

# Índice

[Índice 2](#_Toc124121534)

[Introdução 3](#_Toc124121535)

[Funcionalidades 4](#_Toc124121536)

[Icon e animações 4](#_Toc124121537)

[Ecrã principal 4](#_Toc124121538)

[Ecrã de edição 5](#_Toc124121539)

[Utilização da API 5](#_Toc124121540)

[Persistência de ementas 5](#_Toc124121541)

[Apresentação das imagens dos menus 6](#_Toc124121542)

[Funcionalidades Não Implementadas 7](#_Toc124121543)

# Introdução

O segundo trabalho prático de Arquiteturas Móveis consiste na criação de uma aplicação, em Flutter, que seja capaz de tirar partido de uma API para mostrar o menu semanal da cantina do ISEC. Para além disso esta aplicação tem de ser possível editar a refeição caso o utilizador detete que esta se encontra errada.

# Funcionalidades

## Icon e animações

O icon da aplicação foi alterado através do Android Studio. Adicionalmente, como alguns membros do grupo possuem dispositivos Apple (como um iPhone e Mac) alterámos também o icon para estas plataformas.

Quanto às animações utilizámos o widget Hero para animar as transições entre a página inicial e o ecrã de edição e vice-versa. Os widgets animados foram o nome do dia da semana e a imagem do menu.

 Uma imagem com texto

Descrição gerada automaticamente Uma imagem com texto

Descrição gerada automaticamente

O Hero funciona graças às tags que são dadas ao widget. Com base nestas tags, a transição é efetuada entre dois elementos que tenham a mesma tag. É, por este motivo, bastante importante que as tags sejam únicas.

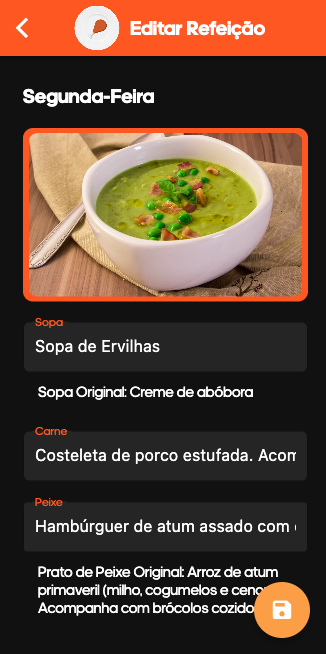
## Uma imagem com texto Descrição gerada automaticamenteEcrã principal

O ecrã principal da aplicação consiste em 5 widgets (MealCard) distribuídos numa ListView com scroll horizontal. Estes elementos são criados através da informação guardada nas shared-preferences ou através de um pedido à API facultada pelos docentes que retorna uma resposta em JSON que contém toda a informação relativa à ementa de uma semana.

Os campos a laranja indicam que a informação foi editada por utilizador e que esta não corresponde à informação original.

No canto inferior direito da página encontra-se um botão que permite ao utilizador atualizar as informações apresentadas pela aplicação.

## Ecrã de edição

O ecrã de edição, como mencionado anteriormente permite a um utilizador efetuar a atualização das informações de uma ementa.

A página consiste na fotografia da ementa (que não pode ser editada) e em TextFormFields que permitem ao utilizador inserir a nova informação.

Estes campos vêm pré-preenchidos de acordo com a informação presente no menu inicial. Se um campo estiver vazio então a API trata de sobrescrever a informação do menu com a informação original.

Quando um campo de uma ementa não corresponde à informação original a aplicação mostra em baixo do campo o valor original da ementa.

Finalmente, existe também um FloatingActionButton que permite ao utilizador submeter as alterações efetuadas. Se esta operação for efetuada com sucesso, o utilizador é redirecionado para o ecrã principal e a informação presente nos widgets é atualizada.

## Utilização da API

Para tirar partido da API

## Persistência de ementas

Para que o utilizador possa ter acesso à ementa offline, utilizámos o plugin shared-preferences para que sempre que o utilizador carregue no botão de atualizar o menu este seja guardado nas Shared Preferences do dispositivo. Ao iniciar a aplicação é carregado o menu a partir daí e os widgets são preenchidos com essa informação.

Uma imagem com texto

Descrição gerada automaticamente Uma imagem com texto

Descrição gerada automaticamente

## Apresentação das imagens dos menus

Em relação à apresentação das imagens dos menus, de modo a sabermos que imagem apresentar recorremos a operadores ternários de modo a identificar a imagem de acordo com o dia da semana. Na figura a baixo apresentamos uma parcialidade do código relativo à criação da MealCard que compõem as todas as informações para cada dia da semana e imagem.

Text

Description automatically generated

A picture containing text

Description automatically generatedUma das decisões de implementação relativamente à apresentação das imagens dos menus passou por apresentar uma imagem default caso não haja imagem para o dia da semana em questão. Na figura a baixo podes ver um excerto de código representativo do modo de obtenção do path da imagem caso exista, novamente recorrendo a operadores ternários.

Text

Description automatically generatedRelativamente à construção do próprio Widget que suporta a imagem, há que fazer igualmente a distinção se há imagem ou não, pois se não existir é um Image.asset visto que a imagem está local, caso contrário terá de ser um Image.network para que seja possível carregar a imagem através da rede.

# Funcionalidades Não Implementadas

Todas as funcionalidades relativas à primeira meta foram implementadas exceto o modo de multijogador que não foi implementado na sua totalidade.

As restantes funcionalidades da aplicação encontram-se completamente implementadas, sendo que seria possível até colocar o jogo na Play Store para mais pessoas poderem jogar!