BASIC DETAILS:

Subject:	TECNOLOGÍAS AVANZADAS		
Id.:	31847		
Programme:	DOBLE GRADO EN INGENIERÍA INFORMÁTICA Y DISEÑO Y DESARROLLO DE VIDEOJUEGOS		
Module:	APLICACIONES DISTRIBUIDAS		
Subject type:	OBLIGATORIA		
Year:	4	**Teaching period:**	Primer Cuatrimestre
Credits:	6	**Total hours:**	150
Classroom activities:	61	**Individual study:**	89
Main teaching language:	Inglés	**Secondary teaching language:**	Castellano
Lecturer:		**Email:**	

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.

Continuous integration / delivery with bitbucket and azure will be used during the subject.

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.
	G05	Capacity to adapt to different environments while being positive and optimistic, orienting your behaviour towards the achievement of goals.
	G06	Capacity to analyse and find a solution to complex problems or unforeseen situations which may arise while working in any type of socio-economic organisation.
	G07	Capacity to work flexibly and with versatility to adapt to the needs and requirements of the work situation.
	G08	Ability to communicate effectively about different matters in a variety of professional situations and with the different media available.
	G09	Capacity to make decisions impartially and rationally.
	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	E01	Capacity to understand the engineering profession and commitment to serve society under the corresponding professional code of conduct.
	E02	Capacity to apply the intrinsic engineering principles based on mathematics and a combination of scientific disciplines.
	E03	Capacity to recognise the technical principles and apply the appropriate practical methods satisfactorily to analyse and solve engineering problems.
	E04	Capacity to maintain an open mind to innovation and creativity within the framework of the engineering profession.
	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.
	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.
	E09	Capacity to maintain professional competences through independent learning and continuous improvement.
	E11	Capacity to remain up-to-date in the technological and business worlds in the area of information and communication technologies.
	E12	Capacity to manage complexity through abstraction, modelling, 'best practices', patterns, standards and the use of the appropriate tools.
	E13	Capacity to identify, assess and use current and emerging technologies, considering how they apply in terms of individual or organisational needs.
	E15	Capacity to understand and go along with the strategic objectives of the company where your are pursuing your professional career.
	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.).
	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.

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
8 - Continuous integration
8.1 - Git (bitbucket)
8.2 - Docker
8.3 - Azure
9 - Final project

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	13
	Other theory activities	2
	Practical exercises	3
	Practical work, exercises, problem-solving etc.	24
	Coursework presentations	15
	Assessment activities	4
Individual study	Tutorials	5
	Individual study	23
	Individual coursework preparation	11
	Group cousework preparation	11
	Project work	19
	Portfolio	20
	Total hours:	150

ASSESSMENT SCHEME:

Calculation of final mark:

Written tests:	10	%
Individual coursework:	40	%
Group coursework:	15	%
Final exam:	20	%
Attendance and participation:	15	%
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.
Irresistable APIs. Designing web APIs that developers will love. Copyright 2016 Kirsten L. Hunter. ISBN 9781617292552
REST in Practice: Hypermedia and Systems Architecture. Copyright 2010 O'Reilly and Associates. ISBN 978-0596805821
Build APIs you won't hate. Copyright 2016 Philip J. Sturgeon. ISBN 978-0692232699

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

* Guía Docente sujeta a modificaciones