Arquitecturas Móveis

Trabalho Prático 2 – 2022/2023

Tema

Pretende-se criar uma aplicação simples em *Flutter*, para apresentação e atualização da ementa semanal da cantina do ISEC, tirando partido de uma API facultada aos alunos.

Funcionalidades

A aplicação deverá permitir a consulta da ementa semanal da cantina do ISEC, incluindo para cada dia informação sobre: sopa, prato de peixe, prato de carne, prato vegetariano e sobremesa. Caso um utente da cantina detete que existe uma alteração relativamente ao menu previsto, poderá submeter uma atualização da informação através da aplicação. Por questões de simplificação do exercício assuma que a cantina apenas oferece almoços e a ementa repte-se todas as semanas (ou seja, apenas é necessário gerir uma semana de ementas, correspondente a 5 dias: segunda a sexta). A informação obtida do servidor para um determinado dia inclui a informação sobre as ementas originais e a última atualização da ementa que tenha sido efetuada por um utente da cantina (caso exista).

Deste modo, a aplicação deverá disponibilizar os seguintes ecrãs:

- **Ecrã principal**
  - Ecrã que será mostrado quando a aplicação for iniciada;
  - Deverá obter a ementa da semana a partir do servidor e listar de forma sucinta a informação
    - A lista deverá iniciar-se com a ementa para o dia atual seguida da informação para os dias seguintes para os quais exista informação
      - Se o dia atual for uma quarta-feira, a lista será constituída pela informação das ementas na seguinte ordem: quarta-feira, quinta-feira, sexta-feira, segunda-feira e terça-feira (notar que apenas é feita a gestão de uma semana);
    - Caso exista uma atualização da ementa original, deverá ser apresentada a informação relativa a essa atualização em vez da original, com uma formatação que permita distinguir da informação original;
  - O utilizador poderá selecionar um dia, o que originará a navegação para um segundo ecrã, onde será possível editar a informação;
  - Deverá existir um botão que permita refrescar a informação a partir do servidor
    - A operação de refrescamento deverá ser manual, exceto quando retorna do ecrã de edição e tenha sido realizada uma atualização da ementa;
    - No arranque da aplicação deverá ser mostrada a informação armazenada localmente relativa à última execução da aplicação. Caso não exista, deverá ser dada essa informação ao utilizador.

- **Ecrã de edição**
  - Neste ecrã será apresentada toda a informação relativa ao menu do dia selecionado, incluindo a informação original e a informação eventualmente atualizada por um utente;

    - Deverão ser disponibilizados campos que permitam editar a ementa para esse dia ou repor a informação original;
    - Deverá ser apresentado ainda um botão para submissão das alterações, caso exista alguma.

O aluno tem liberdade total na criação da *UI*. No relatório final deverá explicar a razão da sua escolha e as opções tomadas relativamente à organização do *layout* e respetivo *design*.

Características adicionais da aplicação que serão avaliadas:

- Mostrar as imagens dos menus diários, caso estas sejam disponibilizadas pelo servidor;
- [*Bónus*] Tirar foto e enviar imagem na alteração do menu com o *plugin camera*
    - A imagem deve ser submetida à API em Base64 (dart-convert);
- [*Bónus*] Aceitar apenas atualizações à ementa caso se esteja junto da cantina
    - Utilizar o plugin location para obter as coordenadas geográficas do dispositivo;
    - Por questão de simplificação assuma um par de coordenadas da cantina (*hard-coded*) e uma distância que pareça razoável para a gestão desta funcionalidade;
- Gerir a persistência da última ementa recebida pela API, através do *plugin shared preferences*;
- Definir um ícone personalizado para a aplicação;
- Utilizar de animações implícitas para apresentação de conteúdos;
- Estrutura e organização geral do projeto.

Restrições

- Para a implementação do projeto deverão ser usadas apenas as bibliotecas nativas do *Flutter* e as referidas anteriormente;
- A gestão de estado pode ser toda feita através do recurso ao `setState()` como realizado nas aulas;
- **As funcionalidades indicadas como *bónus* apenas serão avaliadas caso as restantes estejam implementadas**.

API existente

A API para o desenvolvimento deste trabalho prático deve ser executada com recurso ao *Docker* (`https://www.docker.com/`).

Após o *download* e instalação do *Docker*, deve obter a imagem disponibilizada no Nónio e correr os seguintes comandos para disponibilizar o "servidor", o qual ficará à escuta no porto 8080 do seu computador:

```
docker load < jib-amov-flutter-22-api-image.tar
docker run -p 8080:8080 ktor-amov-flutter-22:0.0.1
```

O "servidor" disponibiliza uma API com dois *endpoints*:

- `http://0.0.0.0:8080/menu`
    - GET *request* – devolve a ementa (ver `menu_get_shema.json`)
    - POST *request* – atualiza o menu de um dia (ver `menu_post_shema.json`)
        - caso algum campo de refeição venha vazio, este será substituído pelo campo do menu original
- `http://0.0.0.0:8080/images/<$path>`
    - GET *request* – devolve a imagem associada ao campo "`img`" de um menu

Notas:

- Caso o texto fornecido numa atualização de menu tenha tamanho nulo, então será assumido o menu original e eliminada a eventual atualização anterior feita a esse menu;
- O porto para aceder aos *endpoints* depende do mapeamento feito no comando `docker run`;
- Nos emuladores, para aceder ao *localhost* do computador o IP é `10.0.2.2`;
- O `content-type` para o POST *request* deve ser "`application/json; charset=UTF-8`".

Cotações

Interface e interação com o utilizador (incluindo *icon* e animações)..... 10,0%
Ecrã principal..... 30,0%
Ecrã de edição 15,0%
Utilização da API (obtenção e atualização de dados) 15,0%
Persistência de ementas (*shared preferences*) 7,5%
Apresentação das imagens dos menus..... 7,5%
Estrutura e organização geral do projecto..... 10,0%
Bónus: foto dos menus..... 10,0%
Bónus: restrição da localização para realizar atualizações 10,0%
Relatório Técnico 5,0%

Realização do trabalho: O mesmo grupo do Trabalho Prático 1 (*kotlin*)

Data de entrega: 08:00 do dia 09.01.2023

Forma de entrega:
Entrega de um único ficheiro em formato **ZIP** através do *Nónio* com o seguinte nome:

`AMOV.2022.2023.TP2.<nr_aluno1>.<nr_aluno2>.<nr_aluno3>.zip`

- **Ficheiros noutros formatos poderão ser alvo de penalização até 5% na nota final**

Este ficheiro deverá incluir:

- todo o código (pastas com os projetos) com todos os recursos essenciais para a compilação e execução.
  - Tendo em consideração que nem todos os alunos poderão ter acesso a equipamento adequado para realização do trabalho para funcionamento em iOS, para uniformização dos critérios de avaliação apenas será avaliada a componente Android. Assim, antes de submeter, deverá remover/eliminar as pastas `/ios` do projeto (caso a tenha) e executar o comando `flutter clean` (ou remover manualmente a pasta `<proj>/build` do projeto e `<proj>/.dart_tool/flutter_build`).
  - **Caso os ficheiros referidos não sejam removidos poderá ser aplicada uma penalização até 5% na nota final**
- relatório técnico (PDF)

Nota: caso o aluno não tenha acesso a equipamento adequado para compilar a versão iOS, poderá optar logo pela exclusão dessa componente durante o processo de criação. Na criação do projeto através da linha de comandos, a restrição ao sistema Android pode ser realizada acrescentando o argumento `--platforms android` a seguir ao `create`.