



Arquiteturas Móveis

Trabalho Prático 2

Docente: Álvaro Nuno Santos

| | | |
|-----------------|---|------------|
| Gonçalo Leite | - | 2018014643 |
| Gonçalo Pereira | - | 2019139650 |
| Inês Estevão | - | 2011016171 |

Índice

| | | |
|----|---------------------|---|
| 1. | Introdução | 3 |
| 2. | Implementação | 4 |

1. Introdução

Este trabalho surge no âmbito da Unidade Curricular de Arquiteturas Móveis, no 1º Semestre do ano letivo de 2023/2024. A sua implementação foi feita recorrendo a Flutter.

Este trabalho tem como foco desenvolver uma aplicação Android que facilite as visitas turísticas a diversas localizações, disponibilizando informação sobre os locais de interesse existentes nesses locais, podendo também consultar locais de interesse, classificando de forma anónima com Gosto ou Não Gosto. É composto por duas componentes essenciais: os Locais e os Locais de Interesse.

No âmbito da gestão e persistência de dados, utilizamos uma base de dados FireBase, consolidando uma abordagem eficiente e confiável.

Ao longo deste trabalho, exploraremos detalhadamente cada componente do sistema, analisando sua importância e função específica.

2. Implementação

A aplicação fornece informações sobre os locais e os pontos de interesse, podendo classificar esses pontos por 'Gosto' ou 'Não Gosto'.

A implementação foi codificada da seguinte forma:

- A aplicação integra-se ao FireBase para armazenar e recuperar dados relacionar aos locais, pontos de interesse e avaliações. Utilizamos a biblioteca `shared_preferences` para armazenar as classificações atribuídas apenas aos pontos de interesse garantindo que cada classificação seja registrada apenas uma vez. utilizando também o método '`setState()`' para atualizar a interface do usuário em resposta à mudança de estado.

```
void _handleLikeDislike(bool isLike) async {  
  SharedPreferences prefs = await SharedPreferences.getInstance();  
  
  if (isLike) {  
    prefs.setBool('${widget.pointOfInterest.id}_isLiked', true);  
    prefs.setBool('${widget.pointOfInterest.id}_isDisliked', false);  
  } else {  
  
    prefs.setBool('${widget.pointOfInterest.id}_isLiked', false);  
    prefs.setBool('${widget.pointOfInterest.id}_isDisliked', true);  
  }  
  
  setState(() {  
    _loadLikeDislikeStatus();  
  });  
}
```

```
void _loadLikeDislikeStatus() async {  
  SharedPreferences prefs = await SharedPreferences.getInstance();  
  setState(() {  
    isLiked = prefs.getBool('${widget.pointOfInterest.id}_isLiked') ?? false;  
    isDisliked = prefs.getBool('${widget.pointOfInterest.id}_isDisliked') ?? false;  
  });  
}
```

O ‘_handleLikeDislike()’ trata da lógica do ‘Gosto’ e do não ‘Gosto’, armazenando depois no ‘shared_preferences’. O ‘_loadLikeDislikeStatus()’ é usado para tratar do estado atual do ‘Gosto’ e do não ‘Gosto’ quando é inicializado a aplicação.