Self-evaluation report

Bachelor's degree in
Software Development and Testing

Universitat Oberta de Catalunya

TABLE OF CONTENTS

Glossary

ANECA - Agencia Nacional de Evaluación de la Calidad y Acreditación

AQU - Agència per a la Qualitat del Sistema Universitari de Catalunya

BTSD - Bachelor’s degree in Techniques for Software Application Development

CA - Continuous Assessment

CEFR - Common European Framework of Reference for Languages

CRUE - Conferencia de Rectores de las Universidades Españolas

ECTS - European Credit Transfer and Accumulation System

EQF - European Qualifications Framework

EQANIE - European Quality Assurance Network For Informatics Education

EX - Exam

FA - Final Assessment

FQ-EHEA - Framework for Qualifications in the European Higher Education Area

ICT - Information and Communications Technologies

ISCED - International Standard Classification of Education

NVAO - Nederlands Vlaamse Accreditatie Organisatie

OUNL - Open University of the Netherlands

QA - Quality Assurance

QPED - Quality-focused Programming EDucation

ST - Synthesis Test

UNED - Universidad Nacional de Educación a Distancia

1. General Overview

1.1. Introduction

1.1.1. Background context

In the last 50 years, Information and Communications Technologies (ICT) have had a profound impact on all aspects of our lives: society, economy, health, culture, etc. Nevertheless, the adoption of ICT is still an ongoing process with untapped potential, which has been called *digital transformation*.

Software is one of the pillars enabling this digital transformation by automating processes, providing services, and controlling and connecting devices. Currently, software lies at the core of almost all products, services, and industries. This trend was already predicted in 2011, in the famous essay "Software is eating the world" by software engineer and entrepreneur Marc Andreessen.

As a result, there has been an ever-growing demand for software development professionals, for instance, doubling in 2021. This rising demand has been recognised globally by members of the public and private sectors as a challenge for the success of digital transformation. For instance:

- In 2021, the European Commission adopted a "Path to the Digital Decade" strategy to achieve digital transformation goals in Europe by 2030. These goals, called "2030 Digital Compass", include a digital skills target of at least 20 million employed ICT specialists in the EU (with a balanced participation in terms of gender). Nevertheless, in 2019 55% of companies recruiting ICT specialists report difficulties in filling vacancies. In particular, the European Commission estimated that the current gap between demand and supply of ICT professionals in the EU reaches 500.000 professionals. Even among ICT experts, software developers are highlighted as positions with a “high magnitude shortage” of professionals.

- The US Bureau of Labor Statistics publishes an Occupational Outlook Handbook, where they predict a 22% growth in the number of jobs for “Software Developers, Quality Assurance Analysts and Testers” from 2020 to 2030. To provide context, this growth is considered “much faster than average”, even within the ICT sector.
- In Spain, reports such as the Employment Map (Fundación Telefónica, Infojobs and Tecnoempleo), or the Guia Spring Professional del mercado laboral (Adecco) identify the roles of software developer, web developer or system administrator as having the highest demand in Spain, with a growing trend. Moreover, the X Informe Infoempleo Adecco shows that bachelor’s degrees in informatics are ranked 3rd according to demand in Spain, only surpassed by degrees in business administration and industrial engineering.
- In the Netherlands, software and application developers are identified as high shortage occupations exhibiting a high demand of specialists with higher education. LinkedIn’s 2020 Emerging Jobs Report for the Netherlands reports that the tech sector drives the majority of emerging professions (such as “full-stack engineers”) and job offers.

Moreover, another noticeable trend in the ICT sector is the globalization of the workplace. The COVID-19 pandemic has promoted remote work, enabling companies to establish international teams. In this context, companies are no longer limited by their location, projects may span several countries and developers need to collaborate remotely and effectively with clients, colleagues, and stakeholders.

The Bachelor’s degree in Software Development and Testing enables students to become software development professionals in a global setting. To this end, it provides a combination of technical skills (algorithms, software engineering best practices, user experience, database design, cloud computing, etc.) and soft skills (teamwork, remote work, communication skills, business and domain knowledge, project management, etc.). From the point of view of technology, students will work on various platforms (i.e., desktop, web, mobile and/or embedded) and be proficient in a variety of programming languages, tools, and services. Moreover, students will be prepared to learn new technologies in order to adapt to future developments.

This joint degree is the successor of the "Bachelor's degree in Techniques for Software Application Development" (BTSD), an official 180 ECTS bachelor's degree offered by UOC from 2021 and which will become extinguished by this joint degree. Even though BTSD was offered in English aiming at a global audience, this new degree expands the international scope of the degree, introduces a focus on quality of software by means of testing and incorporates the expertise from Open University of the Netherlands (OUNL) and Universidad Nacional de Educación a Distancia (UNED).

1.1.2. Aims and objectives of the degree

Graduates from the Bachelor's degree in Software Development and Testing should be capable of creating usable, efficient and high-quality software in any field of application. To this end, they should be able to analyze problems at a suitable level of abstraction; identify the relevant aspects of the problem; assess potential alternatives considering the state of the art; and select and implement the most suitable solution. Furthermore, they should also take into account the deployment, administration, and maintenance of the systems they develop.

To achieve this goal, graduates will learn the principles of software development (i.e., programming, algorithms, data management, software engineering and usability) and the foundations of IT (i.e., computer structure, operating systems, networks, and cloud). They will work with various platforms (i.e., desktop, web, mobile and/or embedded) using a variety of programming languages, tools, and services. Nevertheless, as software development is a rapidly evolving field, graduates should be capable of adapting to emerging technologies and changing environments.

Graduates will apply project management techniques and software development best practices in order to ensure quality and timeliness of the software product. Moreover, given that software development is a team effort, graduates should be able to communicate and collaborate effectively within a team, and with external stakeholders.

Finally, given the potential impact of software in society, graduates will also learn to include considerations such as sustainability, inclusivity, gender perspective or accessibility in their professional activity.

1.1.3. Target graduate profiles

This Bachelor's degree provides the necessary skills and knowledge to work as a **software developer**. This covers a range of roles including *programmers, software engineers, analysts, web developers, app developers* or *software quality assurance (QA) / testers*.

The programme also provides the skills required for the **administration** of the systems where the software applications will be deployed. Therefore, it prepares people to work in roles such as *systems administrator* or *network administrator*.

In the design of this curriculum, we have considered the fundamental rights and equal opportunities between men and women, the principles of equal opportunities and universal accessibility for people with functional diversity, the values of a democratic culture of peace and the principles of sustainability, in accordance with current legislation and the guidelines for the introduction of sustainability in the curriculum prepared by the CRUE.

1.1.4. Specific teaching strategies

One of the core concerns of this degree is teaching techniques, methodologies, tools and best practices to ensure the *quality of software*. Software errors (called *bugs*) can have devastating consequences: economic losses, security vulnerabilities that can be exploited by malicious third parties or, in safety-critical systems, the loss of human lives. Students need to be aware of this reality, as well as the techniques used to avoid, detect and fix bugs in software.

To this end, we build upon the experience of the Erasmus+ European project “Quality-focused Programming EDucation” (QPED, 2020-2023) led by the Open Universiteit (OUNL, Netherlands) with the participation of Universitat Oberta de Catalunya (UOC, Spain), Philipps-Universität Marburg (Germany), and Eindhoven University of Technology (TUE, Netherlands) and Quarterfall (Netherlands). This project focuses on providing resources and tools to improve programming education, introducing the notion of *software testing* from the introductory courses. Moreover, the project aims to encourage software quality beyond functional correctness, considering notions such as code legibility or robustness.

An important aspect of this quality focus in programming education is the use of *automatic assessment tools*. These tools enable continuous formative assessment by providing frequent, informative, and timely feedback about programming assignments. This automatic feedback is a useful complement to the more thorough qualitative feedback provided by course instructors.

