approach", ISSRE, 1999Rumbaugh, J.; Jacobson, I. y Booch, J. (2004). The Unified Modeling Language Reference Manual. Addison WesleyBooch J.; Rumbaugh J. y Jacobson I. (2005). The Unified Modeling Language User Guide. Addison WesleyPage-Jones, M. (2004). Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development. Prentice Hall PTRCooper J. (2000). Java Design Patterns: A Tutorial. Addison WesleyStevens, P. y Pooley, R. (1999). Using UML. Software Engineering with Objects and Components. Addison WesleyHo-Won Jung, Seung-Gweon Kim, and Chang-Sin Chung. Measuring software product quality: A survey of ISO/IEC 9126. IEEE Software, 21(5):10?13, September/October 2004.Omar Alshathry, Helge Janicke, "Optimizing Software Quality Assurance," compsacw, pp. 87?92, 2010 IEEE 34th Annual Computer Software and Applications Conference Workshops, 2010.Robert L. Glass. Building Quality Software. Prentice Hall, Upper Saddle River, NJ, 1992.Roland Petrasch, "The Definition of? Software Quality?: A Practical Approach", ISSRE, 1999

Recomendaciones
Asignaturas que se recomienda haber cursado previamente
Asignaturas que se recomienda cursar simultáneamente
Análisis de sistemas de información/614502006
Asignaturas que continúan el temario

Dirección de proyectos/614502002
Calidad, seguridad y auditoría informática/614502003
Arquitecturas y plataformas móbiles/614502005
Prácticas en empresa/614502011
Trabajo fin de máster/614502012
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