Simple-Weather

computação móvel e ubíqua

DIOGO XAVIER, 8160123

ricardo fernandes, 8160240

2018

Índice

[1. INTRODUÇÃO 2](#_Toc534038182)

[1.1. Propósito do documento 2](#_Toc534038183)

[2. VISÃO GERAL 2](#_Toc534038184)

[2.1. Descrição do projeto 2](#_Toc534038185)

[2.2. Requisitos do sistema 3](#_Toc534038186)

[2.3. Use cases e sua descrição 4](#_Toc534038187)

[2.4. Diagrama de classes 7](#_Toc534038188)

[3. A APLICAÇÃO 9](#_Toc534038189)

[3.1. Estrutura da aplicação 9](#_Toc534038190)

[3.2. Requisitos de implementação obrigatória 16](#_Toc534038191)

[3.3. Requisitos de bonificação 18](#_Toc534038192)

# 1. INTRODUÇÃO

## 1.1. Propósito do documento

O objetivo deste documento passa por descrever todos os elementos que compõem o projeto “simple-weather”. O documento apresentará todos os requisitos, funcionalidades e recursos a que o projeto foi sujeito.

# 2. VISÃO GERAL

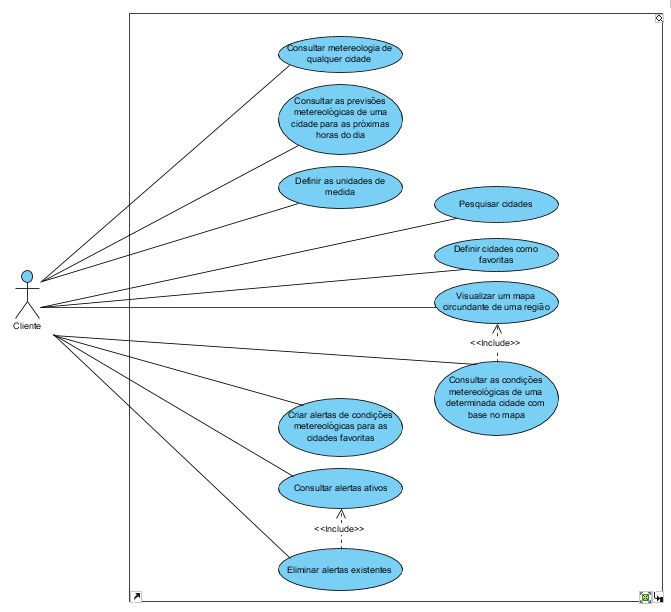
## 2.1. Descrição do projeto

O projeto consiste numa aplicação móvel que permite ao utIlizador consultar os dados metereológicos. A aplicação permite pesquisar cidades e consultar o estado atual do tempo e previsões seguintes, permitindo também possuir uma lista de cidades favoritas que o utilizador deseje. Além disso, é possível observar um mapa que, após a seleção de um determinado local, serão apresentadas as condições metereológicas do mesmo. É possível ainda criar alertas para alterações metereológicas e observar a lista de alertas ativos. Por fim, é possível definir em que unidade de medida algumas informações serão apresentadas.

## 2.2. Requisitos do sistema

* O utilizador deve poder consultar a metereologia atual de qualquer cidade.
* O utilizador deve poder consultar a metereologia de horas futuras ao longo do dia, de qualquer cidade.
* O utilizador deve poder selecionar uma, ou mais, cidades como favoritas às quais deverá ter fácil acesso no dashboard.
* O utilizador deve poder definir qual as unidades que deseja que a aplicação use.
* O utilizador deve poder pesquisar por qualquer cidade que deseje. Adicionando-as como favoritas, caso deseje.
* O utilizador deve poder criar alertas para vários parâmetros fornecidos pela api (temperatura, humidade, velocidade do vento ou pressão atmosférica).
* O utilizador deve poder, facilmente, escolher as cidades para as quais quer criar um determinado alerta.
* O utilizador deve poder visualizar um mapa da sua localização atual e regiões circundantes.
* O utilizador deve poder consultar as condições metereológicas de uma cidade, selecionando um determinado local do mapa.
* O utilizador deve poder consultar a lista de alertas ativos.
* O utilizador deve poder eliminar um alerta ativo caso assim deseje.

