



Universitat de Lleida

DEGREE CURRICULUM  
**SPECIFICATION AND ANALYSIS  
OF INTERACTIVE SYSTEMS**

Coordination: SAYAGO BARRANTES, SERGIO

Academic year 2023-24

## Subject's general information

<b>Subject name</b>	SPECIFICATION AND ANALYSIS OF INTERACTIVE SYSTEMS			
<b>Code</b>	102382			
<b>Semester</b>	2nd Q(SEMESTER) CONTINUED EVALUATION			
<b>Typology</b>	Degree	Course	Character	Modality
	Bachelor's degree in Digital Interaction and Computing Techniques	2	COMPULSORY	Attendance-based
<b>Course number of credits (ECTS)</b>	6			
<b>Type of activity, credits, and groups</b>	Only examination			
<b>Coordination</b>	SAYAGO BARRANTES, SERGIO			
<b>Department</b>	COMPUTER ENGINEERING AND DIGITAL DESIGN			
<b>Teaching load distribution between lectures and independent student work</b>	<p>According to the academic framework of bachelor's degrees of the EPS:</p> <ul style="list-style-type: none"> <li>- 1 ECTS = 25 hours; 6 ECTS = 150 hours</li> <li>- 40% (60h) of in-class work and 60% (90h) of autonomous work</li> </ul> <p>The distribution of hours in this course is:</p> <ul style="list-style-type: none"> <li>- In-class work: theory (28h) + laboratories (28h) = 56 h + 4 hours of exams = 60h</li> <li>- Autonomous work: study (45h) + project (45h) = 90h</li> </ul>			
<b>Important information on data processing</b>	Consult <a href="#">this link</a> for more information.			
<b>Language</b>	Spanish (materials). Catalan & Spanish (in the classroom)			
<b>Distribution of credits</b>	See type of activity, credits and groups			

Teaching staff	E-mail addresses	Credits taught by teacher	Office and hour of attention
SAYAGO BARRANTES, SERGIO	sergio.sayago@udl.cat	0	Make an appointment via e-mail

## Subject's extra information

Software dominates the world and the development of professional software is a formal task. The Specification and Analysis of Interactive Systems (SPAIS) is a mandatory course of 6 ECTS. SPAIS is held during the second semester of the second year of the Bachelor's degree in Digital Interaction and Computing Techniques (GTIDIC).

GTIDIC aims to train qualified professionals in the computing field with a very practical side, giving special emphasis to the design and implementation of interactive applications. The graduates will acquire solid programming knowledge, focusing on mobile and web applications, Internet technologies, administration tools and security systems, and interface design and development.

SPAIS is designed to provide students with a guided and hands-on introduction to Software Engineering, given special emphasis to the design (specification and analysis) of interactive systems, so that students can integrate themselves in professional software engineering teams quickly. To achieve this objective, the student will carry out a project in which they will apply aspects related to Requirement Engineering, and other important elements of Software Engineering, such as domain analysis, UML (Unified Modelling Language), standards and quality in software, and management of projects.

SPAIS complements Object Oriented Programming and Application Platforms, providing students with a Software Engineering perspective, and is an introduction to Interactive Applications Design, where students will learn further aspects about design patterns.

Free software will be used in this course, such as online UML editors (Visual Paradigm) and IntelliJ IDEA

## Learning objectives

1. To know basic aspects of Software Engineering and development process models
2. To be able to design and create use cases
3. To gather and specify different types of requirements in a rigorous and professional way
4. To design the class diagram of an interactive system following the paradigm of object oriented design
5. To be able to use UML modelling tools
6. To understand software code as something that evolves over time
7. To be able to write unit tests
8. To understand the key tenets of object oriented design
9. To understand the concept of responsible design in object oriented design

## Competences

According to the table of competences of the GTIDIC (<https://ja.cat/zvyK4>):

### Basic competences

**CB3.** That students have the ability to gather and interpret relevant data (usually within their area of study) to make judgments that include a reflection on relevant social, scientific or ethical issues.

### Transversal competences

**CT3.** Acquire training in the use of new technologies and information and communication technologies

## General competences

**CG1.** Capacity to conceive, plan and develop projects in the field of ICT

**CG2.** Capacity to design, develop, evaluate and ensure the accessibility, ergonomics, usability and security of computer systems

**CG4.** Capacity to use software engineering methods in the development of interactive computer applications.

**CG9.** Capacity to analyze and synthesize

## Specific competences

**CE3.** Basic knowledge of the use and programming of computers, operating systems and databases, and their use in the development of interactive applications.

**CE6.** Capacity to design, develop, select and evaluate applications and computer systems, ensuring its reliability, security and quality.

**CE10.** Capacity to analyse, design, build and maintain safe and efficient applications, choosing the most suitable paradigm and programming languages.

**CE13.** Knowledge and application of the characteristics, functionalities and structure of the databases, that allow their suitable use, and the design and the analysis and implementation of interactive applications based on them.

**CE15.** Knowledge and application of the principles, methodologies and life cycles of software engineering

**CE16.** Capacity to design and evaluate person-computer interfaces that guarantee the usability of systems, services and computer applications

**CE25.** Being able to analyze, organize, label and visualize the structure that defines the interaction with digital content, through the application of information architecture methods, techniques and tools that facilitate accessibility

**CE26.** Knowing how to apply the principles and standards of accessibility and universal design of the main digital products and services to design experiences that guarantee equal opportunities among their users.

## Subject contents

- Key concepts of Software Engineering
- Requirements engineering (analysis and specification)
- Domain analysis
- Introduction to test-driven development
- Principles of software design

## Methodology

**This subject is offered this year in exam mode only and without scheduled classes**

Methodology	Theory	Laboratory	Independent student's work
Lectures	X		
Integrated project (in laboratories)		X	
Integrated project (at home)			X
Study			X

## Development plan

Week	Theory	Laboratories	Comments
1	Presentation T1.	P1.	
2	T2.	P1 (cont)	P1 ready. It can be submitted.
3	T3.	P2.	
4	T4.	P2 (cont.)	
5	T5.	P3.	P2 ready. It can be submitted.
6	T5	P3 (cont.)	
7	P3 (cont)	P3 (cont.)	P3 ready. It can be submitted.
8	EXAMS	EXAMS	
9	EASTER	P4 .Sprint I	
10	T6. P4 (cont)	P4 (cont)	
11	T7.	T7	P4 ready. It can be submitted.
12	DAY OFF	P5. Sprint II	
13	P5 (cont.)	P5 (cont)	P5 ready. It can be submitted.
14	P6. Sprint III	P6 (cont.)	
15	P6 (cont.)	P6 (cont.)	P6 ready. It can be submitted.
16	EXAMS + RETAKES		

## Evaluation

Parts and Activity	Weight	Mandatory	Minimum mark	Retake
Part I. First written exam	25%	YES	NO	YES
Part II. Second written exam	25%	YES	NO	YES
Part III. Labs	50%	YES	NO	YES

- Nota Final  $\geq 5$
- If Nota Final  $< 5$ , written exam (all contents of the subject)

### Alternative assessment

Students who have the approval to be assessed through alternative assessment (see requirements and procedure in the assessment regulations) must carry out the following activities:

Final written exam (grade  $\geq 5$ )

## Bibliography

[Applying UML and patterns : an introduction to object-oriented analysis and design and iterative development /](#)

[Craig Larman](#)

[Requirements engineering / Jeremy Dick, Elizabeth Hull, Ken Jackson](#)

[Software engineering / Ian Sommerville](#)

[Software quality assurance : from theory to implementation / Daniel Galin](#)