

## BASIC DETAILS:

<b>Subject:</b>	PROGRAMACION CONCURRENTE Y DISTRIBUIDA		
<b>Id.:</b>	30066		
<b>Programme:</b>	GRADUADO EN INGENIERÍA INFORMÁTICA. PLAN 2008 (BOE 15/12/2008)		
<b>Module:</b>	PROGRAMACION Y DESARROLLO		
<b>Subject type:</b>	OPTATIVA		
<b>Year:</b>	3	<b>Teaching period:</b>	Segundo Cuatrimestre
<b>Credits:</b>	3	<b>Total hours:</b>	75
<b>Classroom activities:</b>	34	<b>Individual study:</b>	41
<b>Main teaching language:</b>	Inglés	<b>Secondary teaching language:</b>	Castellano
<b>Lecturer:</b>		<b>Email:</b>	

## PRESENTATION:

This subject deals with the techniques and tools for communication and synchronization of processes in a shared resources environment and in a distributed environment. It studies concurrent programming models.

We'll use NodeJS as programming language for learning concurrent programming techniques.

## PROFESSIONAL COMPETENCES ACQUIRED IN THE SUBJECT:

<b>General programme competences</b>	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.
	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.
	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.
	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.
<b>Specific programme competences</b>	E03	Capacity to recognise the technical principles and apply the appropriate practical methods satisfactorily to analyse and solve engineering problems.
	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.
	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.
	E27	Capacity to write and maintain descriptive documentation of the origin, production and operability of IT systems.
<b>Learning outcomes</b>	R01	Understand the principles and methodologies of concurrent and distributed programming, especially the concepts of synchronisation and communication between processes.
	R02	Know the main difficulties in producing concurrent and distributed programs
	R03	Ability to model concurrent and distributed processes in real time through Petri Nets
	R04	Understand the high performance in supercomputing

## PRE-REQUISITES:

OOP, networks, communications and operating systems knowledge is mandatory.

## SUBJECT PROGRAMME:

Observations:

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.

### Subject contents:

<b>1 - Introduction to concurrent and distributed programming</b>
<b>2 - NodeJS concurrent model</b>
<b>3 - Async/Await &amp; Promises</b>
<b>4 - Concurrency with workers</b>
<b>5 - Concurrency and functional programming</b>
<b>6 - Sockets</b>
<b>7 - Mutex, semaphores, locks</b>
<b>8 - High performance &amp; scalable concurrent applications</b>

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.

#### Student work load:

Teaching mode	Teaching methods	Estimated hours
<b>Classroom activities</b>	Master classes	20
	Practical exercises	6
	Practical work, exercises, problem-solving etc.	6
	Assessment activities	2
<b>Individual study</b>	Individual study	15
	Individual coursework preparation	16
	Group coursework preparation	5
	Research work	5
<b>Total hours:</b>		<b>75</b>

### ASSESSMENT SCHEME:

#### Calculation of final mark:

Written tests:	20	%
Individual coursework:	50	%
Group coursework:	20	%
Attendance and participation:	10	%
<b>TOTAL</b>	<b>100</b>	<b>%</b>

\*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:**

NodeJS design patterns. Copyright 2016 Mario Casciaro, Luciano Mammino. ISBN 978-1-78588-558-7

**Recommended bibliography:**

**Recommended websites:**

NodeJS	<a href="http://nodejs.org">http://nodejs.org</a>
Visual studio code	<a href="https://code.visualstudio.com">https://code.visualstudio.com</a>
Socket.io	<a href="https://socket.io">https://socket.io</a>
RxJS	<a href="https://rxjs-dev.firebaseapp.com">https://rxjs-dev.firebaseapp.com</a>

\* Guía Docente sujeta a modificaciones