1.2. Basic information

Full name of the programme:	Bachelor’s degree in Software Development and Testing
Places:	450
EQF level:	6 (First cycle-Bachelor’s degree)
Degrees awarded:	Bachelor’s degree
Number of ECTS credits:	180 ECTS (3 years)
ISCED field(s) of study:	0613 Software and applications development and analysis

List of the institutions delivering the programme:

Name of the institution	Higher education institution	Degree awarding institution	Role in the consortium
Universitat Oberta de Catalunya (UOC)	yes	yes	coordinator
Open Universiteit (OUNL)	yes	yes	member
Universidad Nacional de Educación a Distancia (UNED)	yes	no	member

Accreditation status per institution:

Name of the institution	Relevant External Quality Assurance Agency	European Approach allowed	Additional information
Universitat Oberta de Catalunya (UOC)	Agència per a la Qualitat del Sistema Universitari de Catalunya (AQU, Spain)	yes	Internal Quality Assurance System Reputation - Pioneering online university
Open Universiteit Netherlands (OUNL)	Nederlands Vlaamse Accreditatie Organisatie (NVAO)	yes	
Universidad Nacional de Educación a Distancia (UNED)	Agencia Nacional de Evaluación de la Calidad y Acreditación (ANECA, Spain)	yes	

1.3. Joint design and delivery

This degree leverages the extensive experience of Universitat Oberta de Catalunya (UOC), Open Universiteit (OUNL) and Universidad Nacional de Educación a Distancia (UNED) in offering ICT bachelor's degrees in a virtual modality:

- UOC has offered ICT bachelor's degrees since 1997. Currently, its bachelor's degrees have more than 6,400 students, 6,400 graduates and 600 academics. These degrees have received positive accreditations from Spanish quality assurance agencies, as well as the Euro-Inf Quality Label from the European Quality Assurance Network For Informatics Education (EQANIE).
- OUNL offers two bachelor programmes (180 EC) in Informatics and Information Science and four master programmes: artificial intelligence (60 EC), which started in

2022/23; Software Engineering (60 EC); Business Process Management and IT (60 EC); and Computer Science (120 EC). Actual figures will follow. In the academic year 2022-2023, 826 students followed the bachelor programme in Informatics and 255 the bachelor programme in Information Science. The number of bachelor diploma's since 2014-2015 is 68 for Informatics and 8 for Information Science. The number of students following the four formerly mentioned master programmes in the academic year 2022-2023 is 687, and 942 master diplomas have been awarded since 2014-2015.

- UNED was founded in 1972 to deliver distance courses in Spanish all over the world using radio, TV and postal communication. Nowadays, UNED takes advantage of technical innovations to support the nearly 4,000 students worldwide currently enrolled in ICT bachelor studies during the course 2021-2022 in 2 degrees (Informatics and Information Technology) taught by 90 professors with the support of 1,000 tutors (which provide content guidance and assess the course assignments) distributed in associated centres placed in 75 different cities from 17 countries, were students can take the exams. The studies are accredited by ANECA, the Spanish official agency. The number of UNED students who have finished ICT bachelor studies is over 65,000.

This degree has been designed in the context of the OpenEU Erasmus+ European University, an alliance of European virtual-learning universities, which begins on December 1, 2024. The OpenEU consortium is coordinated by UOC and has the participation of other universities, being OUNL and UNED two of them. In particular, the main responsibilities of the universities in the context of the OpenEU consortium are:

- UOC leads the Work Package 1 on "Governance and management for a solid and sustainable Open European University, and has an active participation in all other Work Packages (WPs). Furthermore, as coordinator of the consortium, during the first years of the Alliance the UOC will host the OpenEU Secretariat.

- OUNL leads the Work Package on Joint Research and Innovation in Digital Education, and has an active participation in the various WPs of the consortium. OUNL is also a member of all alliance boards.
- UNED leads the Work Package on Internationalisation, and co-leads the Work Package on Making Lifelong Learning for Employability a Reality. UNED also actively participates in all work packages, in addition to being a member of all alliance boards.

2. Learning Outcomes

The learning outcomes of this degree comprise knowledge, skills, and competencies in the field of "**Software and applications development and analysis**", a subfield within ICT.

The level of these learning outcomes correspond to an **EQF Level 6** (first cycle of higher education / bachelor's degree) in the Framework for Qualifications in the European Higher Education Area (FQ-EHEA).

The achievement of these learning outcomes is demonstrated throughout this joint degree by several means and assessment strategies (see Section 5.2 on assessment).

2.1. Knowledge

Upon completing the bachelor's degree, the student will be able to:

- **K1**. Recognise the scientific and mathematical principles that are the foundation of the software development process.
- **K2**. Enumerate and describe the structural elements and operational principles of computers and networks, together with their relevant features from a software perspective.
- **K3**. Identify the features of different types of organizations and the role played in them by ICTs.
- **K4**. Enumerate and describe the steps in the software development life cycle, including the goals, tasks, and challenges involved in each step.
- **K5**. Describe programming languages, tools, and technologies used in the software development process together with the scenarios where they are best suited.

2.2. Skills

Upon completing the bachelor's degree, the student will be able to:

- **S1**. Design, develop and test software applications for different platforms and application domains, applying software engineering best practices.
- **S2**. Verify, validate and measure the quality of software during the development process according to different criteria such as correctness, performance, usability, or security.
- **S3**. Design, set up and manage networks and computer systems, considering the available resources and ensuring their availability and security.
- **S4**. Apply techniques for processing, storing, accessing and managing data, taking into account performance and availability.
- **S5**. Design the user experience of a software application considering the users' goals, needs, and expectations.
- **S6**. Analyze development problems at the appropriate level of abstraction to select the most suitable technological alternative (programming language, algorithm, tool, or component).
- **S7**. Prepare proposals for development projects, taking into account the resources, available alternatives, market conditions and scientific state of the art.
- **S8**. Express ideas clearly and correctly in English both orally and in writing, displaying a command of terms and expressions that are specific to the field of software development, for both academic and professional contexts.

2.3. Competencies

Upon completing the bachelor's degree, the student will be able to:

- **C1**. Collaborate effectively within both academic environments and a software development team or organisation.
- **C2**. Plan and carry out software development projects, presenting the results in a manner suitable for the target

audience (both academic and professional contexts) and defending the conclusions before a panel of experts.

- **C3**. Adapt to new software development technologies and changes in the needs of the market, scientific state of the art, and society.
- **C4**. Apply ICTs in an academic and professional setting.
- **C5**. Conduct academic and professional practice in an honest, ethical, sustainable and socially responsible way, being conscious and respectful of human rights and diversity.

2.4. Regulated Professions

There are no regulated professions related to this degree.

3. Study Programme

3.1. Curriculum

The following table summarises the temporal organisation of the curriculum of the Bachelor's degree in Software Development and Testing:

	Semester 1	**Semester 2**
Year 1	ECTS: 30 Courses: • Online teamwork and cademic skills (basic, S7) • Software engineering (basic, S3) • Fundamentals of programming (basic, S2) • Fundamentals of computers (basic, S6) • Mathematical Foundations I (basic, S1)	ECTS: 30 Courses: • Fundamentals of Information Systems (basic, S7) • Web standards and languages (compulsory, S4) • Programming in practice (basic, S2) • Computer structure (compulsory, S6) • Mathematical Foundations II (basic, S1)
	Semester 3	**Semester 4**
Year 2	ECTS: 30 Courses: • Web programming (compulsory, S4) • Object-oriented programming (compulsory, S2) • Software design patterns (compulsory, S3) • Operating systems (compulsory, S6) • Human factors in computing (compulsory, S3)	ECTS: 30 Courses: • Networks and Internet applications (compulsory, S6) • Introduction to databases (compulsory, S5) • Data structures (compulsory, S2) • Software architecture (compulsory, S3) • Network and system administration (compulsory, S6)
	Semester 5	**Semester 6**
Year 3	ECTS: 30 Courses: • Database design (compulsory, S5) • Software testing (compulsory, S3) • Cloud computing (compulsory, S6) • Optional course I (optional) • Optional course II (optional)	ECTS: 30 Courses: • Mobile app development (compulsory, S8) • Optional course III (optional) • Optional course IV (optional) • Final project (compulsory, S9)

The courses in this degree are conceptually organised into 9 subjects, namely: (S1) mathematical foundations, (S2) programming, (S3) software engineering, (S4) web development, (S5) databases, (S6) computer organisation and administration, (S7) business and soft skills, (S8) application domains, and (S9) projects. In the following, we present each subject in detail. The previous table includes the reference to each subject for each course.

