UNIVERSIDADE DA CORUÑA

Teaching Guide				
Identifying Data				2013/14
Subject (*)	Deseño de Sistemas de Información		Code	614111403
Study programme	Enxeñeiro en Informática			
Descriptors				
Cycle	Period	Year	Type	Credits
First and Second Cycle	2nd four-month period	Fourth	Troncal	6
Language	Spanish			
Prerequisites				
Department	Computación			
Coordinador	Castro Souto, Laura Milagros	E-mail	laura.milagros.castro.souto@udc.es	
Lecturers	Castro Souto, Laura Milagros	E-mail	laura.milagros.castro.souto@udc.es	
Web	campusvirtual.udc.es			
General description	Deseño, propiedades e mantemento do software.			

Study programme competences	
Code	Study programme competences
A1	Aprender de maneira autónoma novos coñecementos e técnicas avanzadas axeitadas para a investigación, o deseño e o desenvolvemento de sistemas e servizos informáticos.
A3	Concibir e planificar o desenvolvemento de aplicacións informáticas complexas ou con requisitos especiais.
A6	Avaliar, definir, seleccionar e auditar plataformas hardware e software para a execución e desenvolvemento de aplicacións e servizos informáticos.
A7	Saber especificar, deseñar e implementar un sistema de información, empregando bases de datos.
A9	Dirixir equipos de traballo ligados ao deseño de produtos, procesos, servizos informáticos e outras actividades profesionais.
A10	Saber especificar, deseñar e implementar unha política de seguridade no sistema.
B1	Aprender a aprender.
B2	Resolver problemas de forma efectiva.
B3	Aplicar un pensamento crítico, lóxico e creativo.
B4	Aprendizaxe autónoma.
B5	Traballar de forma colaborativa.
B7	Comunicarse de maneira efectiva en calquera contorno de traballo.
B8	Traballar en equipos de carácter interdisciplinar.
B9	Capacidade para tomar decisións.
B10	Capacidade de xestión da informática (captación e análises da información).
B11	Razoamento crítico.
B12	Capacidade para a análise e a síntese.
B13	Capacidade de comunicación.
B14	Coñecemento de idiomas.
B15	Motivación pola calidade.
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.
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.

Learning outcomes	
Subject competencies (Learning outcomes)	Study programme competences

Knowledge about concepts and techniques of Software Engineering.	A3 A7 A9	B2 B9 B12 B15	C3 C6 C7 C8
Use of design and implementation tools.	A1 A3 A6 A7 A10	B1 B2 B3 B4 B5 B7 B8 B9 B10 B11 B14 B15	C3 C6 C7 C8
Expert use of modeling languages and design patterns.	A3 A7	B2 B3 B7 B9 B10 B11 B12 B15	C3 C6 C7 C8
Knowledge, identification and understanding of the typical situations where design problems arise.	A1 A3 A6 A7	B1 B2 B3 B4 B5 B7 B8 B9 B10 B11 B12 B13	C3 C6 C7 C8
Identification of success experiences.	A1 A3 A6 A7 A9	B1 B3 B4 B7 B9 B10 B11 B12 B13 B15	C3 C6 C7 C8

Contents	
Topic	**Sub-topic**

Introduction to ISD	a) Definition and objectives b) Design as a phase in the development process c) Basic principles of design d) Introduction to software architectures d) The UML unified modeling language f) The 4+1 model of architecture view
Design strategies	a) Basic concepts b) Structured design c) Object-oriented design
Software architectures	a) Basic concepts b) The Model-View-Controller pattern c) Layered design
Introduction to design patterns	a) Concept of design pattern b) Compositionality (Composite)
Responsibility assignment and delegation	a) Responsibility assignment (Proxy, Decorator) b) Delegation (Decorator, Chain of Responsibility)
Low coupling, instantiation, high cohesion, and subsystem division	a) Low coupling and instantiation (Prototype, Trader, Singleton) b) High cohesion and subsystem division (Facade)
Modeling dynamic behaviour	a) Modeling dynamic behaviour (State)
Reuse	a) Reuse (Strategy, Iterator)
Dependencies among objects	a) Dependencies among objects (Observer)
Design refactoring and preparation for change	a) Design refactoring (Factory Method, Template Method) b) Preparation for change (Visitor)
Encapsulation and abstraction	a) Encapsulation and abstraction (Comand, Memento)
Interface adaptation and complexity handling	a) Interface adaptation (Adapter, Bridge, Abstract Factory) b) Complexity handling (Mediator)
User interfaces	a) Basic concepts b) User interfaces on the web c) Dependencies among objects, encapsulation, abstraction, delegation, and subsystem division
Persistence and concurrency patterns	a) Introduction

