



POLITÉCNICA

INTERNATIONAL  
CAMPUS OF  
EXCELLENCE

COORDINATION PROCESS OF  
LEARNING ACTIVITIES  
PR/CL/001



E.T.S. de Ingeniería de  
Sistemas Informáticos

# ANX-PR/CL/001-01

## LEARNING GUIDE

### SUBJECT

**615001061 - Mobile App Development**

### DEGREE PROGRAMME

61IW - Grado En Ingeniería Del Software

### ACADEMIC YEAR & SEMESTER

2023/24 - Semester 2

## Index

---

### Learning guide

1. Description.....	1
2. Faculty.....	1
3. Prior knowledge recommended to take the subject.....	2
4. Skills and learning outcomes .....	2
5. Brief description of the subject and syllabus.....	3
6. Schedule.....	6
7. Activities and assessment criteria.....	9
8. Teaching resources.....	11
9. Other information.....	12

## 1. Description

---

### 1.1. Subject details

<b>Name of the subject</b>	615001061 - Mobile App Development
<b>No of credits</b>	3 ECTS
<b>Type</b>	Optional
<b>Academic year of the programme</b>	Fourth year
<b>Semester of tuition</b>	Semester 8
<b>Tuition period</b>	February-June
<b>Tuition languages</b>	English
<b>Degree programme</b>	61IW - Grado en Ingeniería del Software
<b>Centre</b>	61 - Escuela Técnica Superior De Ingeniería De Sistemas Informáticos
<b>Academic year</b>	2023-24

## 2. Faculty

---

### 2.1. Faculty members with subject teaching role

<b>Name and surname</b>	<b>Office/Room</b>	<b>Email</b>	<b>Tutoring hours *</b>
Bernardo Tabuenca Archilla (Subject coordinator)	4010	bernardo.tabuenca@upm.es	Sin horario. See website

\* The tutoring schedule is indicative and subject to possible changes. Please check tutoring times with the faculty member in charge.

## 3. Prior knowledge recommended to take the subject

---

### 3.1. Recommended (passed) subjects

The subject - recommended (passed), are not defined.

### 3.2. Other recommended learning outcomes

- Students should possess foundations in programming concepts, including variables, data types, control structures, functions, and basic algorithms.
- Familiarity with object-oriented programming principles, such as classes, objects, inheritance, encapsulation, and polymorphism.
- Basic understanding of UI design principles and familiarity with mobile app basics, such as screens, buttons, forms, and navigation.

## 4. Skills and learning outcomes \*

---

### 4.1. Skills to be learned

CB4 - Knowledge of the fundamentals of the use and programming of computers, operating systems, databases and, in general, software with engineering applications databases and, in general, computer programs with applications in engineering.

CC17 - Ability to design and evaluate human-computer interfaces to ensure accessibility and usability of computer systems, services and applications.

CE4 - Ability to identify and analyse problems and design, develop, implement, verify and document software solutions based on an adequate knowledge of current theories, models and techniques.

CT6 - Critical reasoning: The ability to think critically involves three things: (1) an attitude of being willing to consider in a reflective manner problems and issues that fall within the range of one's experiences, (2) knowledge of the methods of logical enquiry and reasoning, and (3) some skill in the application of those methods.

## 4.2. Learning outcomes

RA100 - Knows and manages the tools for the storage, processing and access to information systems.

RA451 - Gain the skills to manage data effectively within mobile applications

RA452 - Creates interactive user interfaces, handle user input and events, implement navigation and routing, and manage app state effectively.

RA107 - Develops, evaluates and maintains software systems that meet user requirements

RA103 - Develops user interfaces for software

RA453 - Acquire knowledge of best practices and design patterns in mobile app development.

\* The Learning Guides should reflect the Skills and Learning Outcomes in the same way as indicated in the Degree Verification Memory. For this reason, they have not been translated into English and appear in Spanish.

## 5. Brief description of the subject and syllabus

---

### 5.1. Brief description of the subject

The module on mobile app development offers students an immersive journey into the dynamic and ever-expanding field of mobile application development. In this module, students will explore the powerful combination of Flutter, a versatile and cross-platform framework, and Kotlin, a modern and expressive programming language. Through a blend of theoretical concepts, practical exercises, and hands-on projects, students will delve into the intricacies of developing mobile apps for both Android and iOS platforms.