Subject 1. Mathematical foundations	
Number of ECTS credits	12
Type	Basic
Schedule	Semesters 1 and 2
Modality	Virtual
Learning outcomes of the degree associated with this subject	• K1. Recognise the scientific and mathematical principles that are the foundation of the software development process. • S6. Analyse development problems at the appropriate level of abstraction to select the most suitable technological alternative (programming language, algorithm, tool, or component). • S8. Express ideas clearly and correctly in English both orally and in writing, displaying a command of terms and expressions that are specific to the field of software development, for both academic and professional contexts.
Learning outcomes	After completing this subject, students will be able to: • Recognise, explain and use the mathematical concepts and notation used to express knowledge about a particular problem. • Explain the differences and relationship between the different types of numbers (i.e., natural, integer, real and complex). • Solve systems of linear equations. • Apply concepts of linear algebra (e.g., base, linear transformations, diagonalization, etc.) to solve complex problems. • Check properties and perform basic operations on sets. • Manipulate boolean expressions and explain the meaning of a truth table. • Formalise complex natural language statements using a logic notation. • Given a set of logical statements, perform or validate a reasoning to reach sound conclusions.
Courses	All courses have 6 ECTS and are offered in English. • Mathematical Foundations I (Basic: mathematics, Semester 1) • Mathematical Foundations II (Basic: mathematics, Semester 2)

Contents	**Mathematical Foundations I** • Numbers. • Algebraic and geometric tools. • Systems of equations. • Linear applications. • Geometric transformations. **Mathematical Foundations II** • Basic set theory: set operations, relations, functions. • Truth tables and Boole’s algebra. • Language and semantics of propositional logic. • Language and semantics of first-order logic. • Formalisation of natural language expressions. • Theorem proving.

Subject 2. Programming	
Number of ECTS credits	30
Type	Mixed (6 basic, 18 compulsory and 6 optional)
Schedule	Semesters 1, 2, 3, 4 and 5/6
Modality	Virtual
Learning outcomes of the degree associated with this subject	• K5. Describe programming languages, tools, and technologies used in the software development process together with the scenarios where they are best suited. • S1. Design, develop and test software applications for different platforms and application domains, applying software engineering best practices. • S2. Verify, validate and measure the quality of software during the development process according to different criteria such as correctness, performance, usability, or security. • S4. Apply techniques for processing, storing, accessing and managing data, taking into account performance and availability. • S6. Analyse development problems at the appropriate level of abstraction to select the most suitable technological alternative (programming language, algorithm, tool, or component). • S8. Express ideas clearly and correctly in English both orally and in writing, displaying a command of terms and expressions that are specific to the field of software development, for both academic and professional contexts. • C3. Adapt to new software development technologies and changes in the needs of the market and society.

Learning outcomes	After completing this subject, students will be able to: • Recognise and explain different programming paradigms (imperative, object-oriented, functional), understanding their differences and strengths. • Propose algorithms to solve a given problem. • Write algorithms in a given programming language, taking advantage of its features. • Use the most suitable tools for creating, running, maintaining and analysing computer programs (e.g., editors, IDEs, compilers, interpreters, debuggers, etc.). • Detect, understand and fix errors in computer programs. • Measure the temporal and spatial complexity of alternative algorithms for a particular problem, selecting the best alternative. • Describe and use the best known algorithms and data structures to solve fundamental problems (e.g., sorting, search, etc.). • Apply best practices when writing computer code (e.g., naming conventions, comments, modularity and code organisation).
Courses	All courses have 6 ECTS and are offered in English. • Fundamentals of programming (Basic: computing, Semester 1) • Programming in practice (Basic, Semester 2) • Object-oriented programming (Compulsory, Semester 3) • Data structures (Compulsory, Semester 4)
Contents	**Fundamentals of programming** • Algorithm fundamentals: data types, declarations, expressions, algorithmic structures (assignment, conditional statements, loops, etc.), procedures and functions. • Sequences: search/traversal algorithms. • Structured data types: tables and tuples. • Top-down design. • Development tools: IDE, compiler, interpreter. • Programming languages. • Coding conventions (codes, tabs, naming, best practices). **Programming in practice** • History of programming languages. • Top-down design. • Basic data structures: lists, stacks, queues. • Algorithmic complexity. • Fundamental algorithms: sorting (selection-sort, insertion-sort, bubble-sort, quick-sort, merge-sort). • Development tools: IDE, compiler, interpreter, debugger. • Memory management. • Input/output. • Implementing complex algorithms. **Object-oriented programming** • Software design: quality criteria, modularity.

	• Object-oriented paradigm: classes and objects, inheritance, instantiation. • UML class diagrams and their implementation in the Java language. • The Java programming language: fundamentals, data types, operators, arrays, statements, methods, visibility. **Data structures** • Abstract Data Types (ADT): advantages, specification. • Sequential ADTs: stacks, queues, lists. • Tree: types (binary vs. n-ary, sorted, search), tree traversals. • Priority queues: heaps, the heapsort algorithm. • Dictionaries: hash functions, hash tables and AVL trees • Graphs: graph traversals. • Designing data structures: criteria to select the most suitable ADT. • Using and specifying ADT libraries. • Implementing ADTs in Java.

Subject 3. Software engineering	
Number of ECTS credits	30
Type	Mixed (6 basic and 24 compulsory)
Schedule	Semesters 1, 3, 4 and 5
Modality	Virtual
Learning outcomes of the degree associated with this subject	• K4. Enumerate and describe the steps in the software development life cycle, including the goals, tasks, and challenges involved in each step. • K5. Describe programming languages, tools, and technologies used in the software development process together with the scenarios where they are best suited. • S1. Design, develop and test software applications for different platforms and application domains, applying software engineering best practices. • S2. Verify, validate and measure the quality of software during the development process according to different criteria such as correctness, performance, usability, or security. • S5. Design the user experience of a software application considering the users' goals, needs, and expectations. • S6. Analyse development problems at the appropriate level of abstraction to select the most suitable technological alternative (programming language, algorithm, tool, or component). • S7. Prepare proposals for development projects, taking into account the resources, available alternatives and market conditions. • S8. Express ideas clearly and correctly in English both orally and in writing, displaying a command of terms and expressions that are specific to the field of software development, for both academic and professional contexts.

	• C2. Plan and carry out software development projects, presenting the results in a manner suitable for the target audience and defending the conclusions before a panel of experts. • C3. Adapt to new software development technologies and changes in the needs of the market and society. • C5. Conduct academic and professional practice in an honest, ethical, sustainable and socially responsible way, being conscious and respectful of human rights, gender perspective and diversity.
Learning outcomes	After completing this subject, students will be able to: • Enumerate and describe the steps in the software development life cycle, including the goals, tasks, and challenges involved in each step. • Design the static and dynamic components of a software system using the suitable structural and behavioural UML diagrams. • Apply agile project methodologies to software development projects. • Enumerate, describe and apply (when possible) software design patterns in the design of software systems. • Design the user experience of a software application considering the users' goals, needs, and expectations. • Assess the user experience of a given software system. • Design and implement the architecture of complex distributed systems. • Define different types of tests and test strategies in order to check the correct operation of a software system.
Courses	All courses have 6 ECTS and are offered in English. • Software Engineering (Basic, Semester 1) • Human factors in computing (Compulsory, Semester 3) • Software design patterns (Compulsory, Semester 3) • Software architecture (Compulsory, Semester 4) • Software testing (Compulsory, Semester 5)
Contents	**Software Engineering** • Introduction to object-oriented software engineering (i.e., software lifecycle, CASE tools, OMG standards and UML). • Object-oriented paradigm: classes and object, abstraction and classification, inheritance and polymorphism. • UML: static and dynamic diagrams. • Requirement engineering: Gathering and documenting software requirements. • Object-oriented design. • Application to software development use cases. • Modelling in UML (i.e., introduction to CASE tools and best practices). **Human factors in computing** • Introduction to human-computer interaction. • Accessibility and usability. • Elicitation of user requirements.

