

**Instituto Politécnico de Coimbra**  
***Instituto Superior de Engenharia de Coimbra***

## **Arquiteturas Móveis**

### **Trabalho Prático 2 - Flutter**

**Docente: Álvaro Santos**

Marco António de Jesus Coelho – 2018012765 – LEI

Pedro Jorge Fernandes Morais – 2018020733 – LEI

sábado, 7 de janeiro de 2023

# Índice

<b>1</b>	<b>ECRÃS.....</b>	<b>1</b>
1.1	ECRÃ PRINCIPAL .....	1
1.2	ECRÃ DE EDIÇÃO .....	2
1.3	CÂMARA .....	3
<b>2</b>	<b>ANIMAÇÕES.....</b>	<b>4</b>
2.1	ATUALIZAR DADOS .....	4
2.2	TRANSIÇÃO DE ECRÃS .....	5
<b>3</b>	<b>CLASSES.....</b>	<b>6</b>
3.1	CONSTANTS .....	6
3.2	EMENTA .....	7
3.3	DIASEMANA.....	8
3.4	ARGUMENTOSEEDITSCREEN.....	8
<b>4</b>	<b>JSON .....</b>	<b>9</b>
4.1	DIASEMANA.....	9
4.2	EMENTA .....	10
<b>5</b>	<b>SHARED PREFERENCES.....</b>	<b>11</b>
5.1	GUARDAR DADOS.....	11
5.2	LER DADOS .....	11
<b>6</b>	<b>CÂMARA .....</b>	<b>12</b>
<b>7</b>	<b>PERMISSÕES.....</b>	<b>13</b>
7.1	CÂMARA .....	13
7.2	LOCALIZAÇÃO.....	14
<b>8</b>	<b>LOCALIZAÇÃO .....</b>	<b>15</b>
<b>9</b>	<b>REFERÊNCIAS .....</b>	<b>16</b>

# Índice de Figuras

Figura 1 - Ecrã Principal, Sem dados locais .....	1
Figura 2 - Ecrã Principal, Dados originais menu .....	1
Figura 3 – Ecrã Principal, Dados editados menu .....	1
Figura 4 – Ecrã Edição, Dados originais.....	2
Figura 5 - Ecrã Edição, Sem alterações .....	2
Figura 6 - Ecrã Edição, Alteração dados.....	2
Figura 7 - Ecrã Edição, Com alterações.....	2
Figura 8 - Ecrã Câmara .....	3
Figura 9 – Atualizar dados, Antes de atualizar .....	4
Figura 10 - Atualizar dados, Durante atualização.....	4
Figura 11 - Transição de ecrãs, principal e edição.....	5
Figura 12 - Transição de ecrãs, edição e câmara .....	5
Figura 13 - Classes, <i>Constants</i> .....	6
Figura 14 - Classes, <i>Ementa</i> .....	7
Figura 15 - Classes, <i>DiaSemana</i> .....	8
Figura 16 - Classes, <i>ArgumentosEditScreen</i> .....	8
Figura 17 - JSON, Conversão da classe <i>DiaSemana</i> .....	9
Figura 18 - JSON, Conversão da classe <i>Ementa</i> .....	10
Figura 19 - Shared preferences, Guardar dados .....	11
Figura 20 - Shared preferences, Ler dados .....	11
Figura 21 - Câmara, Ecrã de câmara.....	12
Figura 22 - Permissões Câmara, Tirar fotografia e gravar vídeo.....	13
Figura 23 - Permissões Câmara, gravar áudio .....	13
Figura 24 - Permissão localização .....	14
Figura 25 - Localização, Serviço não disponível.....	15
Figura 26 - Localização, Sem permissões.....	15
Figura 27 - Localização, Muito longe do ISEC .....	15



# 1 Ecrãs

No decorrer do desenvolvimento do presente trabalho prático tivemos a necessidade de criar três ecrãs, uma para visualizar as ementas da semana, um que permite editar os dados de uma ementa de um certo dia e um outro que permite tirar uma fotografia de forma a alterar a imagem de uma ementa.

