



Guía Docente				
Datos Identificativos				2013/14
Asignatura (*)	Deseño Software		Código	614G01015
Titulación	Grao en Enxeñaría Informática			
Descriptorios				
Ciclo	Período	Curso	Tipo	Créditos
Grao	1º cuatrimestre	Segundo	Obrigatoria	6
Idioma	Castelán			
Prerrequisitos				
Departamento	Computación			
Coordinación	Mosqueira Rey, Eduardo		Correo electrónico	eduardo.mosqueira@udc.es
Profesorado	Alonso Betanzos, Maria Amparo		Correo electrónico	amparo.alonso.betanzos@udc.es
	Alonso Ríos, David			david.alonso@udc.es
	Fontenla Romero, Oscar			oscar.fontenla@udc.es
	Hernandez Pereira, Elena Maria			elena.hernandez@udc.es
	Monroy Camafreita, Juan			juan.monroy@udc.es
	Mosqueira Rey, Eduardo			eduardo.mosqueira@udc.es
	Paris Fernandez, Javier			javier.paris@udc.es
	Pérez Sánchez, Beatriz			beatriz.perezs@udc.es
	Sanchez Penas, Juan Jose			juan.jose.sanchez.penas@udc.es
Web				
Descrición xeral	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 súa 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 da titulación	
Código	Competencias da titulación
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.
A13	Coñecemento, deseño e utilización de forma eficiente dos tipos e estruturas de datos máis adecuados á resolución dun problema.
A14	Capacidade para analizar, deseñar, construír e manter aplicacións de forma robusta, segura e eficiente, elixindo o paradigma e as linguaxes de programación máis adecuados.
B1	Capacidade de resolución de problemas
B2	Traballo en equipo
B3	Capacidade de análise e síntese
B6	Toma de decisións
B7	Preocupación pola calidade
B9	Capacidade para xerar novas ideas (creatividade)
C1	Expresarse correctamente, tanto de forma oral coma escrita, nas linguas oficiais da comunidade autónoma.
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.
C7	Asumir como profesional e cidadán a importancia da aprendizaxe ao longo da vida.



Resultados da aprendizaxe			
Competencias de materia (Resultados de aprendizaxe)	Competencias da titulación		
Identificar o deseño software como unha das fases do ciclo de vida do software	A7 A13 A14	B1 B7	C1 C3 C7
Coñecer os principios e propiedades básicas da orientación a obxectos	A7 A13 A14	B1 B2 B3	C1 C3 C7
Plasmar un deseño software utilizando os artefactos propios dunha linguaxe de modelado coma UML	A7 A13 A14	B1 B2 B3 B6 B7 B9	C1 C3 C7
Coñecer os principios básicos que representan un bo deseño software	A7 A13 A14	B1 B2 B3	C1 C3 C7
Identificar problemas típicos de deseño e as súas solucións máis comúns	A7 A13 A14	B1 B2 B3 B6	C1 C3 C7
Usar un deseño como guía para a implementación do software	A7 A13 A14	B1 B2 B3 B6 B9	C1 C3 C7

Contidos	
Temas	Subtemas
1. Introducción	? Deseño software ? Deseño orientado a obxectos
2. Elementos Básicos da Orientación a Obxectos	? Clases e obxectos ? Identidade de obxectos ? Estado de obxectos ? Comportamento de obxectos
3. Características Básicas da Orientación a Obxectos	? Abstracción e encapsulamento ? Modularidade ? Composición ? Herencia ? Polimorfismo ? Tipificación ? Ligadura Dinámica
4. Linguaxe Unificada de Modelado (UML)	? Elementos básicos do UML ? Deseño estático: Diagramas de clases ? Deseño dinámico: Diagramas de interacción ? Outros diagramas
5. Principios de Deseño	? Principios SOLID ? Deseño por contrato e subcontratación ? Tipos de herencia



