

Caracterização da Unidade Curricular / Characterization of the Curricular Unit

Designação da Unidade Curricular (UC) / Title of Curricular Unit (CU): Computação Móvel / Mobile Computing

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

Semestre / Semester: 5º

Número de créditos ECTS / Number of ECTS credits: 6

Carga horária por tipologia de horas / Workload by type of hours: TP: 22,5; PL: 22,5; OT: 6; O: 9

Carga letiva semanal / Weekly letive charge: 3h

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

A unidade curricular pretende que o aluno ao longo do semestre desenvolva competências na área da análise, síntese e projeto de aplicações móveis. O aluno deverá desenvolver competências de especificação técnica, desenvolvimento de projeto e teste do protótipo.

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

The curricular unit intends that the student, throughout the semester, develops skills in the area of analysis, synthesis and design of mobile applications. The student should develop technical specification, project development and prototype testing skills.

Conteúdos programáticos:

1. FLUTTER (contextualização)

- Breve história do FLUTTER
- Porquê programar em FLUTTER
- Principais componentes do FLUTTER
- Linguagem DART
- Arquitetura de uma solução FLUTTER
- *Widgets* e árvore de *widgets*
- Versões do FLUTTER
- Alternativas do FLUTTER

2. Setup do ambiente de desenvolvimento

- Instalação / configuração do Android Studio
- Instalação / configuração do Visual Studio Code
- Configuração do emulador Android
- Depuração num dispositivo Android Real

- Constrangimentos à depuração em iOS
- Aplicação *Hello World* em FLUTTER

3. Flutter (introdução)

- *Flutter Material Design* (apresentação)
- Estrutura de um projeto FLUTTER
- Repositórios de código GIT
- Ficheiro *main.dart*
- Tipos de dados em DART
- Operadores (e.g. aritméticos, lógicos)
- Revisão de conceitos Programação Orientada a Objetos (POO) (classes, objetos, herança, exceções, etc.)
- Construtores nomeados, construtores imutáveis, e parâmetros nomeados
- *Null Safety* do DART
- *Spread Operator* do DART
- *DART Iterables*
- *Map()* collection vs. *map()* method
- Convenções do DART
- Exercícios

4. FLUTTER (desenvolvimento)

- Criação de vários projetos em conjunto com os alunos para que aprendam a desenvolver, compilar e depurar, aplicações em FLUTTER
- Exercícios

Syllabus:

1. FLUTTER (context)

- A brief history of FLUTTER
- Why to develop in FLUTTER
- FLUTTER's main components
- DART language
- Architecture of a FLUTTER solution
- Widgets and Widget Tree
- FLUTTER versions
- FLUTTER alternatives

2. Setup of the development environment

- Android Studio installation / configuration
- Visual Studio Code installation / configuration

- Android Emulator setup
- Debugging on a Real Android device
- Debugging constraints on iOS
- Hello World application in FLUTTER~

3. FLUTTER (introduction)

- Flutter Material Design (introduction)
- Structure of a FLUTTER project
- GIT code repositories
- Main.dart file
- Data types in DART
- Operators (e.g. arithmetic, logical)
- Object-Oriented Programming (OOP) concepts review (classes, objects, inheritance, exceptions, etc.)
- Named constructors, immutable constructors, and named parameters
- DART's Null Safety
- DART Spread Operator
- DART Iterables
- Map() collection vs. map() method
- DART Conventions
- Exercises

4. FLUTTER (development)

- Creating several projects together with the students to learn how to develop, compile and debug FLUTTER applications
 - Exercises
-