## 1.1 Ecrã Principal

O ecrã principal é o responsável por mostrar a informação das ementas para os vários dias da semana ao utilizador. Caso a aplicação, ao arrancar, não tenha dados armazenados localmente é mostrada uma mensagem ao utilizador a informar sobre essa situação. As ementas para cada dia mostram as informações dos diversos pratos e ainda uma imagem que represente a ementa, se esta existir.



Figura 1 - Ecrã Principal,  
Sem dados locais

Figura 2 - Ecrã Principal,  
Dados originais menu

Figura 3 – Ecrã Principal,  
Dados editados menu

Os dados que forem atualizados aparecerão a sublinhado, de forma ao utilizador conseguir facilmente identificar quais são os dados originais e quais os atualizados.

## 1.2 Ecrã de Edição

No ecrã de edição é possível corrigir ou alterar os dados de uma ementa para um determinado dia da semana, sendo possível alterar os dados originais ou caso estes já tenham sido alterados anteriormente, voltar à informação original. É ainda possível adicionar uma imagem a esta ementa, ou caso esta exista alterá-la, tendo o utilizador de tirar uma fotografia.

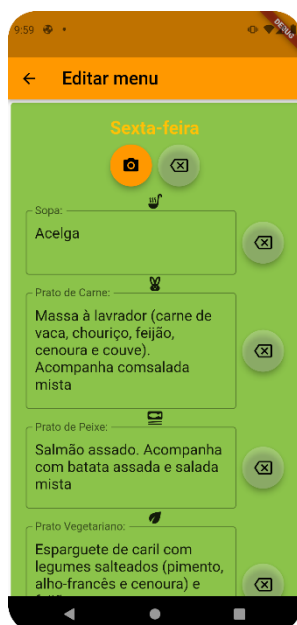


Figura 4 – Ecrã Edição,  
Dados originais

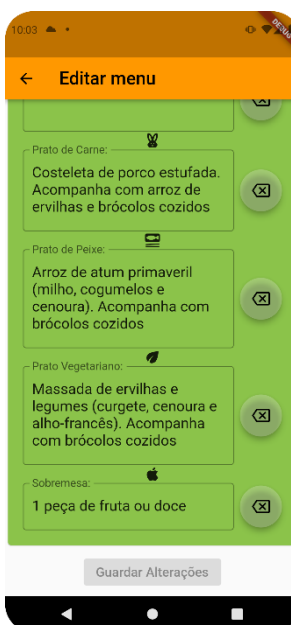


Figura 5 – Ecrã Edição,  
Sem alterações

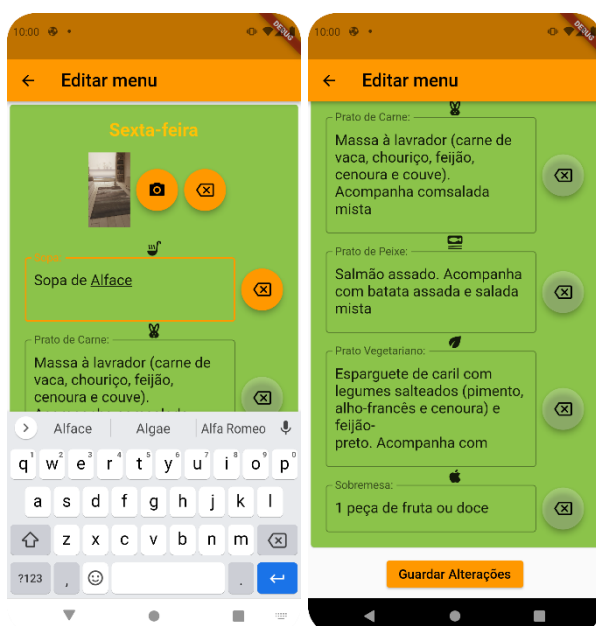


Figura 6 - Ecrã Edição,  
Alteração dados

Figura 7 - Ecrã Edição,  
Com alterações

Ao alterarmos os dados originais num campo, podemos verificar que o botão que fica ao lado desse campo passa a ativo, indicando que os dados do campo não são os originais e, se carregarmos nesse botão voltaremos a ter os dados originais.