	• Metaphors, styles, and paradigms. • Prototyping. • Evaluation. **Software design patterns** • Patterns: concept and types. • Designing software using patterns. • Pattern catalogue. • Practical use cases of applying software design patterns. **Software architecture** • Distributed software architectures. • Microservice architectures: concept, design, implementation, connecting microservices. • Quality in distributed software. • DevOps, Continuous Delivery / Continuous Integration (CD/CI) and agile development. **Software testing** • Overview of testing. • Models for testing. • Design of testcases. • Coverage criteria. • Quality of test sets.

Subject 4. Web development	
Number of ECTS credits	12
Type	Compulsory
Schedule	Semesters 2 and 3
Modality	Virtual
Learning outcomes of the degree associated with this subject	• K5. Describe programming languages, tools, and technologies used in the software development process together with the scenarios where they are best suited. • S1. Design, develop and test software applications for different platforms and application domains, applying software engineering best practices. • S2. Verify, validate and measure the quality of software during the development process according to different criteria such as correctness, performance, usability, or security. • S5. Design the user experience of a software application considering the users' goals, needs, and expectations. • S6. Analyse development problems at the appropriate level of abstraction to select the most suitable technological alternative (programming language, algorithm, tool, or component).

	• S8. Express ideas clearly and correctly in English both orally and in writing, displaying a command of terms and expressions that are specific to the field of software development, for both academic and professional contexts.
Learning outcomes	After completing this subject, students will be able to: • Enumerate and explain the different standards, languages, and technologies involved in the development of a web-based application. • Design and build front-ends for web-based applications using HTML and CSS. • Layout complex websites including tables or forms using HTML and CSS. • Validate the HTML/CSS and check the accessibility of a website. • Develop client-side applications using JavaScript.
Courses	All courses have 6 ECTS and are offered in English. • Web standards and languages (Compulsory, Semester 2) • Web programming (Compulsory, Semester 3)
Contents	**Web standards and languages** • Introduction and basic concepts. • HTML and CSS fundamentals. • Lists, images, links, div, span and the CSS box model • Validation and accessibility. • Tables. • Forms. • Layout. **Web programming** • Client-side programming: scripting languages in browsers. • Introduction to object-oriented programming. • Object orientation in JavaScript. • Introduction to DOM. • DOM manipulation with JavaScript.

Subject 5. Databases	
Number of ECTS credits	12
Type	Compulsory
Schedule	Semesters 4 and 5
Modality	Virtual
Learning outcomes of the degree associated	• K5. Describe programming languages, tools, and technologies used in the software development process together with the scenarios where they are best suited.

<table>
<tr><td>with this subject</td><td>

S4. Apply techniques for processing, storing, accessing and managing data, taking into account performance and availability.
S6. Analyse development problems at the appropriate level of abstraction to select the most suitable technological alternative (programming language, algorithm, tool, or component).
S8. Express ideas clearly and correctly in English both orally and in writing, displaying a command of terms and expressions that are specific to the field of software development, for both academic and professional contexts.
</td></tr>
<tr><td>Learning outcomes</td><td>After completing this subject, students will be able to:

Explain the role and types of database management systems (e.g., NoSQL).
Explain the fundamentals of the relational model and the operations in relational algebra.
Write complex SQL statements to query a database, create tables, insert/delete/update data or define stored procedures.
Define transactions to group related SQL statements in an atomic way.
Execute SQL statements from computer code (e.g., JDBC).
Propose the logical design of a database, considering quality issues (i.e., normalization).
Set up the access control and physical design of a database to ensure its security and efficiency.
Express and check integrity constraints using SQL constructs and triggers.
</td></tr>
<tr><td>Courses</td><td>All courses have 6 ECTS and are offered in English.

Introduction to databases (Compulsory, Semester 4)
Database design (Compulsory, Semester 5)
</td></tr>
<tr><td>Contents</td><td>Introduction to databases

Introduction to data, databases, and database management systems.
Relational model and relational algebra.
SQL language (DML): Create Table, Insert, Delete, Update, Select, Stored Procedures.
Transactions.
Accessing a database from a program (JDBC).

Database design

Logic design of database (transforming the conceptual model to the logic model: data structures, integrity constraints and operations).
Reconsidering the logic and conceptual model (normalization, denormalization).
Specifying and implementation access control in databases
Physical design of a database (e.g., indices, etc.).
Specifying and implementing integrity constraints in a database (triggers).
Processing SQL queries and commands.
</td></tr>
</table>

Subject 6. Computer organisation and administration	
Number of ECTS credits	42
Type	Mixed (6 basic, 30 compulsory and 6 optional)
Schedule	Semesters 1, 2, 3, 4, 5 and 6
Modality	Virtual
Learning outcomes of the degree associated with this subject	• K1. Recognise the scientific and mathematical principles that are the foundation of the software development process. • K2. Enumerate and describe the structural elements and operational principles of computers and networks, together with their relevant features from a software perspective. • S3. Design, set up and manage networks and computer systems, considering the available resources and ensuring their availability and security. • S6. Analyze development problems at the appropriate level of abstraction to select the most suitable technological alternative (programming language, algorithm, tool or component). • S8. Express ideas clearly and correctly in English both orally and in writing, displaying a command of terms and expressions that are specific to the field of software development, for both academic and professional contexts.
Learning outcomes	After completing this subject, students will be able to: • Encode integer and floating-point numbers in binary format and operate with them. • Design digital circuits using logic gates and combinational blocks. • List the basic hardware components of a computer (e.d., processor, memory, I/O devices), explaining their relevant features from a programmer's perspective, and operate with them using the assembly language. • Explain the basic components of operating systems and access them from a computer program. • Explain the architecture and different types of computer networks. • Design and implement applications and systems that communicate using computer networks. • Set up computer systems and networks according to user and organisational needs and regulations, ensuring their security, efficiency, reliability, and availability. • Select, set up and manage cloud services.
Courses	All courses have 6 ECTS and are offered in English. • Fundamentals of computers (Basic: computing, Semester 1) • Computer structure (Compulsory, Semester 2) • Operating systems (Compulsory, Semester 3) • Networks and Internet applications (Compulsory, Semester 4) • Network and system administration (Compulsory, Semester 4) • Cloud computing (Compulsory, Semester 5)

	• Security in Computer Networks (Elective, Semester 5/6) • Embedded Systems (Elective, Semester 5/6)
Contents	**Fundamentals of computers** • Fundamental concepts about computer: Introduction to computer; Hierarchical structure of a computer; Von Newmann architecture. • Digital representation of information: Base change; binary encoding of natural and integer members. Operations. Floating-point representation: Precision, range. • Combinational circuits: Truth tables, two-level circuit design; Karnaugh method; Basic combinational blocks (multiplexer, encoder, adder). • Sequential circuits: D flip-flop; Register; Moore's model • Basic structure of a computer: Description of the CISCA architecture. Introduction to assembly language using the CISCA instruction set. **Computer structure** • Instruction sets and addressing modes. • Input/Output systems (i.e., programmed, interrupt-based, DMA; differences and performance assessment). • Memory hierarchy (i.e., types of memory, characteristics, accessing data in memory, concept of memory hierarchy, analysis of its effectiveness). • Programming in assembly (i.e., control-flow in assembly programs, peripheral management, calling subroutines and passing parameters, programming I/O devices). **Operating systems** • Introduction to operating systems. • Operating systems as virtual machines. • Memory management. • Input-output devices. • File systems. • Concurrency and communication. • State-of-the-art of operating systems. • Introduction to the GNU/Linux operating system and command line interface using shell scripts. • System calls using the C programming language **Networks and Internet applications** • Basic concepts of communication networks and the Internet. • Applications and security on the Internet. • Web-based systems. • Security on the Web. • Internet architectures: TCP/IP. **Network and system administration** • Goals and duties of a system administrator.

