

Caracterização da Unidade Curricular / Characterization of the Curricular Unit

Designação da Unidade Curricular (UC) / Title of Curricular Unit (CU): Desenvolvimento de Aplicações Android / Android Application Development

Área científica da UC / CU Scientific Area: Ciências Informáticas / Computer Science

Semestre / Semester: 3º

Número de créditos ECTS / Number of ECTS credits: 6

Carga horária por tipologia de horas / Workload by type of hours: TP: 45; OT: 6; O: 9

Carga letiva semanal / Weekly letive charge: 3h

Objetivos de aprendizagem (conhecimentos, aptidões e competências a desenvolver pelos estudantes):

Ao completar esta unidade curricular, o aluno será capaz de desenvolver aplicações Android funcionais, utilizando linguagens como Java e Kotlin. Irá compreender a arquitetura do sistema Android, criar interfaces intuitivas, manipular dados e recursos do dispositivo. Além disso, estará preparado para publicar e distribuir aplicações no mercado, alinhado às melhores práticas da indústria.

Intended learning outcomes (knowledge, skills and competences to be developed by the students):

Upon completion of this course, students will be able to develop functional Android applications using languages such as Java and Kotlin. You will understand the Android system architecture, create intuitive interfaces, and manipulate device data and resources. Furthermore, you will be prepared to publish and distribute applications in the market, aligned with industry best practices.

Conteúdos programáticos:

1. Fundamentos e Introdução

- 1.1 Conceitos aplicados ao desenvolvimento de aplicações móveis e ambientes Mobile.
- 1.2 História e evolução dos dispositivos móveis e do sistema operativo Android.
- 1.3 Arquitetura do sistema operativo Android e suas APIs principais.

2. Ambiente de Desenvolvimento

- 2.1 Instalação e configuração do Android Studio.
- 2.2 Criação de projetos e utilização de emuladores.
- 2.3 Estrutura básica de um projeto Android.

3. Programação e Interfaces

- 3.1 Linguagem Java e Kotlin para Android.
- 3.2 Criação de layouts utilizando XML e ferramentas visuais.
- 3.3 Componentes visuais (Widgets, Views, ViewGroup).
- 3.4 Tipos de layouts (LinearLayout, RelativeLayout, ConstraintLayout, etc.).

4. Navegação e Usabilidade

- 4.1 Navegação entre ecrãs (Activities, Fragments).
- 4.2 Boas práticas de design de interface (Material Design).
- 4.3 Responsividade e adaptação a diferentes tamanhos de ecrã.

5. Persistência de Dados

- 5.1 Armazenamento local: SharedPreferences, ficheiros, base de dados SQLite.
- 5.2 Integração com APIs REST e pedidos HTTP.
- 5.3 Manipulação de dados e CRUD (Create, Read, Update, Delete).

6. Recursos Avançados

- 6.1 Utilização de recursos do dispositivo (GPS, câmara, sensores).
- 6.2 Notificações e serviços em segundo plano.
- 6.3 Integração com fornecedores de conteúdos (contactos, agenda).

7. Publicação e Distribuição

- 7.1 Preparação da aplicação para publicação (Google Play Store).
- 7.2 Processo de submissão e monetização.

8. Projetos Práticos

- 8.1 Desenvolvimento de projetos completos (ex: aplicação para registo, CRUD, formulários).
- 8.2 Testes em diferentes dispositivos e emuladores.

Syllabus:

1. Fundamentals and Introduction

- 1.1 Concepts applied to the development of mobile applications and mobile environments.
- 1.2 History and evolution of mobile devices and the Android operating system.
- 1.3 Architecture of the Android operating system and its main APIs.

2. Development Environment

- 2.1 Installation and configuration of Android Studio.
- 2.2 Creating projects and using emulators.
- 2.3 Basic structure of an Android project.

3. Programming and Interfaces

- 3.1 Java and Kotlin languages for Android.
- 3.2 Creating layouts using XML and visual tools.
- 3.3 Visual components (Widgets, Views, ViewGroup).
- 3.4 Types of layouts (LinearLayout, RelativeLayout, ConstraintLayout, etc.).

4. Navigation and Usability

- 4.1 Navigation between screens (Activities, Fragments).
- 4.2 Good interface design practices (Material Design).
- 4.3 Responsiveness and adaptation to different screen sizes.

5. Data Persistence

- 5.1 Local storage: SharedPreferences, files, SQLite database.
- 5.2 Integration with REST APIs and HTTP requests.
- 5.3 Data manipulation and CRUD (Create, Read, Update, Delete).

6. Advanced Features

- 6.1 Use of device resources (GPS, camera, sensors).
- 6.2 Notifications and background services.
- 6.3 Integration with content providers (contacts, calendar).

7. Publication and Distribution

- 7.1 Preparing the application for publication (Google Play Store).
- 7.2 Submission and monetization process.

8. Practical Projects

- 8.1 Development of complete projects (e.g., registration application, CRUD, forms).
- 8.2 Testing on different devices and emulators.