## 1.3 Câmara

O ecrã da câmara permite ao utilizador tirar uma fotografia que representa um menu. É possível utilizar a câmara traseira ou a frontal do dispositivo.



Figura 8 - Ecrã Câmara

## 2 Animações

Durante a execução da aplicação é possível verificar algumas animações, nomeadamente a alteração do tamanho do logo da aplicação quando é realizado o pedido de atualização dos menus.

Existem também animações quando se passa de um ecrã para o outro, sendo a transição do ecrã principal para o ecrã de edição realizada com um slide da direita para a esquerda, e a transição do ecrã de edição para o ecrã da câmara feita com um slide de baixo para cima.

### 2.1 Atualizar dados

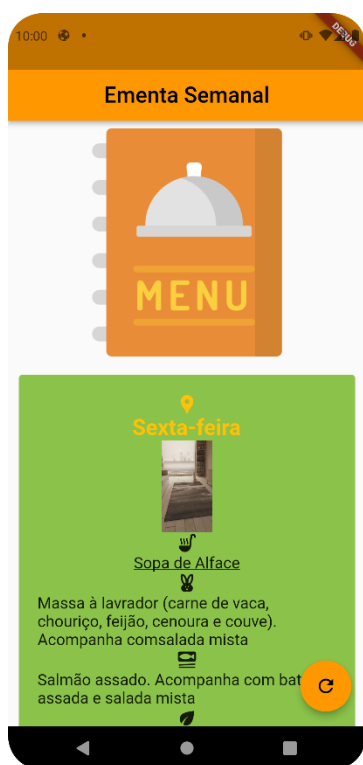


Figura 9 – Atualizar dados,  
Antes de atualizar



Figura 10 - Atualizar dados,  
Durante atualização

Quando fazemos a atualização dos dados podemos verificar que o botão de atualização é substituído por um indicador de progresso circular, e o logotipo reduz o seu tamanho. O tamanho do logotipo retornará ao normal quando os dados forem recebidos.



## 2.2 Transição de ecrãs

Na transição entre os ecrãs existem animações, entre o ecrã principal e o ecrã de edição é feita a transição através de um slide da direita para a esquerda, e entre o ecrã de edição e o da câmara existe um slide de baixo para cima.