## 2.3. Use cases e sua descrição



**1. Consultar a metereologia de qualquer cidade**

* **Descrição:** O utilizador poderá consultar a metereologia atual de qualquer cidade.
* **Verificação:** Serão apresentados, na atividade principal, os dados metereológicos da cidade escolhida.

**2.** **Consultar as previsões metereológicas de uma cidade para as próximas horas do dia**

* **Descrição:** O utilizador poderá verificar as condições metereológicas para o resto do dia da cidade escolhida.
* **Verificação:** O utilizador poderá deslizar, através de uma barra no fundo da atividade principal, para as horas seguintes do dia onde serão apresentados os dados previstos do tempo.

**3. Definir as unidades de medida**

* **Descrição:** O utilizador poderá definir as unidades de medida que a aplicação irá usar (temperatura: C, K e F, e velocidade do vento: m/s ou km/h)
* **Verificação:** Os dados obtidos aparecerão consoante as unidades de medida definidas.

**4. Pesquisar cidades**

* **Descrição:** O utilizador poderá pesquisar por uma cidade à sua escolha.
* **Verificação:** No menu lateral da direita encontra-se uma barra de pesquisa onde será possível pesquisar cidades. Os resultados obtidos aparecerão imediatamente abaixo da barra de pesquisa.

**5. Definir cidades como favoritas**

* **Descrição:** O utilizador poderá definir quais as suas cidades favoritas para a consulta dos dados metereológicos.
* **Verificação:** As cidades definidas como favoritas serão guardadas na base de dados e aparecerão inicialmente no menu lateral da direita.

**6. Consultar um mapa circundante de uma região**

* **Descrição:** O utilizador poderá ver um mapa da sua localidade e regiões circundantes.
* **Verificação:** Será apresentado um mapa da localização atual e regiões circundantes.

**7. Consultar as condições metereológicas de uma determinada cidade com base no mapa**

* **Descrição:** O utilizador poderá visualizar as condições metereológicas de uma determinada cidade selecionando um local do mapa.
* **Verificação:** Após selecionar um local do mapa, serão apresentados os dados metereológicos da cidade escolhida.

**8. Criar alertas de condições metereológicas para as cidades favoritas**

* **Descrição:** O utilizador poderá criar alertas, definindo o seu tipo (alerta de temperatura, alerta de humidade, alerta da velocidade do vento ou alerta da pressão atmosférica).
* **Verificação:** O alerta definido será guardado na base de dados e será apresentado na lista de alertas ativos.

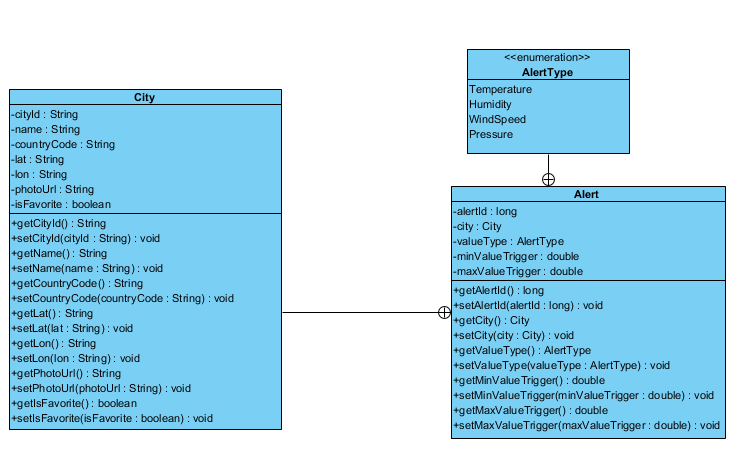
**9. Consultar os alertas ativos**

* **Descrição:** O utilizador poderá consultar os alertas que estão atualmente ativos.
* **Verificação:** Será apresentada a lista de alertas ativos. Cada alerta informa a cidade a que se refere, o tipo de alerta e os seus valores.

