Guia docente					
Datos Identificativos					2013/14
Asignatura (*)	Diseño Software			Código	614G01015
Titulación	Grao en Enxeñaría Informática				
Descriptores					
Ciclo	Periodo	Curso		Tipo	Créditos
Grado	1º cuatrimestre	Segundo		Obligatoria	6
Idioma	Castellano				
Prerrequisitos					
Departamento	Computación				
Coordinador/a	Mosqueira Rey, Eduardo		Correo electrónico	eduardo.mosqueira@udc.es	
Profesorado	Alonso Betanzos, Maria Amparo Alonso Ríos, David Fontenla Romero, Oscar Hernandez Pereira, Elena Maria Monroy Camafreita, Juan Mosqueira Rey, Eduardo Paris Fernandez, Javier Pérez Sánchez, Beatriz Sanchez Penas, Juan Jose		Correo electrónico	amparo.alonso.betanzos@udc.es david.alonso@udc.es oscar.fontenla@udc.es elena.hernandez@udc.es juan.monroy@udc.es eduardo.mosqueira@udc.es javier.paris@udc.es beatriz.perezs@udc.es juan.jose.sanchez.penas@udc.es	
Web					
Descripción general	O Deseño Software é unha fase chave dentro do ciclo de vida do software que establece a ligazón entre os requisitos dun sistema e a sua posterior implementación. O deseño máis habitual hoxe en día é o deseño baseado na orientación a obxectos, que consiste en desenvolver un programa en base a obxectos que intercambian mensaxes. Esta materia introducirá aos alumnos nos elementos e propiedades básicas da orientación a obxectos usando unha linguaxe orientada a obxectos como Java. Aprenderase tamén a como reflectir os artefactos propios do deseño nunha linguaxe de modelado coma a Linguaxe Unificada de Modelado (UML). Finalmente presentaranse aqueles principios básicos que representan un bo deseño e aprenderase a identificar aqueles problemas típicos de deseño e as súas solucións máis comúns representadas como patróns de deseño.				

Competencias de la titulación	
Código	Competencias de la titulación
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.
A13	Conocimiento, diseño y utilización de forma eficiente de los tipos y estructuras de datos más adecuados a la resolución de un problema.
A14	Capacidad para analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados.
B1	Capacidad de resolución de problemas
B2	Trabajo en equipo
B3	Capacidad de análisis y síntesis
B6	Toma de decisiones
B7	Preocupación por la calidad
B9	Capacidad para generar nuevas ideas (creatividad)
C1	Expresarse correctamente, tanto de forma oral como escrita, en las lenguas oficiales de la comunidad autónoma.
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.
C7	Asumir como profesional y ciudadano la importancia del aprendizaje a lo largo de la vida.

Resultados de aprendizaje			
Competencias de materia (Resultados de aprendizaje)	Competencias de la titulación		
Identificar el diseño software como una de las fases del ciclo de vida del software	A7 A13 A14	B1 B7	C1 C3 C7
Conocer los principios y propiedades básicas de la orientación a objetos	A7 A13 A14	B1 B2 B3	C1 C3 C7
Plasmar un diseño software utilizando los artefactos propios de un lenguaje de modelado como UML	A7 A13 A14	B1 B2 B3 B6 B7 B9	C1 C3 C7
Conocer los principios básicos que representan un buen diseño software	A7 A13 A14	B1 B2 B3	C1 C3 C7
Identificar problemas típicos de diseño y sus soluciones más comunes	A7 A13 A14	B1 B2 B3 B6	C1 C3 C7
Usar un diseño como guía para la implementación del software	A7 A13 A14	B1 B2 B3 B6 B9	C1 C3 C7

Contenidos	
Tema	Subtema
1. Introducción	? Diseño software ? Diseño orientado a objetos
2. Elementos Básicos de la Orientación a Objetos	? Clases y objetos ? Identidad de objetos ? Estado de objetos ? Comportamento de objetos
3. Propiedades Básicas de la Orientación a Objetos	? Abstracción y encapsulamiento ? Modularidad ? Composición ? Herencia ? Polimorfismo ? Tipificación ? Ligadura dinámica
4. Lenguaje Unificado de Modelado (UML)	? Elementos básicos del UML ? Diseño estático: Diagramas de clases ? Diseño dinámico: Diagramas de interacción ? Otros diagramas
5. Principios de Diseño	? Principios SOLID ? Diseño por contrato y subcontratación ? Tipos de herencia

6. Patrones de Diseño	? Principios y patrones ? Patrones GoF ? Otros patrones

Planificación			
Metodologías / pruebas	Horas presenciales	Horas no presenciales / trabajo autónomo	Horas totales
Sesión magistral	30	45	75
Prácticas de laboratorio	20	30	50
Seminario	10	10	20
Prueba objetiva	3	0	3
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	Clases magistrales en la exposición de los conocimientos teóricos usando diferentes recursos: pizarra, proyección de material en formato electrónico, apuntes en formato electrónico y los recursos facilitados por el docente de la asignatura en la facultad virtual.
Prácticas de laboratorio	Prácticas diseñadas por el docente de la asignatura basadas en los conocimientos que el estudiante va adquiriendo. Los estudiantes desarrollarán estos trabajos preferiblemente en grupo (que serán como máximo de dos personas). Se utilizará una herramienta de modelado para construir los artefactos de diseño y se empleará un lenguaje orientado a objetos (Java) para realizar la implementación de los mismos.
Seminario	Tutorías de grupos reducidos (TGRs) en los que se propondrán actividades relacionadas con los conocimientos adquiridos en teoría o práctica
Prueba objetiva	Prueba escrita mediante la que se valora los conocimientos adquiridos por el estudiante. Cada estudiante deberá aplicar sus conocimientos tanto a nivel teórico como a nivel práctico.