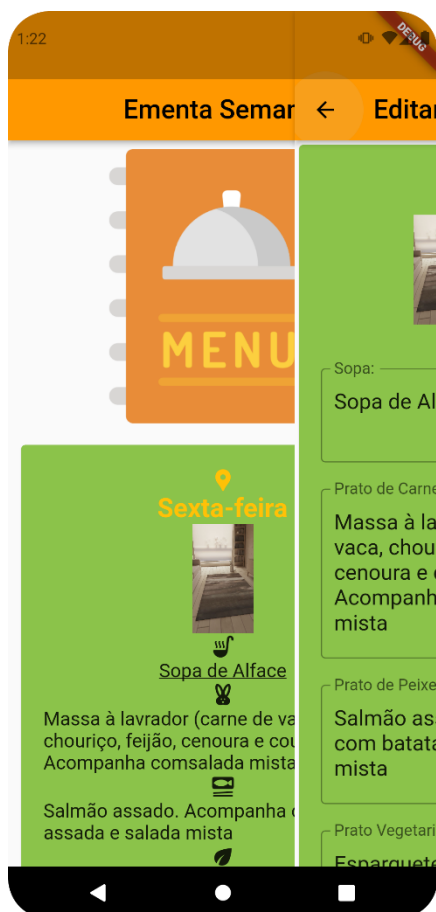


Figura 11 - Transição de ecrãs, principal e edição

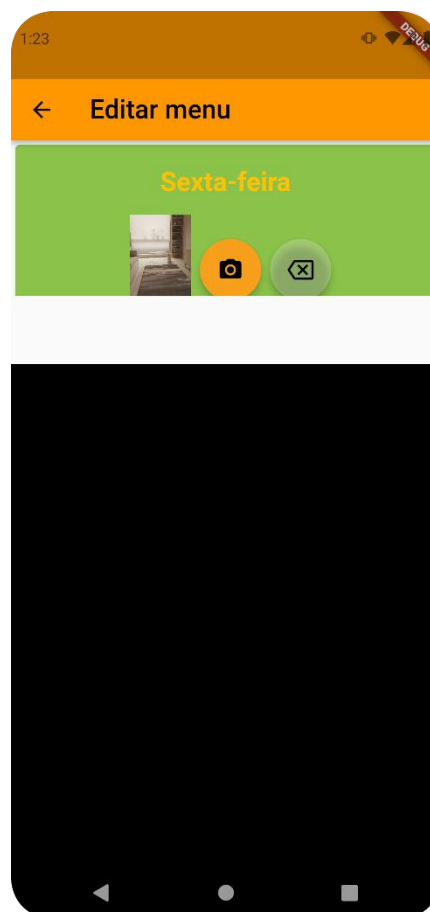


Figura 12 - Transição de ecrãs, edição e câmara

## 3 Classes

No decorrer do desenvolvimento do programa sentimos a necessidade de utilizar quatro classes auxiliares, sendo elas:

### 3.1 Constants

A classe *Constants* contém algumas variáveis estáticas, sendo que estas permitem fazer a tradução do nome dos dias da semana para português, converter um número entre 1 e 5 para o dia da semana correspondente, guardam ainda o IP e Porto para o servidor Docker, URL's utilizados para aceder às ementas e obter a imagem para uma ementa, a chave utilizada no *shared preferences* para guardar as ementas, bem como as coordenadas do ISEC com o intuito de verificar se o utilizador se encontra muito longe ou não para poder editar uma ementa.

```
6  class Constants {
7      static const Map<int, String> diasSemana = {
8          1: 'MONDAY',
9          2: 'TUESDAY',
10         3: 'WEDNESDAY',
11         4: 'THURSDAY',
12         5: 'FRIDAY'
13     };
14
15     static const Map<int, String> diasSemanaPortugues = {
16         1: 'Segunda-feira',
17         2: 'Terça-feira',
18         3: 'Quarta-feira',
19         4: 'Quinta-feira',
20         5: 'Sexta-feira'
21     };
22
23     static const String server = '192.168.1.40:8080';
24     static const String ementaMenuUrl = 'http://$server/menu';
25     static const String ementaImageUrl = 'http://$server/images/';
26     static const String sharedPreferencesValueName = 'diasSemanaEmenta';
27     static const double latitudeIsec = 40.19272104727294;
28     static const double longitudeIsec = -8.411595263284767;
29 }
```

Figura 13 - Classes, *Constants*

## 3.2 Ementa

A classe Ementa representa a ementa para um determinado dia da semana, possuindo métodos que permitem transformar uma instância da classe para JSON ou então converter de JSON para uma instância da classe. Para além das informações recebidas pelo servidor Docker guarda ainda numa String os *charCodes* de uma imagem, fazendo posteriormente a conversão desta String para Array de bytes de forma a mostrar a imagem na aplicação. Tem ainda um construtor que recebe todos os atributos da classe.

```
30 class Ementa {
31     Ementa.fromJson(Map<String, dynamic> json)
32         : img = json['img'],
33           imageBytes = null,
34           weekDay = json['weekDay'],
35           soup = json['soup'],
36           fish = json['fish'],
37           meat = json['meat'],
38           vegetarian = json['vegetarian'],
39           desert = json['desert'];
40
41     Map<String, dynamic> toJson() {
42         return {
43             'img': img,
44             'weekDay': weekDay,
45             'soup': soup,
46             'fish': fish,
47             'meat': meat,
48             'vegetarian': vegetarian,
49             'desert': desert,
50         };
51     }
52
53     Ementa(this.img, this.imageBytes, this.weekDay, this.soup, this.fish,
54           this.meat, this.vegetarian, this.desert);
55
56     late String? img;
57     late String? imageBytes;
58     late String? weekDay;
59     late String? soup;
60     late String? fish;
61     late String? meat;
62     late String? vegetarian;
63     late String? desert;
64 }
```