**10. Eliminar alertas existentes**

* **Descrição:** O utilizador poderá eliminar um alerta ativo.
* **Verificação:** Na lista de alertas, cada alerta possui um botão de eliminar. É através desse botão que o alerta escolhido pode ser eliminado.

## 2.4. Diagrama de classes



* **City:** representa a cidade que foi definida como favorita pelo utilizador

**Atributos:**

* + **cityId:** número de identificação da cidade
  + **name:** nome da cidade
  + **countryCode:** código do país onde a cidade se localiza
  + **lat:** latitude
  + **lon:** longitude
  + **photoUrl:** url da imagem da cidade
  + **isFavorite:** verificar se o utilizador colocou a cidade como favorita
* **Alert:** representa o alerta criado pelo utilizador

**Atributos:**

* **alertId:** número de identificação do alerta
* **city:** cidade a que o alerta se refere
* **valueType:** tipo de alerta (temperatura, humidade, velocidade do vento, ou pressão atmosférica)
* **minValueTrigger:** valor mínimo do tipo escolhido
* **maxValueTrigger:** valor máximo do tipo escolhido

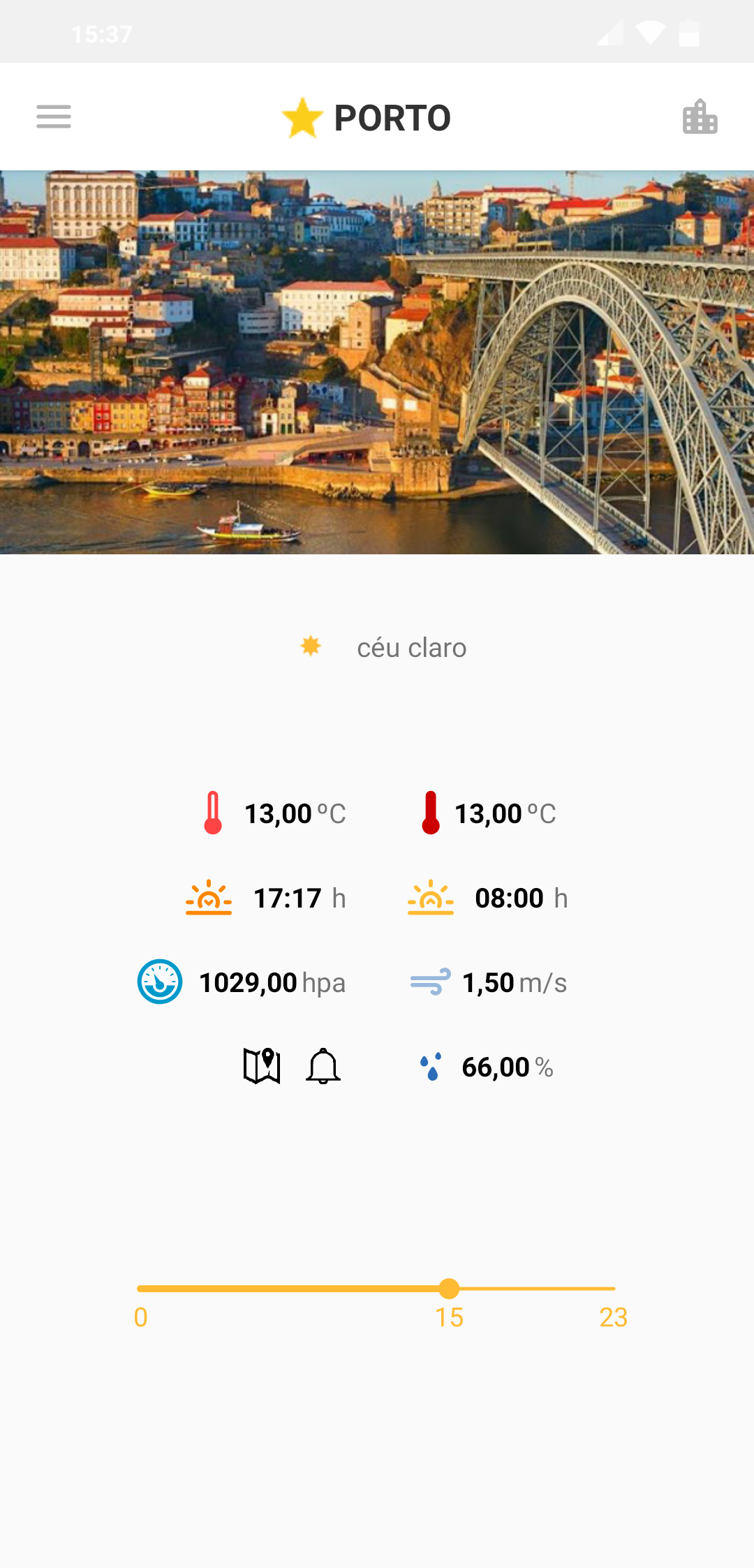
# 3. A APLICAÇÃO

## 3.1. Estrutura da aplicação

A aplicação divide-se em três atividades, sendo que praticamente todo o processo encontra-se na atividade principal. Esta atividade possui dois menus laterais, cada um com o seu próprio propósitio.

* **Início**

Figura 1- Página inicial da aplicação



Após iniciar a aplicação é mostrada a página inicial (definida por um fragmento) em que são mostrados os dados metereológicos da localização atual do utilizador. Os dados apresentados são a descrição do tempo, temperatura mínima, temperatura máxima, hora do nascer e pôr do sol, pressão atmosférica, velocidade do vento e humidade. Existe ainda uma barra abaixo que permite consultar as previsões para o resto do dia.

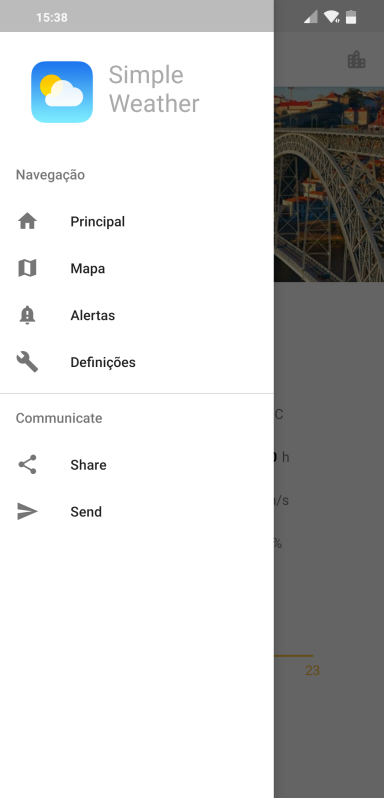
* **Menus laterais**

Figura 2 - Menu lateral da esquerda

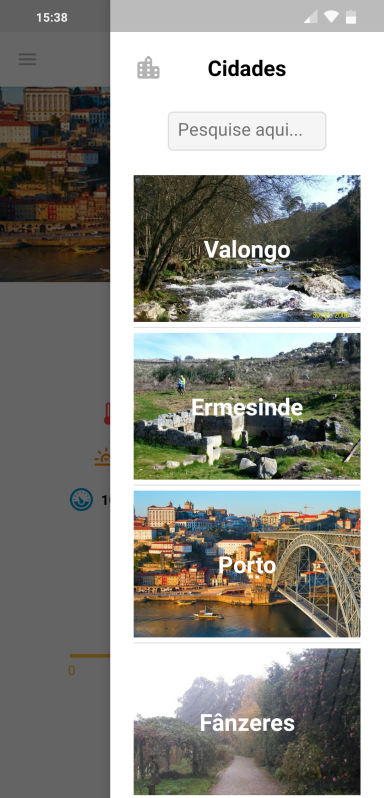
****

Figura 3 - Menu lateral da direita que inicialmente apresenta as cidades favoritas

Existem dois menus laterais: o menu da direita e o menu da esquerda.