	• Communication between different operating systems. • Concepts and tools for network and system administration. • Generic user needs. • Integrating software in an organisation. • Strategies to maximise the security of a computer system. • Data protection: Legal aspects of users and system administrators. • Planning the computer system of an organisation • Managing an IT department. • Contingency plans. • Security-related laws and regulations. **Security in Computer Networks** • Security and risks in computer networks, computer attacks and defence. • Firewalls and intrusion detention systems. • Security in the communication layer (Wireless networks, IPsec, TLS) and in the application layer (SSH, VPN, Secure email). **Cloud computing** • Introduction to Cloud Computing, IaaS/PaaS/SaaS models. • Virtualization (i.e., hypervisors). • Architectures and types of Cloud services. • Providers of public cloud and related services (e.g., Amazon AWS and Microsoft Azure). • DevOps. **Embedded Systems** • Identification and study of embedded systems (medical devices, autonomous systems, etc.) • Microcontrollers and basic peripherals • Advanced peripherals • Real-time operating systems

Subject 7. Business and soft skills	
Number of ECTS credits	18
Type	Mixed (12 basic and 6 optional)
Schedule	Semesters 1, 2 and 5/6
Modality	Virtual
Learning outcomes of the degree associated with this subject	• K3. Identify the features of different types of organizations and the role played in them by ICTs. • K5. Describe programming languages, tools and technologies used in the software development process together with the scenarios where they are best suited.

	• S6. Analyze development problems at the appropriate level of abstraction to select the most suitable technological alternative (programming language, algorithm, tool or component). • S7. Prepare proposals for development projects, taking into account the resources, available alternatives and market conditions. • S8. Express ideas clearly and correctly in English both orally and in writing, displaying a command of terms and expressions that are specific to the field of software development, for both academic and professional contexts. • C1. Collaborate effectively within a software development team or organization. • C2. Plan and carry out software development projects, presenting the results in a manner suitable for the target audience and defending the conclusions before a panel of experts. • C3. Adapt to new software development technologies and changes in the needs of the market and society. • C4. Apply ICTs in an academic and professional setting. • C5. Conduct academic and professional practice in an honest, ethical, sustainable and socially responsible way, being conscious and respectful of human rights, gender perspective and diversity.
Learning outcomes	After completing this subject, students will be able to: • Select and use tools and services to create, edit and share different types of content (e.g., documents, diagrams, spreadsheets, presentations, videos, audio, etc.). • Apply generative artificial intelligence tools in academic and professional settings, taking into account its benefits, shortcomings and ethical implications. • Collaborate effectively with a team of peers in an online setting, using the most suitable tools and strategies. • Manage software development projects remotely. • Understand the goals and characteristics of different types of documents, as well as the dimensions to examine when crafting them (context, goal, format, structure, vocabulary, syntax, punctuation, …). • Communicate effectively in academic and professional settings. • Identify the different types of information systems, their defining features and their role in organisations.
Courses	All courses have 6 ECTS and are offered in English. • Online teamwork and academic skills (Basic:computing, Semester 1) • Fundamentals of information systems (Basic: business, Semester 2) • Communication skills (Elective, Semester 5/6)

Contents	**Online teamwork and academic skills** • Online browsing, communication and collaboration. • Remote preparation and presentation of documents. • Effective and ethical use of generative AI tools. • Tools and techniques for remote software development. • Remote IT project planning and management. • Project management principles from PMI. **Communication skills** • Basic concepts in theory of oral communication, application of concepts in business conversations. • Writing technical texts and how to approach the process methodically. • Professional presentations: structure, formulation, delivery. • Reflection, analysis, feedback on the aforementioned points. **Fundamentals of information systems** • What information systems are. • Organizational Strategy, Competitive Advantage, and information systems. • Ethics and Privacy. • Information Security. • Data and Knowledge Management. • eBusiness and eCommerce. • Social Computing. • Information Systems within the organisation. • Customer Relationship Management and Supply Chain Management. • Business Analytics. • Acquiring Information Systems and Applications. • Artificial Intelligence.

Subject 8. Application domains	
Number of ECTS credits	18
Type	Mixed (6 compulsory and 12 optional)
Schedule	Semesters 5 and 6
Modality	Virtual
Learning outcomes of the degree associated with this subject	• K5. Describe programming languages, tools and technologies used in the software development process together with the scenarios where they are best suited. • S1. Design, develop and test software applications for different platforms and application domains, applying software engineering best practices.

	• S4. Apply techniques for processing, storing, accessing and managing data, taking into account performance and availability. • S6. Analyze development problems at the appropriate level of abstraction to select the most suitable technological alternative (programming language, algorithm, tool or component). • C3. Adapt to new software development technologies and changes in the needs of the market and society. • C5. Conduct academic and professional practice in an honest, ethical, sustainable and socially responsible way, being conscious and respectful of human rights, gender perspective and diversity.
Learning outcomes	After completing this subject, students will be able to: • Understand the specific context, goals, and challenges in a given application domain and identify the most suitable methods, algorithms, technologies, and tools. • Design, develop, deploy and test mobile applications. • Understand the challenges faced when building distributed systems and the different algorithms and systems that can be used to address them. • Understand the concept of blockchain and smart contract, as well as its potential applications and limitations. • Use existing languages, libraries, and tools to develop blockchain applications.
Courses	All courses have 6 ECTS and are offered in English. • Mobile app development (Compulsory, Semester 6) • Distributed systems (Elective, Semester 5/6) • Blockchain and smart contracts (Elective, Semester 5/6)
Contents	**Mobile app development** • Evaluation of new techniques, framework, and libraries for mobile app development. • Modern approach to mobile app development. • Ensuring mobile app quality in practice. **Distributed systems** • Algorithmic perspective on distributed systems: snapshots, waves, deadlock detection, termination, election. • Probabilistic distributed algorithms. • Implementation of algorithms. • Correctness and complexity of distributed algorithms. **Blockchain and smart contracts** • Introduction to blockchains (i.e., operating principles, types, consensus protocols, hashes and cryptography, wallets, transactions, advantages, challenges and applications). • Smart contracts (i.e., introduction to smart contracts, standards, programming in Solidity, deploying a smart contract, oracles). • Front-end development (i.e., introduction to front-end development of Apps, JavaScript libraries for interacting with smart contracts, programming in JavaScript).

	• Decentralized and permanent storage (e.g., introduction to the problem, IPFS / Piñata, Other solutions).

Subject 9. Projects	
Number of ECTS credits	24
Type	*Mixed (12 compulsory and 12 optional)*
Schedule	*Semesters 5 and 6*
Modality	*Virtual*
Learning outcomes of the degree associated with this subject	This subject contributes to all learning outcomes identified for the bachelor's degree.
Learning outcomes	After completing this subject, students will be able to: • Define, plan and execute a complex software development project on a particular topic. • Apply, adapt and combine all the learning outcomes acquired during the bachelor's degree using professional, creative and innovative criteria. • Research a specific problem or domain, identifying and citing the relevant sources of documentation and information. • Work autonomously as a software developer by setting goals, defining priorities and creating a schedule with milestones. • Show initiative and proactivity to introduce innovations, identify problems, search for information and propose creative solutions. • Consider quality concerns in the preparation of deliverables and artefacts (software, documents, presentations, …). • Adapt to new scenarios and unexpected problems, proposing the best possible solutions according to the situation. • Communicate and defend the decisions taken and the results of the project effectively to a technical audience, both orally and in writing.
Courses	All courses have 6 ECTS and are offered in English. • Final project (Compulsory, Semester 6) • Internship (Elective, Semester 5)
Contents	**Final project** • The student will have to develop a complete project applying all the knowledge, skills, and competencies acquired during the degree. **Internship** • The contents of the internship will depend on the particular organisation and project where the internship is performed.

The mapping between the learning outcomes of the Bachelor’s degree and the learning outcomes of the subjects can be found here.

3.2. Credits

The workload in the Bachelor’s degree in Software Development and Testing is measured using European Credit Transfer System (ECTS) credits.

The degree includes 180 ECTS credits, organized into courses with 6 ECTS credits, with the “Final project” and “Internship” (included as an elective course) having 12 ECTS credits to account for their broader scope.