Figura 14 - Classes, Ementa

### 3.3 DiaSemana

A classe *DiaSemana* representa um dia da semana, sendo que é constituído pelo nome do dia em questão, a ementa original para esse dia, e se existir a ementa atualizada.

Tal como a classe *Ementa*, possui um método que permite inicializar um objeto com dados a partir de JSON.

```
66 class DiaSemana {
67     DiaSemana.fromJson(Map<String, dynamic> json, this.dia)
68         : original = Ementa.fromJson(json['original']),
69         update =
70             json['update'] == null ? null : Ementa.fromJson(json['update']);
71
72     Map<String, dynamic> toJson() {
73         return {
74             'dia': dia,
75             'original': original,
76             'update': update,
77         };
78     }
79
80     String dia;
81     final Ementa original;
82     Ementa? update;
83 }
```

Figura 15 - Classes, *DiaSemana*

### 3.4 ArgumentosEditScreen

A classe *ArgumentosEditScreen* permite armazenar os dados que serão passados para o ecrã de edição de um menu para determinado dia da semana. Esta classe permite armazenar os dados sobre o dia da semana em questão, bem como uma função que será chamada quando as atualizações do menu forem concluídas. Esta função irá realizar a atualização dos dados no ecrã principal.

```
85 class ArgumentosEditScreen {
86     ArgumentosEditScreen(this.diaSemana, this.callback);
87
88     DiaSemana diaSemana;
89     Function callback;
90 }
```

Figura 16 - Classes, *ArgumentosEditScreen*

## 4 JSON

O formato JSON é utilizado no contexto da aplicação para receber e enviar pedidos HTTP ao servidor Docker. As classes *Ementa* e *DiaSemana* permitem armazenar estes dados diretamente, possuindo métodos que transformam o objeto JSON recebido numa instância da classe. São efetuados dois pedidos HTTP em que o formato utilizado é o JSON, sendo que existe um terceiro pedido HTTP que não utiliza o formato JSON, sendo este o pedido que obtém a imagem para um determinado dia da semana.

### 4.1 DiaSemana

Métodos da classe *DiaSemana* que permitem converter uma instância da classe de e para JSON.

```
DiaSemana.fromJson(Map<String, dynamic> json, this.dia)
: original = Ementa.fromJson(json['original']),
  update =
    json['update'] == null ? null : Ementa.fromJson(json['update']);

Map<String, dynamic> toJson() {
  return {
    'dia': dia,
    'original': original,
    'update': update,
  };
};
```

Figura 17 - JSON, Conversão da classe *DiaSemana*

## 4.2 Ementa

Métodos da classe *Ementa* que permitem converter uma instância da classe de e para JSON.

```
31      Ementa.fromJson(Map<String, dynamic> json)
32      : img = json['img'],
33        imageBytes = null,
34        weekDay = json['weekDay'],
35        soup = json['soup'],
36        fish = json['fish'],
37        meat = json['meat'],
38        vegetarian = json['vegetarian'],
39        desert = json['desert'];
40
41      Map<String, dynamic> toJson() {
42        return {
43          'img': img,
44          'weekDay': weekDay,
45          'soup': soup,
46          'fish': fish,
47          'meat': meat,
48          'vegetarian': vegetarian,
49          'desert': desert,
50        };
51      }
```

Figura 18 - JSON, Conversão da classe *Ementa*

## 5 Shared preferences

De forma a conseguirmos guardar dados localmente entre execuções da aplicação são utilizadas as *shared preferences*. Estas permitem que quando a aplicação for aberta não seja logo feito um pedido ao servidor Docker, mas que sejam mostrados os dados guardados na última execução da aplicação. Para tal, sempre que fazemos um pedido ao servidor Docker os dados recebidos para além de serem mostrados no ecrã principal também são guardados na *shared preferences*. Estes dados são carregados quando a aplicação é aberta.