6. Padróns de Deseño	? Principios e patróns ? Patrones GoF ? Outros patróns
----------------------	--

Planificación			
Metodoloxías / probas	Horas presenciais	Horas non presenciais / traballo autónomo	Horas totais
Sesión maxistral	30	45	75
Prácticas de laboratorio	20	30	50
Seminario	10	10	20
Proba obxectiva	3	0	3
Atención personalizada	2	0	2
*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 preferiblemente en grupo (que serán como máximo de dúas persoas). Empregarase unha ferramenta de modelado para construír os artefactos de deseño e aplicarase unha linguaxe orientada a obxectos (Java) para realizar a implementación dos mesmos.
Seminario	Titorías de grupos reducidos (TGRs) nos que se proporán actividades relacionadas cos coñecementos adquiridos en teoría ou práctica
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 Seminario	A atención personalizada ao estudiantado comprende non só as titorías, presenciais ou virtuais, para a discusión de dúbidas, senón tamén as seguintes actuacións: - Seguemento do labor realizado nas prácticas de laboratorio propostas pola equipa docente. - Avaliación dos resultados obtidos nas prácticas, participación en seminarios realizados por cada estudante. - Encontros personalizados para resolver dúbidas sobre os contidos da asignatura.

Avaliación		
Metodoloxías	Descrición	Cualificación



Prácticas de laboratorio	Dous boletíns de exercicios baseados na programación en Java e na orientación a obxectos. Unha práctica de deseño centrada no uso de principios e patróns de deseño. A nota de prácticas repártese equitativamente entre os dous boletíns e a práctica de deseño As prácticas son obrigatorias e é necesario obter unha nota mínima de 4 para poder facer media cos outros elementos avaliábeis. En caso de non chegar á nota mínima implicará que non se poida obter máis dun 4,5 na nota final da materia Un alumno considerárase presentado á materia se presenta as prácticas da mesma (independentemente de se se presenta ou non á proba obxectiva)	35
Seminario	Evaluación continua de actividades propostas ao longo do curso	5
Proba obxectiva	Proba escrita realizada ao final do curso sobre contidos teórico-prácticos. A proba obxectiva é obrigatoria para aprobar a materia e tamén é obrigatorio obter unha nota mínima de 4 para poder facer media cos outros elementos avaliábeis. En caso de non chegar á nota mínima implicará que non se poida obter máis dun 4,5 na nota final da materia	60

Observacións avaliación

Aspectos a ter en conta para a avaliación de segunda oportunidade (Xullo):

A nota da proba obxectiva (exame teórico-práctico) gardarase se é igual ou maior que 5 sobre 10. A nota dos seminarios (TGRs) e das prácticas de laboratorio gardarase na segunda oportunidade. Existirá a posibilidade de mellorar a nota das prácticas de laboratorio tal e como se indicará na plataforma virtual. Na segunda oportunidade as porcentaxes son iguais e tamén rexen as normas dun mínimo dun 4 en prácticas e proba obxectiva. Considerárase que un alumno se presenta á segunda oportunidade se volve entregar as prácticas ou se presenta á proba obxectiva.

Fontes de información

Bibliografía 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
Bibliografía complementaria	- 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

Recomendacións

Materias que se recomenda ter cursado previamente

Proceso Software/614G01019
 Interfaces Home Máquina/614G01022
 Internet e sistemas distribuídos/614G01023
 Ensenar de Requisitos/614G01027

Materias que se recomenda cursar simultaneamente

Paradigmas de Programación/614G01014

Materias que continúan o temario

Programación I/614G01001
 Programación II/614G01006

Observacións



A materia asume que os estudantes saben programar e coñecen as estruturas de datos (Programación II) aínda que nunca utilizaron unha linguaxe orientada a obxectos. Ao principio do curso, e segundo vanse introducindo os conceptos propios da orientación a obxectos, os alumnos familiarízanse cos aspectos básicos da linguaxe de programación Java.

(*)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