The credits of the degree are distributed among the types of courses in the following way:

Type of course	ECTS credits
Basic	48
Compulsory	96
Elective	24
Mandatory internship	0
Final project	12
Total	**180**

3.3. Workload

3.3.1. Degree and course workload

The Bachelor's degree in Software Development and Testing includes **180 ECTS** credits.

In this bachelor's degree, 1 ECTS credit corresponds to **25 hours** of student workload. This workload includes all activities to be performed in the course, such as reading learning resources, installing and using software tools, participating in the classroom forums, or solving the continuous and final assessment activities. The design of each course takes this workload into account when defining learning outcomes, choosing learning resources or designing assessment activities.

All courses in the bachelor's degree have **6 ECTS** (150 hours of student workload) with the exception of the "Internship" and "Final project" which have **12 ECTS** each (300 hours of student workload). Considering a typical semester with 15 weeks of continuous assessment and 2–3 weeks of final assessment, this roughly corresponds to a workload of about **8–9 hours per week** for each course (i.e., assuming an ideal course where the workload is evenly distributed across the semester).

3.3.2. Duration of the degree

The Bachelor's degree in Software Development and Testing is intended to be completed in a period **from 3 years (for full-time students) to 6 years (for part-time students)**.

This bachelor's degree is a distance education programme offered through a virtual campus. The flexibility and asynchronicity of this learning model makes it particularly amenable to part-time students. Thus, according to historical data from similar degrees at UOC/OUNL/UNED, part-time students are expected to make up the vast majority of the student population, with a minority of full-time students.

In order to accommodate part-time students who may have family or work commitments, there is no minimum number of courses to be enrolled in each semester. Nevertheless, tutors recommend enrolling about **2–3 courses per semester for part-time students** and a maximum of **5 courses per semester for full-time students**. In this way, a proper balance between availability, workload, and progress within the degree can be reached. Students who want to enrol more than 6 courses in a given semester need to formally request permission to do so, justifying why this exceptional enrolment is necessary and how their background and personal circumstances contribute to their chances of success.

The following table summarizes the ideal workload and time to complete the degree for these two student profiles:

	Target workload per semester		Target time to complete the degree	
Student profile	**Courses**	**ECTS Credits**	**Years**	**Semesters**
Part-time student	2-3	12-18 (avg. 15)	6	12
Full-time student	5	30	3	6

Notice that the time to complete the degree may be shorter for students that can recognize credits by means of their prior education and background (see Section 4.2 - Recognition).

4. Admission and Recognition

4.1. Admission

4.1.1. General admission regulations and procedures

The joint degree will apply UOC's academic regulations for admission to bachelor's degrees. Title 1 of these regulations is dedicated to admission requirements and regulations. Specific support will be provided to students with accessibility requirements (see commitment to accessibility).

On the other hand, the general information about the admission and enrolment process for a bachelor's degree at UOC is presented on the following website.

4.1.2. Specific admission criteria for this degree

English is the language of instruction for this bachelor's degree. Thus, admission to the degree requires candidates to demonstrate a level of proficiency in English of at least B2, according to the Common European Framework of Reference for Languages (CEFR).

Proof of competency in English can be provided in different ways:

- By means of an official certificate accrediting this level of linguistic competence in English.
- Having passed the English courses corresponding to level B2 in university degrees or universities' foreign language centres.
- Having completed the baccalaureate or a university degree abroad in English.
- Having completed a university degree focused on the study of foreign languages, with at least 30 ECTS devoted to English exercising the four language skills (i.e., speaking, listening, reading and writing).

- Taking an official English test (such as LinguaSkill) that assesses the four language skills (i.e., speaking, listening, reading and writing) and providing a certificate of a level at least equivalent to B2.

4.2. Recognition

4.2.1. General recognition regulations and procedures

Title 5 of UOC's academic regulations are devoted to recognition of qualifications and prior learning. On the other hand, general information about the procedures for credit recognition is described in the portal.

4.2.2. Specific recognition criteria for this degree

The following tables indicate the specific recognition criteria for students accrediting vocational training (see first table); professional experience or non-official university degrees (see second table); or extinguished degrees (see last table).

Recognition of vocational training	At most: 45 + 12 ECTS
Credit recognition of vocational training will not exceed 25% of the total number of credits of the degree (180 ECTS). Recognition tables will be validated by the Generalitat de Catalunya. *Furthermore, according to the Spanish regulations of vocational training (art. 130 of RD659/2023, only in Spanish), an additional amount of up to 12 credits can be recognized for students accrediting specialized vocational training directly related to the degree. An example of these specialized vocational training is "Curso de especialización de Desarrollo de aplicaciones en lenguaje Python". The full list of courses will be published in the recognition tables, which will be validated by the Generalitat de Catalunya.*	

Recognition of professional experience **Recognition of non-official university degrees**	At most: 24 ECTS
Credit recognition cannot exceed 15% of the total number of credits of the degree (180 ECTS).	

Recognition of credits for extinguished degrees	At most: 168 ECTS
Students from the Bachelor's degree in Techniques for Software Application development, extinguished by this joint degree, will be able to validate courses according to the following table.	

Bachelor’s degree in Techniques for Software Application Development (old degree)	Bachelor’s degree in Software Development and Testing (new degree)
Algebra	Mathematical Foundations I
Logic	Mathematical Foundations II
Web-based Teamwork	Online teamwork and academic skills
Fundamentals of Programming	Fundamentals of Programming
Programming in Practice	Programming in Practice
Web Programming	Web Programming
Software Engineering	Software Engineering
Computer Structure	Computer Structure
Entrepreneurship	Entrepreneurship
Object-Oriented Programming	Object-Oriented Programming
Data Structures	Data Structures
Web Standards and Languages	Web Standards and Languages
Software Design Patterns	Software Design Patterns
Software Architecture	Software Architecture
Human-Computer Interaction	Human-Computer Interaction
Introduction to Databases	Introduction to Databases
Database Design	Database Design
Operating Systems	Operating Systems
Networks and Internet Applications	Networks and Internet Applications
Network and System Administration	Network and System Administration
Cloud Computing	Cloud Computing
Mobile Application Development	Mobile app development
Business and IT Management	Fundamentals of Information Systems
Communication Skills for ICT Professionals	Communication Skills

Distributed Systems	Distributed Systems
Security in Computer Networks	Security in Computer Networks
E-commerce	E-commerce
Fundamentals of Computers	Fundamentals of Computers
Embedded systems	Embedded systems
Internship	Internship

4.3. Mobility

UOC's academic regulations cover mobility in its article 31 and in chapter 3 of Title III. External students applying for a placement at UOC can find all the relevant information in the University portal.

Students in the programme can find information about mobility in the University portal. Guides are also available to help students process all the procedures related to Erasmus grants (e.g., Welcome Presentation, Guide for Bachelor's and Master's degrees, and Guide for internships).

This project takes place in the context of the OpenEU Erasmus+ European University initiative. OpenEU provides a digital vehicle for students and staff to build their curricula, enhance their employability and careers, and contribute to a more diversified and inclusive higher education landscape and stronger base for regional developments in Europe.

OpenEU will set the Alliance-wide means for an inclusive, seamless and greener mobility for OpenEU learners and staff by combining different formats of mobility: virtual, hybrid, blended and physical. By 2028, at least 50% of students who graduate from an OpenEU University and at least 10% of OpenEU full-time academic and administrative staff will have participated in at least one mobility programme.

5. Learning, Teaching and Assessment

5.1 Learning and teaching

5.2.1. Basic, mandatory, and optional courses

The courses and their types are shown in the following table.

Course Type	Course Name
Basic	• Fundamentals of information systems • Fundamentals of programming • Fundamentals of computers • Mathematical foundations I • Mathematical foundations II • Online teamwork & Academic Skills • Programming in practice • Software engineering
Mandatory	• Cloud computing • Computer structure • Data structures • Database design • Human factors in computing • Introduction to databases • Mobile app development • Network and internet applications • Network and system administration • Object-oriented programming • Operating systems • Software architecture • Software design patterns • Software testing • Web programming • Web standards and languages
Optional	• Blockchain and smart contracts • Communication skills • Distributed systems • Embedded systems

- Internship
- Security in Computer Networks

Learning activities in this degree

During the Bachelor's degree, the courses will put into practice different learning activities and methods. In the following, we briefly describe the most representative ones.