**- Menu lateral da direita**

Este menu permite fazer a pesquisa de cidades. Os resultados aparecem imediatamente abaixo dessa barra em forma de lista (figura abaixo). Ao selecionar uma cidade a página inicial apresentará os dados metereológicos da mesma. Além disto, aquando da abertura deste menu, será apresentada a lista de cidades favoritas do utilizador (fig. 3).

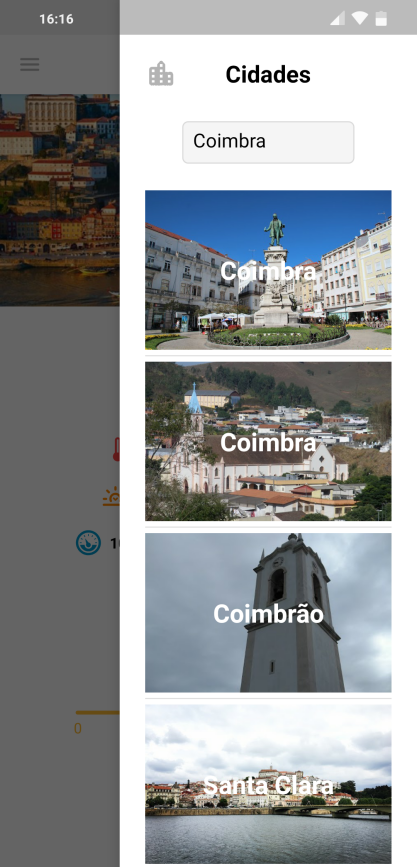


Figura 4 - Resultados da pesquisa "Coimbra"

**- Menu lateral da esquerda**

Neste menu são apresentados vários pontos de escolha (fig. 2). São eles:

* + **Principal:** Representa a página inicial onde são mostrados os dados metereológicos da localização atual inicialmente, das cidades pesquisadas através do menu da direita ou através de um local escolhido no mapa (fig. 5).

