BASIC DETAILS:

Subject:	TECNOLOGÍAS AVANZADAS
Id.:	30548
Programme:	GRADUADO EN INGENIERÍA INFORMÁTICA (SEMIPRESENCIAL). 2008 (BOE 15/12/2008)
Module:	APLICACIONES DISTRIBUIDAS
Subject type:	OBLIGATORIA

Year:	4	**Teaching period:**	Primer Cuatrimestre
Credits:	6	**Total hours:**	150
Classroom activities:	16	**Individual study:**	134
Main teaching language:	Inglés	**Secondary teaching language:**	Castellano
Lecturer:	NOVOA MINGUEZ, RAUL (T)	**Email:**	rnovoa@usj.es

PRESENTATION:

To move forward as a system architecture engineer, the student needs to gain experience in advanced web technologies. This subject deals with some Javascript technologies used often in Internet applications that large enterprise organizations require, and also provides knowledge that can be harnessed to prepare a certification as architect of NodeJS systems. API REST (Representational State Transfer) is designed to take advantage of existing protocols. While REST can be used over nearly any protocol, it usually takes advantage of HTTP when used for Web APIs Web Services.

Since many of us are now writing or maintaining RESTful microservices and these services/ APIs are exposed to the web and distributed over different networks, they are vulnerable to risks and security threats which affect the processes based on them. Hence, testing becomes necessary to ensure they perform correctly. To test these APIs, it's very important to automate REST API test cases instead of relying on manual testing.

PROFESSIONAL COMPETENCES ACQUIRED IN THE SUBJECT:

General programme competences	G04	Capacity to always commit to working responsibly - creating a strong sense of duty and fulfilment of obligations.
	G10	Critical and analytical capacity when assessing information, data and courses of action.
	G11	Ability to get on in a multicultural or international environment, interacting with people of different nationalities, languages and cultures.
	G12	Capacity to undertake professional activities with integrity, respecting social, organisational and ethical norms.
	G13	Capacity to use individual learning strategies aimed at continuous improvement in professional life and to begin further studies independently.
	G14	Capacity for abstraction to handle various complex knowledge models and apply them to examining and solving problems.
	G15	Capacity to structure reality by means of linking objects, situations and concepts through logical mathematical reasoning.
Specific programme competences	E06	Capacity to apply quality assurance processes to processes and products.
	E07	Capacity to work effectively in project teams, where appropriate assuming executive responsibilities, and consider the human, technological and financial sides.
	E09	Capacity to maintain professional competences through independent learning and continuous improvement.
	E12	Capacity to manage complexity through abstraction, modelling, 'best practices', patterns, standards and the use of the appropriate tools.
	E15	Capacity to understand and go along with the strategic objectives of the company where your are pursuing your professional career.
	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.
	E21	Capacity to perform tests that verify the validity of the project (functional, data integrity, performance of the computer applications, equipment, communications, etc.).
	E22	Capacity to undertake implementation tasks which require a high degree of technical awareness in different spheres (programming, configuration of hardware and communications equipment, etc.).
	E23	Capacity to design and implement security policies in order to preserve the integrity of the operational environment.

	E24	Capacity to draw up and develop effective project plans for systems based on information and communication technologies.
	E25	Capacity to analyse viability, design development plans, estimate resources, run and oversee the execution of software-intensive engineering projects.
Learning outcomes	R01	Understand the object-oriented focus through distant objects.
	R02	Use basic Servlet structure.
	R03	Develop JSP applications which perform Web services.
	R04	Apply AJAX technology.
	R05	Manage the Java Enterprise Edition platform to develop large applications.

PRE-REQUISITES:

The student should have passed the subjects: Oriented Object Programming I and II

SUBJECT PROGRAMME:

Subject contents:

1 - Basic resources
2 - Overview
2.1 - History
2.2 - Introduction to API design
2.3 - Maturity model
3 - Resources and representation
4 - The design procedure
5 - Status codes &headers
6 - Authentication &authorization
7 - Testing

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:

Day by day the student must compose his portfolio of the subject. It include lectures summary and solved exercises. According to the previous paragraph calendar, several exercises will be proposed and its solutions will be discussed some time later. The solutions of everyday exercises will be included in the portfolio and sent to the PDU when they are required. The lecturer will upload his lectures notes on the PDU (Plataforma Docente Universitaria). Self-learning: after class, students have to finish the task has left uncompleted in class or do remain ones. All task must be uploaded to PDU in the time ordered. Students are recommended to consult the lecturer by e-mail.

At the end of the term students must pass a test. It could be composed of some questions, exercises, or oral presentation.

Student work load:

Teaching mode	Teaching methods	Estimated hours
Classroom activities	Master classes	14
	Assessment activities	2
Individual study	Tutorials	11
	Individual study	55
	Individual coursework preparation	44
	Portfolio	24
	Total hours:	150

ASSESSMENT SCHEME:

Calculation of final mark:

Written tests:	40	%
Individual coursework:	20	%
Group coursework:	15	%
Final exam:	25	%
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:

RESTful Web APIs by Leonard Richardson and Mike Amundsen (O'Reilly). Copyright 2013 Leonard Richardson and Amundsen.com, Inc., and Sam Ruby. 978-1-449-35806-8.

Recommended bibliography:

RESTful Web Clients byMike Amundsen (O'Reilly). Copyright 2017 Mike Amundsen.com, Inc., and Sam Ruby. 978-1-491-92190-6

Recommended websites:

NodeJS	http://nodejs.org
Docker	https://www.docker.com
PostgreSQL	https://www.postgresql.org
Kitematic	https://kitematic.com
pgAdmin	https://www.pgadmin.org
Visual studio code	https://code.visualstudio.com
Postman	https://www.getpostman.com