1. OVERVIEW

Subject area	Advanced Scientific Computing
Degree	Bachelor's Degree in Physics [Computing and Data Analysis Speciality]
School/Faculty	School of Architecture, Engineering and Design
Year	Third
ECTS	6
Type	Elective [compulsory for the Computing and Data Analysis Speciality]
Language(s)	Spanish
Delivery mode	On campus
Semester	S5

2. INTRODUCTION

This subject area covers a series of basic elements of computer science and programming, including the implementation of data structures, the study of algorithms (with a special focus on efficiency) and an introduction to concurrent and distributed programming. The main aim is to provide students with a solid foundation in these fields, facilitating subsequent academic and professional development, and to improve their data handling and practical problem-solving techniques.

3. SKILLS AND LEARNING OUTCOMES

Key skills (CB, by the acronym in Spanish):

- CB2: Students can apply their knowledge to their work or vocation in a professional manner and possess the skills which are usually evident through the forming and defending of opinions and resolving problems within their study area.
- CB5: Students have developed the learning skills necessary to undertake further study in a much more independent manner.

General skills (CG, by the acronym in Spanish):

- CG2: Ability to plan and perform independent work when managing projects associated with different areas of physics.
- CG3: To understand and express oneself in a language of science other than Spanish in a professional setting.

- CG4: To convey knowledge, procedures, results and scientific ideas in the field of physics, both orally and in writing.
- CG5: To understand diverse phenomena that, despite being physically different, share certain similarities, allowing known solutions to be applied to new problems.

Transversal skills (CT, by the acronym in Spanish):

- CT2: Independent learning: A range of skills in order to choose research, analysis, evaluation and information management strategies from different sources, as well as to learn and put into practice what has been learnt independently.
- CT3: Teamwork: Ability to integrate and collaborate actively with other people, areas and/or organisations to reach common goals.
- CT5: Problem solving: Be able to critically evaluate information, separate complex situations into their constituent parts, recognise patterns, and consider alternatives, different approaches and perspectives in order to find optimal solutions and negotiate efficiently.
- CT7: Leadership: To be able to direct, motivate and guide others, recognising their skills and abilities in order to effectively manage their development and common interests.

Specific skills (CE, by the acronym in Spanish):

- CE06: To understand key experimental models and to perform experiments independently, describing, analysing and critically assessing experimental data.
- CE07: To use the most suitable electronic instruments and IT tools to study physical problems and search for solutions.

Learning outcomes (RA, by the acronym in Spanish):

By achieving these learning outcomes, students will be able:

- RA1: To describe, design and use data types and structures to solve problems.
- RA2: To assess the computational complexity of a problem and describe algorithmic strategies that could solve the problem, developing and implementing solutions that provide the best response to the defined requirements.
- RA3: To list and apply the main principles and basic techniques of concurrent and distributed programming.

The following table shows how the skills developed in the subject area match up with the intended learning outcomes:

Skills	Learning outcomes
CB2, CB5, CG2, CG4, CG5, CT2, CT3, CT5, CE06, CE07	RA1
CB2, CB5, CG2, CG3, CG4, CG5, CT2, CT3, CT5, CT7, CE06, CE07	RA2

CB2, CB5, CG2, CG3, CG4, CG5, CT2, CT3, CT5, CT7, CE06, CE07	RA3

4. CONTENTS

- Data structures (linear, hierarchical, relational and functional).
- Advanced algorithm design techniques. Efficiency and complexity.
- Concurrent programming. Communication and synchronisation mechanisms.
- Introduction to distributed systems.

5. TEACHING/LEARNING METHODS

The types of teaching/learning methods are as follows:

1. Case studies: Discussion of real cases that allow for practical application of the acquired theoretical knowledge.
2. Collaborative learning: Students learn to collaborate with other people (classmates and professors) in order to find creative, comprehensive and constructive solutions to questions and problems that arise from the given case studies, using all relevant knowledge and material resources available.
3. Problem-based learning: Students are given problems and asked to solve them, working individually or in groups.
4. Lectures: Presentations by the professor with the necessary technological tools to maximise comprehension of the learning content.
5. Workshop-based learning: Students acquire knowledge through learning to use the tools and equipment needed in their profession. In other words, "learning by doing".
6. Guided academic activities: Individual and group work that is more independent, including information searches, written summaries, debates and the public defence of projects.

6. LEARNING ACTIVITIES

The types of learning activities, plus the amount of time spent on each activity, are as follows:

Learning activity	Number of hours
Lectures	22
Asynchronous lectures	4

Oral presentations of projects and debates	6
Report writing	21
Assessment	6
Practical activities (problems, written work, projects, workshops and/or lab work)	21
Group tutorials	16
Independent working	54
TOTAL	**150**

7. ASSESSMENT

The assessable activities, the assessment criteria for each of them and their weighting with regard to the total course grade are set out in the following table.

Assessable task (module)	Assessment criteria	Weighting
Final integrative test	Ability to analyse and solve theoretical/practical problems using the techniques learnt during the subject area.	30%
Midterm activity	Ability to analyse and solve problems using the techniques learnt up until this point.	20%
Collaborative sessions	• Ability to collaborate effectively with fellow students. • Ability to correctly apply the programming techniques studied in the lectures.	20%
Individual tasks	• Ability to analyse and solve problems using the techniques learnt up until this point.	10%
Group challenges	• Ability to analyse and solve more complex problems. • Ability to collaborate effectively with fellow students. • Ability to communicate effectively both orally and in writing.	10% reports + 10% defence oral = 20%

8. BIBLIOGRAPHY

The recommended bibliography is indicated below:

- Stroustrup, B., *The C++ Programming Language*. 4th Edition. Addison-Wesley Professional (2013).
- Bryant R. E. and O'Hallaron D. R., *Computer Systems: A Programmer's Perspective*. Second edition. Addison-Wesley (2010).
- Herlihy M. y Shavit N., *The Art of Multiprocessor Programming*. Morgan Kaufmann, Revised edición (2012).
- Williams A., *C++ Concurrency in Action*. Manning Publications, 2nd edition (2019).
- Cormen T. H., et al., *Introduction to Algorithms*, 3rd Edition. MIT Press (2009).
- Joyanes Aguilar L. et al., *Estructuras de datos en C++*. McGraw-Hill, 2007.