By leveraging the Flutter framework, students will gain a comprehensive understanding of its unique widget-based architecture, allowing them to craft visually appealing and responsive user interfaces. They will learn to design layouts, incorporate interactive elements, and handle user input and events with ease. Additionally, students will explore Flutter's robust set of pre-built widgets and customizable components, enabling them to create immersive user experiences that seamlessly adapt to different device sizes and orientations.

The module will also focus on the Kotlin programming language, equipping students with the necessary skills to write clean, concise, and efficient code. They will dive into Kotlin's object-oriented nature, learning about classes, objects, inheritance, and other fundamental concepts. With this solid foundation, students will be able to implement powerful data structures, create modular code, and apply industry-standard design patterns such as MVVM or

MVC.

Furthermore, students will discover the essential aspects of mobile app development, including navigation and routing, state management, data integration from external APIs, and local data storage using databases. They will gain insights into best practices for app performance optimization, error handling, and testing, ensuring the delivery of robust and high-quality mobile applications.

Throughout the module, students will engage in hands-on projects that simulate real-world scenarios, allowing them to apply their knowledge and skills in practical contexts. They will have the opportunity to develop mobile apps from scratch, tackling challenges related to user experience, data management, and app functionality. The module encourages experimentation, creativity, and critical thinking, empowering students to become proficient mobile app developers capable of transforming their ideas into innovative and user-friendly mobile applications.

By the end of the module, students will have acquired a comprehensive skill set in mobile app development with Flutter and Kotlin. They will possess the ability to develop cross-platform mobile apps, understand the nuances of user interface design, effectively manage app state, and integrate data from various sources. With this expertise, they will be well-prepared to embark on a rewarding career in the rapidly evolving world of mobile app development or continue their exploration of advanced topics in the field.

## 5.2. Syllabus

1. Introduction to Flutter and User Interfaces
  - 1.1. Setting up the Development Environment
  - 1.2. Flutter Basics: Widgets and Layouts
  - 1.3. Building User Interfaces with Flutter
2. Navigation and State Management in Flutter
  - 2.1. Handling User Input and Events in Flutter
  - 2.2. Implementing Navigation and Routing in Flutter
  - 2.3. State Management in Flutter Apps
3. Data Integration and UI Design in Flutter
  - 3.1. Consuming APIs in Flutter Apps
  - 3.2. Handling Asynchronous Operations in Flutter
  - 3.3. Advanced UI Design in Flutter: Styling and Theming
4. Advanced Topics in Flutter and Deployment

- 4.1. Animations and Gestures in Flutter
- 4.2. Local Data Storage in Flutter: Working with Databases
- 4.3. Error Handling and Validation in Flutter Apps
- 4.4. Deploying Flutter Apps: App Store and Google Play Store Guidelines
- 5. Introduction to Kotlin and Basic Concepts
  - 5.1. Setting up the Development Environment
  - 5.2. Introduction to Kotlin Programming Language
  - 5.3. Variables, Data Types, and Control Flow in Kotlin
- 6. Object-Oriented Programming in Kotlin
  - 6.1. Classes and Objects
  - 6.2. Inheritance and Polymorphism
- 7. Advanced Topics in Kotlin
  - 7.1. Extensions and Lambdas
  - 7.2. Coroutines for Asynchronous Programming
- 8. Advanced State Management and Firebase Integration in Kotlin
  - 8.1. Advanced State Management in Flutter: Provider or Riverpod
  - 8.2. Advanced Widget Composition in Flutter
  - 8.3. Firebase Integration in Flutter Apps
  - 8.4. Real-time Data Sync with Firebase Cloud Firestore