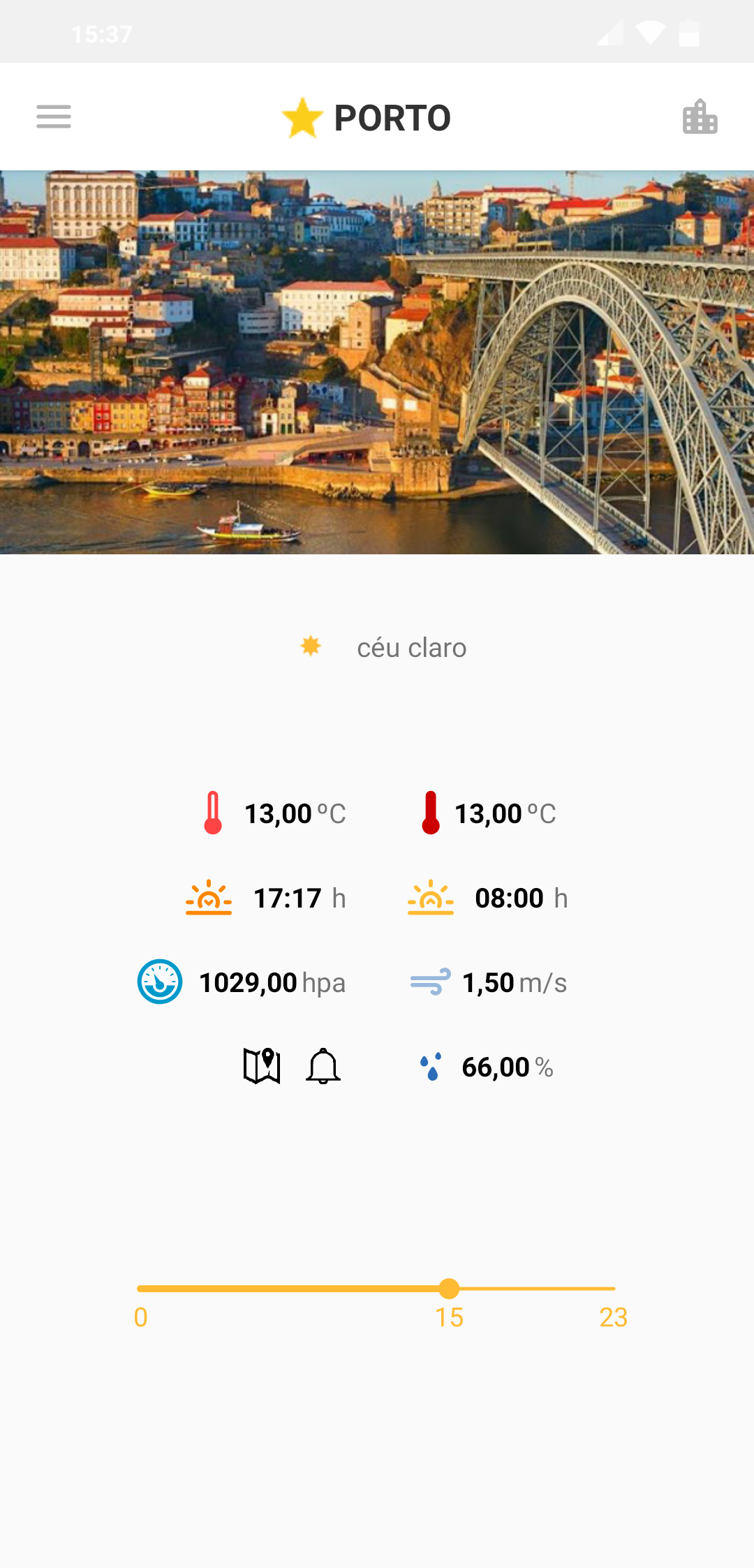


Figura 5 - Dados metereológicos da cidade do Porto. A estrela ao lado do nome da cidade indica se a cidade está marcada como favorita ou não (as cidades que apresentem a estrela de cor amarela estão marcadas como favoritas).

* + **Mapa:** Apresenta um mapa da localização atual do utilizador e regiões circundantes (fig. 6), podendo mover-se para qualquer lado que deseje. É possível ainda selecionar um ponto do mapa para verificar as condições metereológicas desse local (fig. 7).

Figura 7- Cidade apresentada após selecionar "Vila Nova de Gaia" no mapa

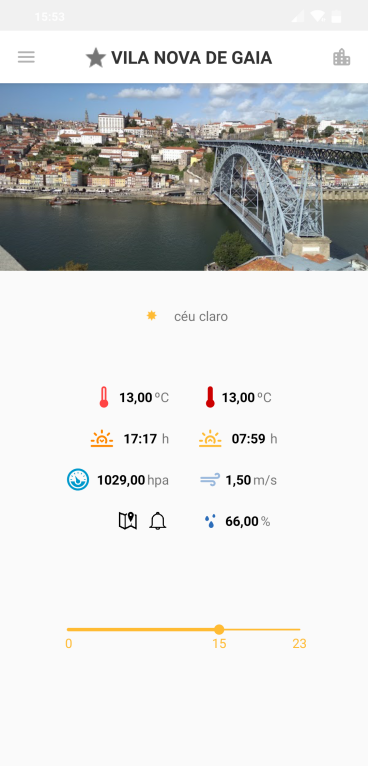


Figura 6- Mapa

* + **Alertas:** Nesta secção são apresentados todos os alertas ativos que foram criados (fig. 8). Cada alerta possui um botão de eliminar que permite eliminar um determinado alerta, caso assim o utilizador deseje. No canto inferior direito possui um botão que permite abrir a página para criar um novo alerta.

