

BASIC DETAILS:

Subject:	INGENIERÍA DEL SOFTWARE		
Id.:	31369		
Programme:	GRADUADO EN DISEÑO Y DESARROLLO DE VIDEOJUEGOS. 2013 (BOE 28/03/2014)		
Module:	INFORMÁTICA		
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:	PEREZ PEREZ, MARIA FRANCISCA (T)	Email:	mfperes@usj.es

PRESENTATION:

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

PROFESSIONAL COMPETENCES ACQUIRED IN THE SUBJECT:

General programme competences	G01	Ability to use learning strategies independently for use in the continuous improvement of professional practice.
	G03	Ability to achieve common results through teamwork in a context of integration, cooperation and encouraging critical discussion.
	G05	Ability to communicate in Spanish and English for professional issues in oral and written form.
	G07	Ability to handle different complex knowledge models through a process of abstraction and its application to approach and solve problems.
Specific programme competences	E16	Ability to fully manage and plan software projects and handle suitable tools to do so.
	E19	Ability to recognise and apply the principles, methodologies and life cycle of software engineering.
	E30	Ability to design, develop, select and evaluate applications and systems, ensuring reliability, safety and quality, according to ethical principles and legislation and regulations.
Learning outcomes	R01	Design video games.
	R02	Properly implement agile methodologies in video game development.
	R03	Define video game development plans.

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, Aybüke 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/