Os únicos dados que não são guardados na *shared preferences* são os bytes das imagens associadas a cada ementa, sendo que estas têm de ser obtidas através de um novo pedido ao servidor Docker.

### 5.1 Guardar dados

Os dados são guardados na *shared preferences* no formato JSON.

```
224 Future<void> _saveSharedPreferences() async {  
225     var prefs = await SharedPreferences.getInstance();  
226     prefs.setString('diasSemanaEmenta', jsonEncode(_diasSemanaEmenta));  
227 }
```

Figura 19 - Shared preferences, Guardar dados

### 5.2 Ler dados

Ao lermos os dados da *shared preferences* vamos primeiro ler para uma String. Se esta String estiver com o valor *null* significa que não existem dados guardados, caso contrário vamos realizar a decodificação dos dados de formato JSON para as nossas classes.

```
178 var prefs = await SharedPreferences.getInstance();  
179 String? userPref = prefs.getString('diasSemanaEmenta');  
180  
181 if (userPref == null) {  
182     _diasSemanaEmenta = null;  
183 } else {  
184     _diasSemanaEmenta = [];  
185     List<dynamic> lista = jsonDecode(userPref);
```

Figura 20 - Shared preferences, Ler dados

## 6 Câmara

De forma a atualizarmos a imagem que representa uma ementa para um determinado dia utilizamos a câmara do dispositivo para tirar uma fotografia. Para isto temos um ecrã que permite utilizar a câmara e tirar uma fotografia, sendo que esta devolve o caminho onde a fotografia foi guardada para o ecrã de edição através de uma função *callback* recebida por argumentos quando entramos no ecrã da câmara.

O ecrã da câmara foi fortemente influenciado pelo exemplo encontrado na internet (How to Use Camera in Flutter — Camera Package).



Figura 21 - Câmara, Ecrã de câmara



## 7 Permissões

Devido à utilização da câmara e da localização é preciso pedir permissão ao utilizador para a utilização destas funcionalidades do dispositivo. Para tal ao abriremos a aplicação é logo pedido ao utilizador permissões para utilizar estas funcionalidades.

### 7.1 Câmara

Para a utilização da câmara são pedidas as permissões de utilização da câmara e do microfone.