1. **Debate:** Discussion among peers about a selected topic, following a predetermined structure. It promotes communication skills and the ability to reason and provide arguments to support a position.
2. **Information search:** Identifying relevant sources of information (e.g., publications, reports, documentation, or databases), crafting suitable search queries and selecting, understanding and processing information, possibly collating multiple sources.
3. **Case study:** Starting from the description of a complex realistic scenario in a professional context, a technical problem must be analysed in order to propose that suits the particular features of the scenario being studied.
4. **Written exposition:** Creation of a textual document (e.g., report, website, requirement document, project or documentation) that describes a problem, project or technical solution. It promotes communication and synthesis skills.
5. **Oral presentation:** Exposing the key ideas about a problem, project or technical solution, considering temporal constraints and taking advantage of available aids (e.g., slides).
6. **Problem solving:** Addressing challenges using a diversity of knowledge and skills.
7. **Theoretical questions:** Answering questions about the facts, principles, and theories in a software development field.

8. **Exercises and practical activities:** Applying known methods, tools, and techniques to address a particular situation.
9. **Programming:** Developing computer code for a method, application, or service that satisfies a set of given requirements. Depending on the assignment, this code may be created from scratch or interacting with existing code.
10. **Project:** Defining, planning, executing and finalising a project that applies a comprehensive set of skills.

Learning methodologies in this degree

This bachelor's degree will consider the following learning methodologies:

1. **Guided instruction by reading scientific and technical documentation:** A curated selection of scientific and technical documents is read to gain information about a particular method, tool, or technology. Reading is followed by examples and exercises to consolidate learning.
2. **Learning by doing:** A practical skill is trained by solving realistic tasks and situations, learning from successes and mistakes.
3. **Case-based learning:** The learning activity is focused around a case study that resembles a real-world scenario, that provides motivation and features that need to be taken into account.
4. **Problem-based learning:** Learning activities are focused around a problem that needs to be solved. This problem motivates learning the necessary skills and competencies to solve it.
5. **Project-based learning:** Learning activities are focused around the planning and execution of a complex project with particular goals.
6. **Collaborative learning:** Resolution of complex activities as a team effort, requiring the definition of the target goals, work

schedules, distribution of work, communication, and coordination.

See [Annex A.2.1. Activities, methodologies and assessment methods]() for a mapping among the activities, methodologies, and assessment methods

5.1.2. Mandatory internships

The degree does not include mandatory internships. Specific support will be provided for those students taken the optional internship course. In particular, the coordinating university will provide assistance with placement (e.g., helping with the definition of the internship agreement for the company and the student) and supervision (e.g., following the learning process and ensuring that the internship addresses the required competencies of the course). More details about the optional 12 ECTS “Internship” course are provided in the annex.

5.1.3. Final project

The final project involves the definition, planning, execution and defence of a software development project.

Final projects are organised around **thematic areas**, which may deal with a particular technology or application domain. In each area, a team of expert course instructors act as the **project supervisors** who will monitor and support the students during the semester. Furthermore, depending on the area, students may propose a topic for their final topic (to be validated by the project supervisor) or receive a closed set of topics from which to choose. In both scenarios, the project should have a practical orientation, apply the learning outcomes acquired during the degree and have a scope that corresponds to its workload.

The first activity of the final project is the definition of a **project plan**, outlining the goals, tasks, resources, schedule, and risks. The second and third assessment activities correspond to the **execution of the project**, in which the student carries out and monitors the proposed plan. During this time, the project supervisor provides mentoring, support and personalised feedback. The project concludes with a **presentation and defense** of the project in front of a board of experts, which will be conducted synchronously through a public videoconference.

Learning methodologies of the final project

- Learning by doing
- Project-based learning

Learning activities in the final project

- Debate
- Information search
- Written exposition
- Oral presentation
- Project

(These activities have been described in Section 5.2.1)

5.2 Assessment of students

5.2.1. Assessment of basic, mandatory and optional courses

The following table lists the assessment systems that will be used in this bachelor's degree.

	Assessment system
1	Continuous assessment tests
2	Practical activities
3	Final project
4	Synthesis tests
5	Exams

The **assessment model** focuses on continuous assessment (CA) and may include a final assessment (FA). The CA consists of carrying out and passing the activities/tests established in the course plan, through which the achievement of the learning outcomes of the degree is ensured progressively, and those defined for each course. For the courses with FA, the final assessment is carried out at the end of the semester. There are two types of final tests: the Synthesis Test (ST), to assess students who have passed the continuous assessment (CA), with a maximum duration of 1 hour. The Exam (EX) evaluates the achievement of the learning outcomes of the subject in full, and with a maximum duration of 2 hours.

The assessment model of each course takes into consideration **pedagogical aspects** and, whenever possible, takes advantage of innovative methods to facilitate and encourage students to achieve the learning outcomes of the course. Technical aspects are also taken into account to ensure an effective and rigorous evaluation process.

To ensure a common assessment criteria not only in the courses but also across the institutions involved in the Bachelor’s Degree, the programme will deploy standardized rubrics whenever possible and once validated by the faculty staff. Furthermore, all courses will follow a common grading scheme according to the approach provided within the framework of the programme. These measures aim to offer a unified grading scheme in the programme, regardless of the institution providing the course.

To verify identity and authorship in student submissions, UOC applies identity recognition systems. To this end, students may be required to identify themselves by means of an ID, as well as the use of a microphone, camera or other tools during the evaluation. To detect and prevent plagiarism, specific tools and strategies will be used.

Article 113 of the UOC’s academic regulations defines situations that are considered “*irregular conduct*” as well as the academic disciplinary proceedings that may result from this conduct.

UOC systematically reviews the effectiveness of the mechanisms that ensure the rigour of the evaluation and the academic results to improve the process.

5.2.2. Assessment of mandatory internships

The degree does not include mandatory internships. Details about the optional 12 ECTS "Internship" course are provided in the annex A.2.2. Internship.

5.2.3. Assessment of the final project

Final projects are assessed by means of a public and synchronous defense in front of a board of examiners. Title IV of UOC's academic regulations states the rules governing final projects.

The board of examiners is formed by three members:

- The course instructor who has tutored the student during the semester.
- The coordinating professor of the final project.
- Another expert, which can be a lecturer at one of the institutions or a member of the affiliated teaching staff.

During the semester, the course plan defines several milestones in which the student submits partial deliverables to his final project supervisor. These milestones are used to define the project goals and schedule, check progress, and provide early feedback.

The final submission includes several deliverables:

- A **document**, describing the goals, planning, execution and results of the project.
- A **presentation** of the project in video format that summarises the most relevant aspects of the project.
- In most cases, there will also be a **product**, such as an application, tool, or service, developed as part of the project.

The defense of the final project takes place after the final submission, once the board of examiners has been reviewed the deliverables. In a public synchronous session, the board of examiners raises questions to the student about the goals, development, and results of the project. The student's answers may lead to further questions and clarifications.

The final mark will consider the following elements:

1. Continuous work during the semester
2. Project document
3. Presentation
4. Product (if applicable)
5. Answers during the defense

The final project supervisor provides an assessment of (1), and the board of examiners assesses elements (2-5). Considering this information, the board of examiners prepares an assessment report that summarises the board's conclusions and the proposed mark.

6. Student Support

Student support will be provided via the coordinating university, thus enabling a centralised and coherent support service. However, the other participating universities will also be involved in the support process whenever students require specific assistance related to the courses they offer. Through the established coordination bodies, regular contact will be maintained to ensure that partner institutions contribute as needed, allowing for efficient follow-up and a shared responsibility in supporting students across the joint programme.