Planning			
Methodologies / tests	Ordinary class hours	Student?s personal work hours	Total hours
Guest lecture / keynote speech	21	21	42
Seminar	10	15	25
Laboratory practice	15	15	30
Objective test	5	0	5
Supervised projects	0	24	24
Personalized attention	24	0	24
(*)The information in the planning table is for guidance only and does not take into account the heterogeneity of the students.			

Methodologies	
Methodologies	Description
Guest lecture / keynote speech	Expositive sessions to present theoretical aspects using different materials: board, slides, electronic notes and resources provided by the teacher using the virtual campus.
Seminar	Practical seminars in which acquired concepts are developed further, with a practical example as baseline. The teacher conducts the session, involving the students actively.

Laboratory practice	Practical sessions designed by the teacher on the grounds of the concepts that the students are presented with. Students develop practical exercises in small groups, preferably in pairs.
Objective test	Written test to assess student's knowledge acquisition. Students must develop both their knowledge of theoretical aspects, to be demonstrated by answering questions, and their knowledge of practical aspects, to be demonstrated by solving practical exercises, similar to those they have been previously addressing during practical sessions, seminars, and personal tasks.
Supervised projects	Personal tasks proposed by the teacher and developed by the students in an individual fashion, or in group. This tasks can be evaluated via test exams or individually during tutoring hours.

Personalized attention	
Methodologies	**Description**
Laboratory practice Objective test Supervised projects	Personalised attention to students involves not only classical tutor hours, either in face-to-face or virtual meetings, for discussion of questions, but also the following: - Monitoring of the work related to the proposed personal tasks assigned by the teacher. - Evaluation of the results of the personal work, tasks, and exercises performed by the students. - Discussion to solve the challenges encountered by the students in addressing both theoretical and practical tasks and exercises.

Assessment		
Methodologies	**Description**	**Qualification**
Laboratory practice	Evaluation of the practical exercises in a continuous manner thorough the course in a final defence. Despite these exercises being solved in groups, there are two components in this evaluation: - Evaluation of the group work, in which coordination and collaboration is evaluated. - Personal evaluation, where contribution of each individual is evaluated. Among the aspects to be evaluated, we find: - Rigorousness in achieving the objectives of each exercise, applying the techniques proposed in the subject. - Comprehension of the concepts involved in the exercise. - Originality in the solutions. - Responsibility in finishing the exercises in due time and according to the given instructions, as well as use of given material.	30
Objective test	Written test with three different parts: short questions, short design-and-implement exercises, and design problem with special interest in motivation of design decisions.	60
Supervised projects	The following aspects will be taken into account: - Knowledge of the contents of the proposed tasks. - Knowledge of the theoretical and practical aspects of the subject. - Participation and/or suitable tracking in the development of the subject.	10
Others		

Assessment comments

Sources of information	
Basic	- Gamma, E.; Helm, R.; Johnson, R. y Vlissides J. (1996). Design Patterns: Elements of Reusable Object-oriented Software. Addison Wesley - Arnold K., Gosling J. y Holmes D. (2005). The Java Programming Language. Prentice Hall - 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
Complementary	- Grand M. (2002). Patterns in Java. Volumen 1. John Wiley & Sons - 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

Recommendations
Subjects that it is recommended to have taken before
Integración de Sistemas/614111503
Subjects that are recommended to be taken simultaneously
Análise e Deseño Orientado a Obxectos/614111602 Interfaces co Usuario/614111624 Programación Orientada a Obxectos/614111636
Subjects that continue the syllabus
Análise de Sistemas Informáticos/614111402
Other comments

(*)The teaching guide is the document in which the URV publishes the information about all its courses. It is a public document and cannot be modified. Only in exceptional cases can it be revised by the competent agent or duly revised so that it is in line with current legislation.