Atención personalizada	
Metodologías	Descripción
Prácticas de laboratorio Seminario	La atención personalizada al estudiante comprende no solo las tutorías, presenciales o virtuales, para la discusión de dudas, sino también las siguientes actuaciones: - Seguimiento de la labor realizada en las prácticas de laboratorio propuestos por el docente. - Evaluación de los resultados obtenidos en las prácticas, participación en seminarios realizados por el estudiante. - Encuentros personalizados para resolver dudas sobre los contenidos de la asignatura.

Evaluación		
Metodologías	Descripción	Calificación

Prácticas de laboratorio	Dos boletines de ejercicios basados en la programación en Java y en la orientación a objetos. Una práctica de diseño centrada en el uso de principios y patrones de diseño. La nota de prácticas se reparte equitativamente entre los dos boletines y la práctica de diseño Las prácticas son obligatorias y es necesario obtener una nota mínima de 4 para poder hacer media con los otros elementos evaluables. En caso de no llegar a la nota mínima implicará que no se pueda obtener más de un 4,5 en la nota final de la asignatura Una alumno se considerará presentado a la asignatura si presenta las prácticas de la misma (independientemente de si se presenta o no a la prueba objetiva)	35
Seminario	Evaluación continua de actividades propuestas a lo largo del curso	5
Prueba objetiva	Prueba escrita realizada al final del curso sobre contenidos teórico-prácticos. La prueba objetiva es obligatoria para aprobar la asignatura y también es obligatorio obtener una nota mínima de 4 para poder hacer media con los otros elementos evaluables. En caso de no llegar a la nota mínima implicará que no se pueda obtener más de un 4,5 en la nota final de la asignatura	60

Observaciones evaluación
Aspectos a tener en cuenta para la evaluación de segunda oportunidad (Julio): La nota de la prueba objetiva (examen teórico-práctico) se guardará si es igual o mayor que 5 sobre 10.La nota de los seminarios (TGRs) y de las prácticas de laboratorio se guardará para la segunda oportunidadExistirá la posibilidad de mejorar la nota de las prácticas de laboratorio tal y como se indicará en la plataforma virtualEn la segunda oportunidad los porcentajes son iguales y también rigen las normas de un mínimo de un 4 en prácticas y prueba objetiva. Se considerará que un alumno se presenta a la segunda oportunidad si vuelve a entregar las prácticas o se presenta a la prueba objetiva.

Fuentes de información	
Básica	- Gamma, E.; Helm, R.; Johnson, R. y Vlissides J. (1996). Design Patterns: Elements of Reusable Object-oriented Software.. Addison Wesley - Booch J.; Rumbaugh J. y Jacobson I. (2006). El Lenguaje Unificado de Modelado (2ª ed.). Addison Wesley - Eckel, B. (2007). Piensa en Java (4ª ed.). Prentice-Hall - Arnold K., Gosling J. y Holmes D. (2005). The Java Programming Language. Prentice Hall - Martin, R.C. (2004). UML para programadores Java. Pearson
Complementária	- Freeman, E., Freeman, E., Bates, B. (2004). Head First Design Patterns. O'Reilly - Sierra, K., Bates, B. (2005). Head First Java. O?Reilly - Grand M. (2002). Patterns in Java. John Wiley & Sons - Rumbaugh, J.; Jacobson, I. y Booch, J. (2004). The Unified Modeling Language Reference Manual. Addison Wesley - Stevens, P. y Pooley, R. (2006). Using UML. Software Engineering with Objects and Components. Addison Wesley

Recomendaciones
Asignaturas que se recomienda haber cursado previamente
Proceso Software/614G01019 Interfaces Hombre Máquina/614G01022 Internet y Sistemas Distribuidos/614G01023 Ingeniería de Requisitos/614G01027
Asignaturas que se recomienda cursar simultáneamente
Paradigmas de Programación/614G01014
Asignaturas que continúan el temario
Programación I/614G01001 Programación II/614G01006
Otros comentarios

La asignatura asume que los estudiantes saben programar y conocen las estructuras de datos (Programación II) aunque nunca han utilizado un lenguaje orientado a objetos. Al principio del curso, y según se van introduciendo los conceptos propios de la orientación a objetos, los alumnos se familiarizan con los aspectos básicos del lenguaje de programación Java.

(*) 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