Regarding the offered student support, UOC offers to their students **Online support channels, explained as follows at the Student Guide.** Our goal is to provide personalized answers to student doubts **quickly, efficiently and with the highest levels of quality**. UOC provides different **online support channels**, to which students can send their queries any day of the week, **from anywhere and at any time**. Students can find support via the Virtual Campus (i.e., Help Service), social media accounts (e.g., UOC's twitter), and other means. In addition to these help channels, we also occasionally provide other channels to help students to swiftly deal with any queries or incidents, for instance, those that could arise when sitting final tests or defending final projects online.

7. Resources

7.1 Staff

As it can be seen in the annex **A.1.10. Academic staff CVs**, the teaching staff of the Bachelor's Degree program includes a total of 26 members. The 96% of the teaching staff hold a doctoral degree. The most significant teaching contributions emphasize extensive experience in both online and face-to-face education, as well as innovative approaches recognized in institutional settings and related to the topics developed in the program. In terms of scientific output, faculty members actively participate in research projects and publish regularly in indexed journals, with several holding prestigious accreditations and awards. This combination of teaching excellence and research rigour highlights the high-quality profile of the academic team supporting the program.

The coordinating University will offer special training sessions to introduce and describe the facilities and the assessment methods used during the Bachelor's Degree (see next section and section 5.2, respectively) to the other Universities participating in the programme. These sessions will mainly focus on the teaching and learning approach followed, including the learning management system, thus ensuring that the teaching staff is versed in the methodological and technical aspects of their courses, and guaranteeing a proper alignment along all courses.

7.2 Facilities

The UOC offers distance learning and uses ICT through its Virtual Campus, to provide spaces, tools, and resources that allow the development of learning activity, as well as information and procedures. In its campus, the virtual classroom allows access to the courses enrolled and consult their teaching plan, resources, activities, record of evaluation results, as well as facilitate the relationship between teaching staff and students. It has two basic communication spaces: the teacher's announcements and the discussions.

The typology of classrooms can be for:

- Standard courses: the teaching action follows a common learning plan. The attention is carried out through the personal mailboxes of each student, group mailboxes and the dynamization of the course instructor. The ratio of students per classroom is a maximum of 70.
- Courses with special dedication: the elements of individualization prevail so that each student or small groups of students follow a differentiated learning itinerary. The ratio of students is proposed to be a maximum of 50 per classroom. In Project courses , it is advisable not to exceed 40 students per classroom.
- The final project needs to carry out an individualized and personalized follow-up and tutoring work. The ratio of students per classroom in these subjects is 10 to 15 maximum, being commonly less than 10.

For more information, see the Student Guide with information of services as:

- The educational model and ho to study at the UOC
- Library
- Online support services and UOC locations
- Support and guidance for graduates
- Services for students with disabilities
- Personal Mentoring

8. Transparency and Documentation

All three institutions will publish public information about the Joint Programme on their respective websites. This information will include: presentation and description of the programme, its curriculum, recognition of qualifications and prior learning, its faculty and management staff, a study on the job market, access requirements, enrolment and prices, and contact form. Moreover, specific information about the specificities of the programme will also be included, such as the involved universities and the main characteristics of the educational and assessment model which will be followed. As for example, this page with relevant information on the different programmes offered in English by UOC:

Study at the UOC | UOC

Moreover, as it has been already told, each academic course a Student Guide is published with a compilation of information to explain to new students how to enrol and study at UOC, the main characteristics of the university functioning and educational model, as well as the services that the university provides to them before, while and after finishing their studies.

Information about the quality of the programmes is also offered at the Quality of the degrees section of the university's web. In the same way, there is public information about the Faculty of Computer Science, Multimedia and Telecommunications.

9. Quality Assurance

The Quality Assurance System applied to guarantee the quality of this programme will be that of the UOC. In particular, the programme will deploy the Internal Quality Assurance System provided by the Universitat Oberta de Catalunya, as coordinating University, to guarantee the quality of the programme.

The Internal Quality Assurance System follows the corresponding manual, which was developed in 2009 as part of the AUDIT programme. The manual, certified by AQU Catalunya in 2009, updated in 2020 and eventually certified by AQU Catalunya in 2021, allows all the activities aimed at guaranteeing course quality, broadening stakeholder participation mechanisms and embedding the review and continuous improvement system to be compiled in one single document. The Internal Quality Assurance System has been designed in line with international quality standards. This means that it focuses on processes, in the broad sense, to accommodate and integrate any process that the UOC may identify and develop in the future.

More information about the Internal Quality Assurance System can be found at this page: Internal Quality Assurance System - Quality - UOC.

A. Annexes

A.1. Mandatory annexes

A.1.1. Documents supporting the legal status of the partner institutions

1) *Documents supporting the legal status of the partner institutions:*

UOC: Document1 and Document2

OUNL: Document 1 and document 2

UNED: Document1 and Document2

A.1.2. Cooperation agreement

2) *Cooperation agreement*
3) Addendum Mobility
4) Specific agreement in terms of mobility

A.1.3. *Documents supporting each partner's legal basis*

5) *Documents supporting each partner's legal basis for:*
    a) *Participating in the joint programme*
    b) *(Joint) degree awarding rights (if applicable)*

The information has been provided in section A1.2

A.1.4. List of intended learning outcomes

6) *List of intended learning outcomes, including:*
    a) *Matrix of alignment with Framework for Qualifications in the European Higher Education Area (FQ-EHEA)*
        i) *The learning outcomes of the Bachelor's degree can be found here.*
        ii) *The learning outcomes of the Bachelor's degree developed by each course per subject can be found here.*

iii) *The mapping between the learning outcomes of the Bachelor's degree and the learning outcomes of the subjects can be found here.*

b) *Matrix of alignment with applicable national qualifications framework. A detailed mapping between the Learning Outcomes of the Bachelor's degree and the Spanish and Dutch Qualification Frameworks can be found here. Both frameworks are aligned with the European QUalification Framework.*

	Spanish Qualification Framework *MECU/MECES*	***Dutch Qualification Framework*** *NLQF*	***European Qualification Framework*** *EQF*
Bachelor Degree 180 ECTS	*Level 6 MECU* *Level 2 MECES* *(Specific Qualification Framework for Higher Education)*	*Level 6*	*Level 6*

A.1.5. Course syllabi of all partners

7) Course syllabi of all partners

Syllabi

A.1.6. Structure of the curriculum / study plan

8) Structure of the curriculum / study plan

Find the details in section 3, study programme

A.1.7. Official documents indicating admission requirements and selection procedures

9) Official documents indicating admission requirements and selection procedures

UOC academic regulations, chapter II

A.1.8. Official documents outlining procedure for recognition of qualifications

10) Official documents outlining procedure for recognition of qualifications

UOC academic regulations, title V

A.1.9. Students' assessments regulations

11) Students' assessments regulations

UOC academic regulations, title VI

A.1.10. Academic staff CVs (all partners)

12) Academic staff CVs (all partners)

A.1.11. Relevant documents constituting internal quality assurance system

13) Relevant documents constituting internal quality assurance system

Internal quality assurance system
EUC reports system AQU
RUCT

A.1.12. Diploma supplement (sample)

14) Diploma supplement (sample)

Sample

A.2. Additional annexes

A.2.1. Activities, methodologies and assessment methods

The mapping among the activities, methodologies and assessment methods is shown in this table.

A.2.2. Internship

The degree includes the option to do an internship course during the Bacherlor's Degree. More information about methodological aspects and a list of the main companies with educational cooperation agreements with the coordinating University can be found here.

Additionally, in the context of this international Bachelor's Degree, collaborations with a wider range of companies and institutions with cooperation agreements with the participant Universities in the joint degree will also be explored.

ImpEA

Facilitating implementation of the European Approach for Quality Assurance of Joint Programmes

Erasmus+

Project leader

Project partners

Deusto

EUSKAL UNIBERTSITATE SISTEMAREN KALITATE AGENTZIA
AGENCIA DE CALIDAD DEL SISTEMA UNIVERSITARIO VASCO
UNIBASQ

Akkreditiert durch AQAS www.aqas.de

Adam Mickiewicz University Poznań

Carl von Ossietzky universität Oldenburg

ENQA
European Association for Quality Assurance in Higher Education

eca european consortium for accreditation

Warsaw School of Tourism and Hospitality Management