

## BASIC DETAILS:

<b>Subject:</b>	COMPUTACIÓN DE ALTO RENDIMIENTO		
<b>Id.:</b>	33715		
<b>Programme:</b>	DOBLE GRADO EN FARMACIA Y BIOINFORMÁTICA. PLAN 2019		
<b>Module:</b>	INFORMÁTICA		
<b>Subject type:</b>	OBLIGATORIA		
<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>	CARRO FERNANDEZ, JESUS (T)	<b>Email:</b>	jcarro@usj.es

## PRESENTATION:

In this subject, the students will learn complex computational systems and how they work. When the clock frequency of a processor can not be increased it is necessary to look for other ways to increase the computational capacity. Through this subject, the students will learn how this can be solved using parallelization in the same computer, with more than one computer or using other parts of the computer as the GPU.

## PROFESSIONAL COMPETENCES ACQUIRED IN THE SUBJECT:

<b>General programme competences</b>	G01	Use learning strategies autonomously for their application in the continuous improvement of professional practice.	
	G02	Perform the analysis and synthesis of problems of their professional activity and apply them in similar environments.	
	G03	Cooperate to achieve common results through teamwork in a context of integration, collaboration and empowerment of critical discussion.	
	G05	Communicate professional topics in Spanish and / or English both orally and in writing.	
	G06	Solve complex or unforeseen problems that arise during the professional activity within any type of organisation and adapt to the needs and demands of their professional environment.	
	G07	Choose between different complex models of knowledge to solve problems.	
	G09	Apply information and communication technologies in the professional field.	
	G10	Apply creativity, independence of thought, self-criticism and autonomy in the professional practice.	
	<b>Specific programme competences</b>	E02	Develop the use and programming of computers, databases and computer programs and their application in bioinformatics.
		E03	Apply the fundamental concepts of mathematics, logic, algorithmics and computational complexity to solve problems specific to bioinformatics.
E04		Program applications in a robust, correct, and efficient way, choosing the paradigm and the most appropriate programming languages, applying knowledge about basic algorithmic procedures and using the most appropriate types and data structures.	
E05		Implement well-founded applications, previously designed and analysed, in the characteristics of the databases.	
E07		Apply the principles, methodologies and life cycles of software engineering to the development of a project in the field of bioinformatics.	
E08		Evaluate applications and computer systems, previously designed, developed and selected, ensuring their reliability and quality, in accordance with ethical principles and current legislation and regulations.	
E09		Develop and maintain descriptive documentation of the genesis, production and operation of computer systems.	
E10		Design and deploy the architecture of IT systems through the definition of software, hardware and the necessary communications according to some requirements.	
E11		Apply the principles and techniques of concurrent or parallel computing for the creation and simulation of bio-inspired processes.	

## PRE-REQUISITES:

## SUBJECT PROGRAMME:

**Subject contents:**

<b>1 - Introduction</b>
1.1 - The need for computing
1.2 - Moore's Law
1.3 - Pipelining
1.4 - Parallel programming
<b>2 - Multiprocessors</b>
2.1 - Introduction
2.2 - Shared memory
2.3 - Multithreading hardware
2.4 - Multicore processors
2.5 - OpenMP
<b>3 - Grid Computing</b>
3.1 - Introduction
3.2 - Classification
3.3 - Cluster Architecture
3.4 - MPI
3.5 - Volunteer computing
<b>4 - GPGPU</b>
4.1 - Introduction to heterogeneous computing
4.2 - OpenCL
4.3 - CUDA

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:**

Classes are conducted exclusively through English using CLIL. Students are expected to participate in class and in all class-related activities in English. Classes are practical and communicative. Many class activities will be conducted in pairs and groups in the format of information gaps, debates, project preparation, problem solving, simulations, presentations etc.

This methodology enables maximum student participation and talking time in class. It also encourages cooperate learning and meaningful interaction between students and the development of professional competences.

Students are expected to complete all independent study tasks, which will be uploaded on the PDU.

**Theory Sessions:** Lectures will be used to explain the basis of the different chapters. Wherever possible, explanations will be accompanied by images, text or sounds to be used as practical examples and discussion topics. During the sessions, the lecturer will propose activities or to look for information out of the class and he will resolve doubts.

**Practical Sessions:** There are individual labs. During practice, students will use problem-based learning methodological strategy

**Problem based learning:** At the end of the subject, the students will work in a project of HPC for Bioinformatics. They will use all the tools learned through the subject to reduce the amount of time that is necessary to complete the task.

**Student work load:**

Teaching mode	Teaching methods	Estimated hours
---------------	------------------	-----------------

<b>Classroom activities</b>	Master classes	11
	Practical work, exercises, problem-solving etc.	5
	Laboratory practice	2
	Assessment activities	2
	Video class/Webinar/Videolesson/Podcast	2
	Collaborative activities	2
	Individual activities	6
<b>Individual study</b>	Online tests	4
	Tutorials	3
	Individual study	12
	Individual coursework preparation	21
	Recommended reading	5
<b>Total hours:</b>		<b>75</b>

### ASSESSMENT SCHEME:

#### Calculation of final mark:

Written tests:	30	%
Online tests:	10	%
Laboratories:	35	%
Final project:	25	%
<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:

HENNESSY, John L. and PATTERSON, David A. Computer Architecture: A Quantitative Approach. Morgan Kaufmann, 3a Edición, 2003.

SANDERS, Jason and Kandrot, Edward. CUDA by example: an introduction to general-purpose GPU programming. Addison-Wesley, 2011.

#### Recommended bibliography:

BUYYA, R. High Performance Cluster Computing: Architectures and Systems. Upper Saddle River, NJ. Prentice Hall, 1999

BUYYA, R. High Performance Cluster Computing: Programming and Applications. Upper Saddle River, NJ. Prentice Hall, 1999

HWANG, K. Advanced Computer Architecture. New York: McGraw-Hill, 1993.

STALLINGS, William. Organización y arquitectura de computadores. Prentice Hall, 2006.

#### Recommended websites:

BOINC	<a href="http://boinc.berkeley.edu/">http://boinc.berkeley.edu/</a>
CUDA	<a href="https://developer.nvidia.com/cuda-zone">https://developer.nvidia.com/cuda-zone</a>
HTCondor	<a href="https://research.cs.wisc.edu/htcondor/">https://research.cs.wisc.edu/htcondor/</a>