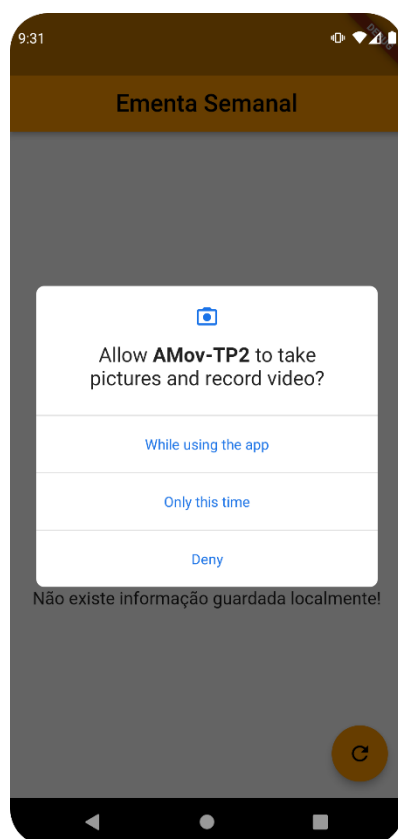


Figura 22 - Permissões Câmara,  
Tirar fotografia e gravar vídeo

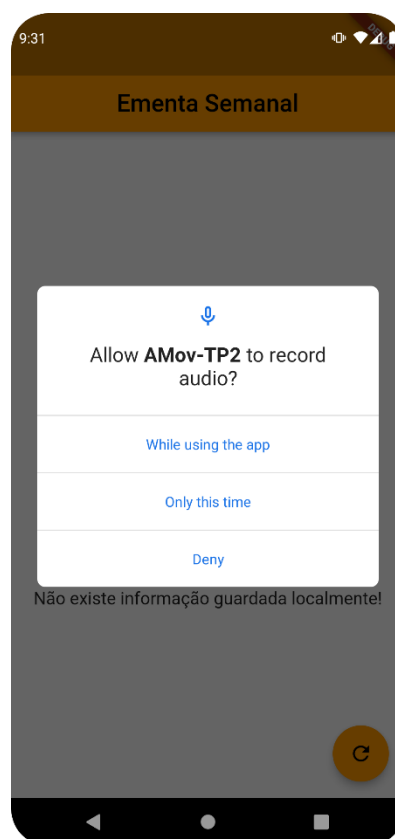


Figura 23 - Permissões Câmara,  
gravar áudio

## 7.2 Localização

Para a atualização de uma ementa temos de verificar a localização do utilizador de forma a determinar se este se encontra muito longe ou não do ISEC. Para isto temos de pedir ao utilizador a permissão para utilizar o serviço da localização.

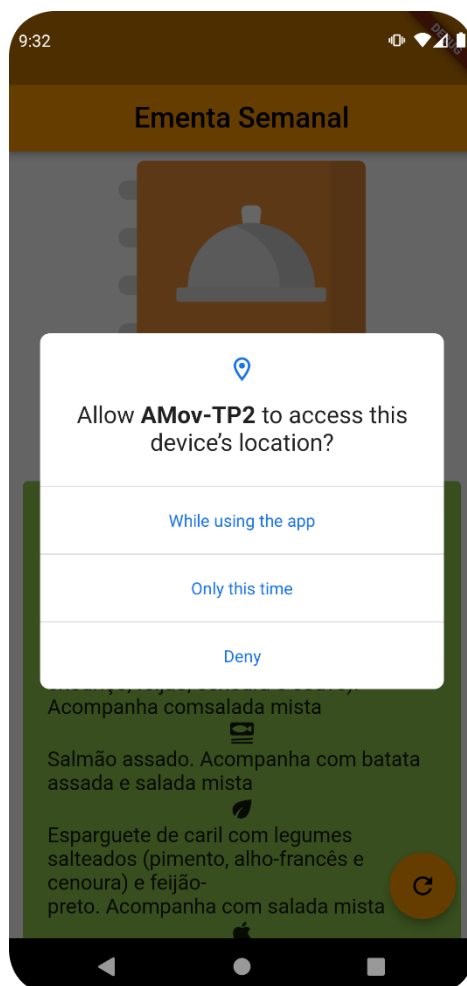


Figura 24 - Permissão localização

## 8 Localização

O plugin da localização permite obter a localização atual do dispositivo. Esta funcionalidade é utilizada para verificar se o utilizador se encontra próximo do ISEC para determinar se este pode ou não alterar a ementa de um determinado dia. As coordenadas do ISEC estão *hard coded* no programa, sendo comparadas as coordenadas obtidas com este plugin com as coordenadas guardadas. Para utilizarmos este plugin é necessário que o serviço de localização esteja ativo no dispositivo e que o utilizador tenha dados permissões de localização à aplicação. Caso uma destas condições não se verifique ou o utilizador esteja muito longe para editar um menu é mostrada uma mensagem ao utilizador a informar deste problema.



Figura 25 - Localização, Serviço não disponível



Figura 26 - Localização, Sem permissões



Figura 27 - Localização, Muito longe do ISEC

## 9 Referências

Putra, F. (7 de Janeiro de 2023). *How to Use Camera in Flutter — Camera Package*. Obtido de Medium: <https://medium.com/@fernandoptr/how-to-use-camera-in-flutter-flutter-camera-package-44defe81d2da>