## 6. Schedule

### 6.1. Subject schedule\*

Week	Classroom activities	Laboratory activities	Distant / On-line	Assessment activities
1		<b>Introduction to Flutter and User Interfaces</b> Duration: 01:00 Lecture  <b>Introduction to Flutter and User Interfaces</b> Duration: 01:00 Laboratory assignments		
2		<b>Navigation and State Management in Flutter</b> Duration: 01:00 Lecture  <b>Navigation and State Management in Flutter</b> Duration: 01:00 Laboratory assignments		
3		<b>Data Integration in Flutter</b> Duration: 01:00 Lecture  <b>Data Integration in Flutter</b> Duration: 01:00 Laboratory assignments		
4		<b>Asynchronous Operations and UI Design in Flutter</b> Duration: 01:00 Lecture  <b>Asynchronous Operations and UI Design in Flutter</b> Duration: 01:00 Laboratory assignments		
5		<b>Animations and Gestures in Flutter</b> Duration: 01:00 Lecture  <b>Animations and Gestures in Flutter</b> Duration: 01:00 Laboratory assignments		
6		<b>Local Data Storage and Error Handling in Flutter</b> Duration: 01:00 Lecture  <b>Local Data Storage and Error Handling in Flutter</b> Duration: 01:00 Laboratory assignments		



7		<p><b>Testing and Debugging Flutter Apps</b> Duration: 01:00 Lecture</p> <p><b>Testing and Debugging Flutter Apps</b> Duration: 01:00 Laboratory assignments</p>		
8		<p><b>Advanced State Management and Deployment in Flutter</b> Duration: 01:30 Laboratory assignments</p>		<p><b>Flutter test</b> Online test Continuous assessment Presential Duration: 00:30</p> <p><b>Flutter project</b> Group work Continuous assessment Presential Duration: 00:00</p>
9		<p><b>Introduction to Kotlin and Basic Concepts</b> Duration: 01:00 Lecture</p> <p><b>Introduction to Kotlin and Basic Concepts</b> Duration: 01:00 Laboratory assignments</p>		
10		<p><b>Object-Oriented Programming in Kotlin</b> Duration: 01:00 Lecture</p> <p><b>Object-Oriented Programming in Kotlin</b> Duration: 01:00 Laboratory assignments</p>		
11		<p><b>Advanced Topics in Kotlin: Extensions and Lambdas</b> Duration: 01:00 Lecture</p> <p><b>Advanced Topics in Kotlin: Extensions and Lambdas</b> Duration: 01:00 Laboratory assignments</p>		
12		<p><b>Kotlin Coroutines for Asynchronous Programming</b> Duration: 01:00 Lecture</p> <p><b>Kotlin Coroutines for Asynchronous Programming</b> Duration: 01:00 Laboratory assignments</p>		
13		<p><b>Advanced State Management in Kotlin</b> Duration: 01:00 Lecture</p> <p><b>Advanced State Management in Kotlin</b> Duration: 01:00 Laboratory assignments</p>		

14		<b>Firestore Integration in Kotlin</b> Duration: 01:00 Lecture  <b>Firestore Integration in Kotlin</b> Duration: 01:00 Laboratory assignments		
15		<b>Real-time Data Sync with Firestore</b> Duration: 01:00 Laboratory assignments  <b>Advanced Topics and Project Integration in Kotlin</b> Duration: 00:30 Laboratory assignments		<b>Kotlin project</b> Group work Continuous assessment Presential Duration: 00:00  <b>Kotlin test</b> Online test Continuous assessment Presential Duration: 00:30
16				
17				<b>Flutter project</b> Group work Final examination Presential Duration: 00:00  <b>Kotlin project</b> Group work Final examination Presential Duration: 00:00  <b>Flutter test</b> Online test Final examination Presential Duration: 01:00  <b>Kotlin test</b> Online test Final examination Presential Duration: 01:00

Depending on the programme study plan, total values will be calculated according to the ECTS credit unit as 26/27 hours of student face-to-face contact and independent study time.

\* The schedule is based on an a priori planning of the subject; it might be modified during the academic year, especially considering the COVID19 evolution.

