



# **Weather**

## Relatório Técnico

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

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>2</b>
<b>2</b>	<b>API Escolhida</b>	<b>2</b>
<b>3</b>	<b>Shared Preferences</b>	<b>2</b>
<b>4</b>	<b>Internacionalização</b>	<b>2</b>
<b>5</b>	<b>Interface do Utilizador</b>	<b>2</b>
<b>6</b>	<b>Suporte para iOS</b>	<b>3</b>
<b>7</b>	<b>Conclusão</b>	<b>3</b>
<b>8</b>	<b>Anexos</b>	<b>3</b>

# 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 API Escolhida

A API que escolhemos para a realização deste trabalho foi [OpenWeatherMap](#). Acharmos 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 segundo 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.

## 3 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.

## 4 Internacionalização

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

## 5 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.

## 6 Suporte para iOS

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

## 7 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.

## 8 Anexos

Lista de Figuras

Pedaços de Código