

BASIC DETAILS:

Subject:	INGENIERÍA DEL SOFTWARE		
Id.:	30063		
Programme:	GRADUADO EN INGENIERÍA INFORMÁTICA. PLAN 2008 (BOE 15/12/2008)		
Module:	INGENIERIA DEL SOFTWARE		
Subject type:	OBLIGATORIA		
Year:	3	Teaching period:	Primer Cuatrimestre
Credits:	6	Total hours:	150
Classroom activities:	67	Individual study:	83
Main teaching language:	Inglés	Secondary teaching language:	Castellano
Lecturer:		Email:	

PRESENTATION:

This subject addresses the following contents to be applied during the development of software: methodologies, agile methods, basics for software design, and testing strategies.

PROFESSIONAL COMPETENCES ACQUIRED IN THE SUBJECT:

General programme competences	G01	Leadership capacity to be able to influence a group so they achieve some specific objectives collectively and efficiently.
	G02	Innovative capacity to propose and find new and efficient ways to undertake any task and/ or function within the professional environment - highly motivated by quality.
	G03	Capacity to work in multidisciplinary teams to achieve common objectives, placing group interests before personal ones.
	G04	Capacity to always commit to working responsibly - creating a strong sense of duty and fulfilment of obligations.
	G09	Capacity to make decisions impartially and rationally.
Specific programme competences	E07	Capacity to work effectively in project teams, where appropriate assuming executive responsibilities, and consider the human, technological and financial sides.
	E08	Capacity to communicate productively with clients, users and colleagues both orally and in writing, so as to pass on ideas, solve conflicts and achieve agreements.
	E12	Capacity to manage complexity through abstraction, modelling, 'best practices', patterns, standards and the use of the appropriate tools.
	E16	Capacity to understand an application demesne so as to be able to develop suitable IT applications.
	E17	Capacity to identify and analyse user needs with the intention of designing effective, usable IT solutions which can be incorporated into the user's operating environment.
	E18	Capacity to identify and define the requirements to be satisfied by IT systems to cover the stated needs of organisations or individuals.
	E19	Capacity to design and define the architecture of IT systems (software, hardware and communications) under the requirements agreed upon by the parties involved.
	E20	Capacity to undertake the detailed design of the components of a project (procedures, user interface, equipment characteristics, communications system parameters, etc.).
	E21	Capacity to perform tests that verify the validity of the project (functional, data integrity, performance of the computer applications, equipment, communications, etc.).
	E27	Capacity to write and maintain descriptive documentation of the origin, production and operability of IT systems.
Learning outcomes	R01	Analyse user specifications.
	R02	Model demesne problems.
	R03	Synthesise solution demesne analysis models.
	R04	Design solutions to problems.
	R05	Master programming.
	R06	Master methodologies for organised software construction.
	R07	Understand and interpret descriptive software process documents.
	R08	Interact in English in a work situation.
	R09	Pursue research tasks.
	R10	Productively work in a team.
	R11	Generate technical documentation.

PRE-REQUISITES:

SUBJECT PROGRAMME:

Subject contents:

1 - Introduction to Software Engineering
1.1 - Introduction
1.2 - Motivation
2 - Modeling Techniques
2.1 - Analysis
2.2 - Design
2.3 - Case study
3 - Software Process
3.1 - Objectives
3.2 - The Software Process
3.3 - The Software Lifecycle
3.4 - Methodologies
3.5 - Case study
4 - Implementation and Software Testing
4.1 - Basic foundations
4.2 - Programming principles and guidelines
5 - Design and redesign
5.1 - Interface design patterns
5.2 - Design patterns
5.3 - Refactorings

Subject planning could be modified due unforeseen circumstances (group performance, availability of resources, changes to academic calendar etc.) and should not, therefore, be considered to be definitive.

TEACHING AND LEARNING METHODOLOGIES AND ACTIVITIES:

Teaching and learning methodologies and activities applied:

This course will use the following methodologies in order to give the students the best opportunity to develop their competences: lectures, practical cases, exercises and coursework presentations.

Participation in class will be accounted in the final score. All readings, practices and works will be announced using the Online University Platform (pdu.usj.es).

Student work load:

Teaching mode	Teaching methods	Estimated hours
Classroom activities	Master classes	25
	Other theory activities	4
	Practical exercises	9
	Practical work, exercises, problem-solving etc.	6
	Debates	3
	Coursework presentations	4
	Laboratory practice	10
	Other practical activities	2
	Assessment activities	4
Individual study	Tutorials	4
	Individual study	27
	Individual coursework preparation	17

	Group coursework preparation	17
	Research work	4
	Compulsory reading	6
	Recommended reading	4
	Other individual study activities	4
	Total hours:	150

ASSESSMENT SCHEME:

Calculation of final mark:

Written tests:	60 %
Individual coursework:	15 %
Group coursework:	20 %
Participation:	5 %
TOTAL	100 %

*Las observaciones específicas sobre el sistema de evaluación serán comunicadas por escrito a los alumnos al inicio de la materia.

BIBLIOGRAPHY AND DOCUMENTATION:

Basic bibliography:

Sommerville, Ian (2004). Software Engineering, 7th Ed., Pearson.
Pressman, Roger (2005). Software Engineering. A Practitioners Approach, 6th Ed., McGraw-Hill.

Recommended bibliography:

Bjørner, Dines (2006). Software Engineering 3. Domains, Requirements and Software Design, Springer.
Shoval, Peretz (2007). Functional and Object Oriented Analysis and Design: an Integrated Methodology, Idea Group Publishing.
Jalote, Pankaj (2005). An Integrated Approach to Software Engineering, Springer.
McConnell, Steve (2003). Professional Software Development, Addison Wesley
Kroll, Per (2006). Agility and Discipline Made Easy: Practices from OpenUP and RUP, Addison Wesley.
Sangwan, Raghvinder et al (2007). Global Software Development Handbook, Auerbach Publications.
Tomayko, James et al (2004). Human Aspects of Software Development, Charles River Media.
Peckham, Joan (ed.) (2003). Practicing Software Engineering in the 21st Century, IRM Press.
Aurum, Aybuke et al (2005). Engineering and Managing Software Requirements, Springer.
Gunderloy, Mike (2004). Coder to Developer: Tools and Strategies for Delivering Your Software, Sybex
Booch, Grady et al (2007). Object-Oriented Analysis and Design with Applications, 3th Ed., Addison-Wesley.
Pidd, Michael (ed.) (2004). Systems Modelling. Theory and Practice, John Wiley.

Recommended websites:

Center for Systems and Software Engineering: The aim of this site is to work towards evolving and unifying theories and practices of systems and software Engineering.	http://csse.usc.edu/csse/
IEEE Transactions on Software Engineering: Technical articles and news about Software Engineering issues	https://www.computer.org/csdl/trans/ts/index.html
The Podcast for Professional Software Developers: Here you can download audio episodes relating experiences of software engineers	http://www.se-radio.net/

* Guía Docente sujeta a modificaciones