## 7. Activities and assessment criteria

### 7.1. Assessment activities

#### 7.1.1. Assessment

Week	Description	Modality	Type	Duration	Weight	Minimum grade	Evaluated skills
8	Flutter test	Online test	Face-to-face	00:30	15%	5 / 10	CC17 CE4 CB4
8	Flutter project	Group work	Face-to-face	00:00	35%	5 / 10	CB4 CC17 CE4 CT6
15	Kotlin project	Group work	Face-to-face	00:00	35%	5 / 10	CB4 CC17 CE4 CT6
15	Kotlin test	Online test	Face-to-face	00:30	15%	5 / 10	CB4 CE4 CC17

#### 7.1.2. Global examination

Week	Description	Modality	Type	Duration	Weight	Minimum grade	Evaluated skills
17	Flutter project	Group work	Face-to-face	00:00	35%	5 / 10	CE4 CB4 CC17 CT6
17	Kotlin project	Group work	Face-to-face	00:00	35%	5 / 10	CB4 CC17 CE4 CT6
17	Flutter test	Online test	Face-to-face	01:00	15%	5 / 10	CB4 CC17 CE4
17	Kotlin test	Online test	Face-to-face	01:00	15%	5 / 10	CB4 CC17 CE4

#### 7.1.3. Referred (re-sit) examination

Description	Modality	Type	Duration	Weight	Minimum grade	Evaluated skills
Flutter test	Online test	Face-to-face	01:00	15%	5 / 10	CB4 CC17 CE4
Kotlin test	Online test	Face-to-face	01:00	15%	5 / 10	CE4 CB4 CC17
Flutter project	Group work	Face-to-face	01:00	35%	5 / 10	CB4 CC17 CE4 CT6
Kotlin project	Group work	Face-to-face	01:00	35%	5 / 10	CE4 CT6 CB4 CC17

## 7.2. Assessment criteria

The evaluation is distributed in two parts that will evenly evaluate the learning goals of the subject (RA451, RA452, RA453, RA100, RA103, y RA107):

### Project Development (70%)

- Mobile Application Projects: Students will work in groups of two to develop two different mobile applications throughout the semester.
- Innovative and Goal oriented: Evaluation will consider the originality and goal-oriented aspects of the app ideas and features.
- Functionality and User Experience: Assessment will focus on the functionality, usability, and overall user experience of the developed applications.
- Implementation of Concepts: Students' ability to effectively apply the concepts learned in the module, such as user interface design, navigation, data integration, state management, and error handling, will be evaluated.
- Documentation Quality: Evaluation will consider the completeness, clarity, and organisation of the project documentation, including project plans, design documents, and user guides.
- Group Collaboration: Evaluation will assess the level of collaboration, cooperation, and effective teamwork

demonstrated by the students throughout the project development.

- Task Allocation: Assessment will consider the fair distribution of tasks, responsibilities, and contributions within the group.

### Online Test (30%)

- Knowledge Assessment: An online test will be conducted to evaluate students' understanding and comprehension of key concepts addressed along the semester, including Flutter and Kotlin fundamentals, UI design, state management, data integration, and asynchronous programming.

The evaluation in both ordinary and extraordinary periods will be evaluated following the same procedures.

## 8. Teaching resources

---

### 8.1. Teaching resources for the subject

Name	Type	Notes
Computers	Equipment	
Flutter tutorials	Web resource	<a href="https://docs.flutter.dev/reference/tutorials">https://docs.flutter.dev/reference/tutorials</a>
Get started with Kotlin	Web resource	<a href="https://kotlinlang.org/docs/home.html">https://kotlinlang.org/docs/home.html</a>
Mobile application development slides	Bibliography	Documentation of the subject
Flutter for beginners	Bibliography	Author: Thomas Bailey, Alessandro Biessek, Trevor Wills  Published by Packt Publishing  ISBN 978-1-80056-599-9

Android Programming with Kotlin for Beginners	Bibliography	By John Horton <a href="https://learning.oreilly.com/library/view/android-programming-with/9781789615401/">https://learning.oreilly.com/library/view/android-programming-with/9781789615401/</a>
---	--------------	---

## 9. Other information

---

### 9.1. Other information about the subject

Students will be invited to implement projects that address Sustainable Development Goals of the UNESCO Agenda 2030. More specifically, addressing goals 4, 10, and 11 will be encouraged.