isec
Politécnico de Coimbra

Weather

Relatório

Ângelo Paiva, 2019129023
Jan Frank, 2017009793
Pedro Henriques, 2019129770

Arquiteturas Móveis
Licenciatura de Engenharia Informática, Ramo de Desenvolvimento de Aplicações
22 de janeiro de 2022

Conteúdo

1 Introdução **2**

2 Funcionamento Geral da Aplicação **2**

3 API Escolhida **2**

4 Shared Preferences **2**

5 Internacionalização **3**

6 Interface do Utilizador **3**

7 Suporte para iOS **3**

8 Conclusão **3**

9 Anexos **3**

1 Introdução

Neste trabalho foi projetada uma aplicação para consultar metereologia. Esta aplicação foi escrita em Dart com recurso a Flutter.

Este relatório irá servir para explicar a implementação e a razão por trás das decisões que foram tomadas.

2 Funcionamento Geral da Aplicação

A primeira vez que a aplicação é aberta após ter sido instalada esta, por defeito, faz um pedido implícito de modo a ter dados para preencher inicialmente o ecrã. Após isto, sempre que é aberta, esta preenche o ecrã com a informação armazenada localmente nas **shared preferences**.

Para atualizar os dados, o utilizador tem de pressionar o botão de refresh que se encontra no canto superior direito.

3 API Escolhida

A API que escolhemos para a realização deste trabalho foi OpenWeatherMap. Achamos que esta fornece bastante informação e, ao ler reviews online, chegámos também à conclusão que é das que tem mais precisão em relação às previsões.

Para além disto, esta tem também um elevado limite de pedidos para a versão gratuita, podendo fazer até 60 pedidos por minuto ou 1,000,000 de pedidos por mês.

Apesar de fornecer informação para minuto, decidimos ignorar porque não era detalhada o suficiente para ser útil para o utilizador normal.

Para além de usar a API para obter previsões metereológicas, também a usamos para obter o nome da cidade associada às coordenadas.

Por fim, esta API também nos fornece um pacote de ícones relativo às previsões metereológicas. Para cada previsão vem associado um campo **icon** que contém um ID para uma imagem que, para obter, basta usar um link fornecido por eles.

4 Shared Preferences

A última atualização feita é sempre guardada offline com recurso a shared preferences. Nesta temos dois campos, um para guardar o nome da cidade onde o pedido foi feito chamado **city** e outro onde guardamos toda a informação recebida da API chamado **weatherData**.

Para guardar a informação recebida da API nas shared preferences foi necessário codificar o JSON para String.

5 Internacionalização

A nossa aplicação tem suporte para inglês e português com recurso ao plugin intl.

6 Interface do Utilizador

A nossa interface foi inspirada nesta interface. Apesar de termos inicialmente baseado nessa, consideramos que a nossa interface se tornou única e com a sua própria personalidade com o decorrer do projeto.

O objetivo era ter uma interface minimalista mas que não falhasse a apresentar informação ao utilizador.

O ecrã de detalhe só está disponível para as previsões diárias por acharmos que, para o utilizador comum, não é necessário obter informações mais detalhadas para as previsões por hora. Deste modo mantemos a aplicação minimalista.

Para mostrar as inúmeras previsões futuras, sejam as diárias ou as por hora, usámos uma ListView com scroll horizontal por acharmos que é o que mais se adequa ao layout da aplicação, ao invés de seguir a estratégia usada pelo exemplo que nos inspirou de ter uma segunda página para listar mais para além daquelas que aparecem no ecrã.

7 Suporte para iOS

Como tínhamos membros com máquina da Apple, a aplicação foi testada em iOS e funciona corretamente.

8 Conclusão

Para além de ter sido um verdadeiro desafio, este trabalho foi uma excelente oportunidade para conhecer melhor **Flutter** e todas as vantagens que nos proporciona.

9 Anexos

Lista de Figuras

Pedaços de Código