**Arquiteturas Móveis**

A red text on a black background

Description automatically generated

**Relatório Trabalho Prático – Flutter**

Realizado por:

Diogo Oliveira – 2021146037

Lara Bizarro – 2021130066

Tomás Laranjeira – 2021135060

Índice

[**1.** **Introdução** 3](#_Toc186997504)

[**2.** **Ecrãs** 4](#_Toc186997505)

[**2.1.** **main** 4](#_Toc186997506)

[**2.2.** **add\_contact\_screen** 5](#_Toc186997507)

[**2.3.** **contact\_details\_screen** 6](#_Toc186997508)

[**2.4.** **edit\_contact\_screen** 7](#_Toc186997509)

[**2.5.** **list\_contact\_screen** 8](#_Toc186997510)

[**3.** **Class Contact** 9](#_Toc186997511)

[**4.** **Decisões tomadas** 10](#_Toc186997512)

[**4.1.** **Persistência de dados** 10](#_Toc186997513)

[**4.2.** **Histórico de contactos modificados** 10](#_Toc186997519)

[**4.3.** **Obtenção de localização** 10](#_Toc186997526)

[**4.4.** **Visualização do mapa** 10](#_Toc186997527)

# **Introdução**

O presente relatório descreve o desenvolvimento de uma aplicação móvel em Flutter, que tem como objetivo a gestão de contactos pessoais, oferecendo funcionalidades para armazenamento e consulta de informações básicas, como nome, e-mail, telefone e data de nascimento.

A aplicação permite ao utilizador associar imagens aos contactos, seja a partir da galeria do dispositivo ou por meio da câmara, bem como registar e consultar as localizações geográficas dos encontros com cada contacto.

Adicionalmente, inclui-se a funcionalidade de manter um histórico autónomo dos 10 últimos contactos modificados ou com novas localizações adicionadas, promovendo uma gestão eficiente e organizada.

Esta aplicação é uma adaptação da versão originalmente desenvolvida em Kotlin durante as aulas práticas.

# **Ecrãs**

## **main**

Este ecrã mostra uma lista de todos os contactos guardados. Ao clicar no icon de lista, aparecemmais informações dos contactos. É também a partir deste ecrã que se adiciona um novo contact ou se vê o histórico dos ultimos contactos modificados.

A screenshot of a phone

Description automatically generated A screenshot of a phone

Description automatically generated

Figura 1- main

## **add\_contact\_screen**

Este ecrã apresenta os utilizadores com os vários campos a preencher para criar um novo contacto. Os campos *Name*, *Phone Number* e *Email* são de preenchimento obrigatório enquanto que a seleção de data de nascimento e de imagem são opcionais.

A screenshot of a computer

Description automatically generated

Figura 2 - add\_contact\_screen

## **contact\_details\_screen**

Este ecrã mostra toda a informação relativa a um contacto, incluindo um mapa com marcadores nas localizações em que o utilizador se encontrou com este contacto, para guardar a localização atual basta clicar no icone de marcador no canto superior direito.

Caso o utilizador queira editar alguma informação relativa ao contacto basta clicar no icone de edição no canto superior direito o que o levará para um diferente ecrã.

A screenshot of a phone

Description automatically generated

Figura 3 - contact\_details\_screen

## **edit\_contact\_screen**

Este ecrã é semelhante ao *add\_contact\_screen*, no entanto os campos já aparecem preenchidos com a informação pré-existente.

A screenshot of a phone

Description automatically generated

Figura 4 - edit\_contact\_screen

## **list\_contact\_screen**

Este ecrã mostra uma lista dos ultimos contactos que foram modificados. Quer seja a alteração de um campo ou a inserção de uma nova localização.

A black rectangular object with a white background

Description automatically generated

Figura 5 - list\_contact\_screen

# **Class Contact**

Optámos por criar um ficheiro *contact.*dart apenas com a class Contact de forma a ser mais fácil a sua utilização em vários outros ficheiros.

Esta classe tem variáveis em que guarda as informações de cada contacto anteriormente referidas. Existem támbem métodos *toJson* e *fromJson* para auxiliar a guardar e obter os dados quando a aplicação é encerrada.

A screen shot of a computer program

Description automatically generated

Figura 6 - Class Contact

# **Decisões tomadas**

## **Persistência de dados**

Para o armazenamento de dados dos contactos optámos pela utilização de um ficheiro de formato json guardado na diretoria temporária do android.



## **Histórico de contactos modificados**

Como indicado no enunciado para a criação do histórico de contactos modificados foi utilizada a biblioteca *shared\_preferences*.



## **Obtenção de localização**

Apesar do enunciado pedir a utilização da biblioteca *location* para a obtenção de localização, durante as aulas práticas o docente informou que também seria aceitado o uso da biblioteca *geolocator*, por isso optámos pela utilização da alternativa.

## **Visualização do mapa**

Inicialmente optámos pela utilização da biblioteca *google\_maps\_flutter*, no entanto tivémos dificuldades na implementação o que nos levou à utilização da biblioteca *flutter\_map*.

# **Conclusão**

O desenvolvimento da aplicação "Contacts" em Flutter permitiu consolidar os conceitos abordados nas aulas práticas, adaptando uma solução inicial em Kotlin para Flutter. A aplicação criada cumpre os requisitos estabelecidos, proporcionando ao utilizador uma interface intuitiva e funcionalidades avançadas, como a gestão de contactos com integração de localização e suporte de imagens.

Um dos principais desafios enfrentados durante o desenvolvimento foi a apresentação do mapa na aplicação. Apesar da liberdade na escolha das bibliotecas para esta funcionalidade, ajustar as marcações das localizações e garantir uma integração fluida com a interface revelou-se mais complexo do que o previsto.