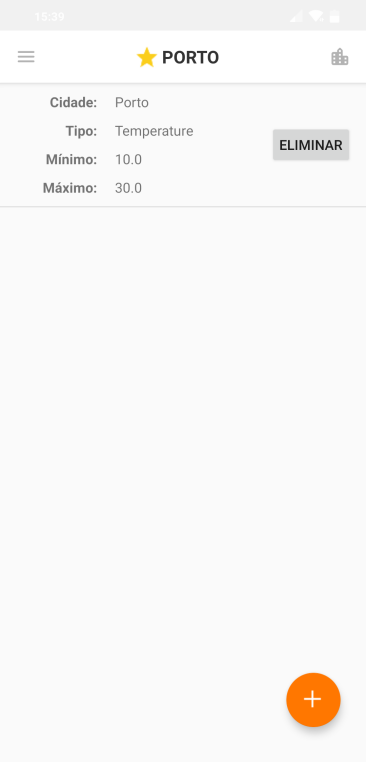


Figura 8 - Lista de alertas ativos. O alerta pode ser eliminado, caso o utilizador assim deseje. O botão no canto inferior permite criar um novo alerta.

* + **Definições:** Apresenta preferências que o utilizador pode escolher em relação às unidades de medida em que os dados irão ser apresentados (ver abaixo).

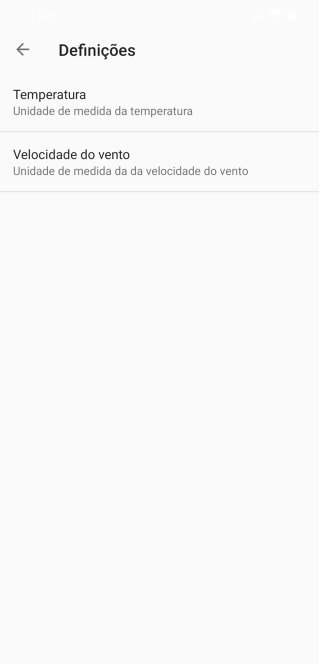
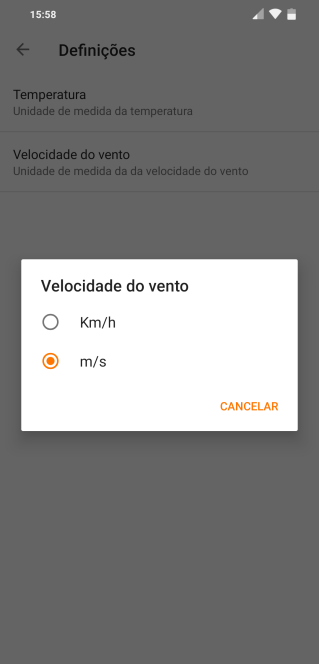
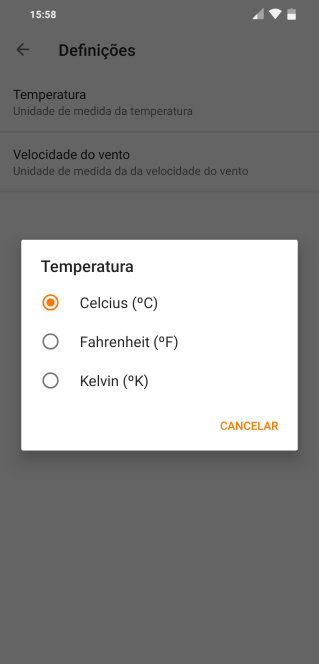


Figura 11 - Opções da velocidade do vento

Figura 10 - Opções da temperatura

Figura 9 - Definições

**Nota:** À exceção das definições, todos os outros pontos são representados através de fragmentos.

* **Novo alerta**

Esta página permite criar um novo alerta (ver abaixo). Para ser criado é preciso definir a cidade a que o alerta se refere, o tipo de alerta (alerta de temperatura, alerta de humidade, alerta da velocidade do vento ou alerta da pressão atmosférica) e o valor mínimo e máximo do tipo de alerta escolhido. Após a criação do alerta, este ficará ativo e será apresentado na lista de alertas existentes.

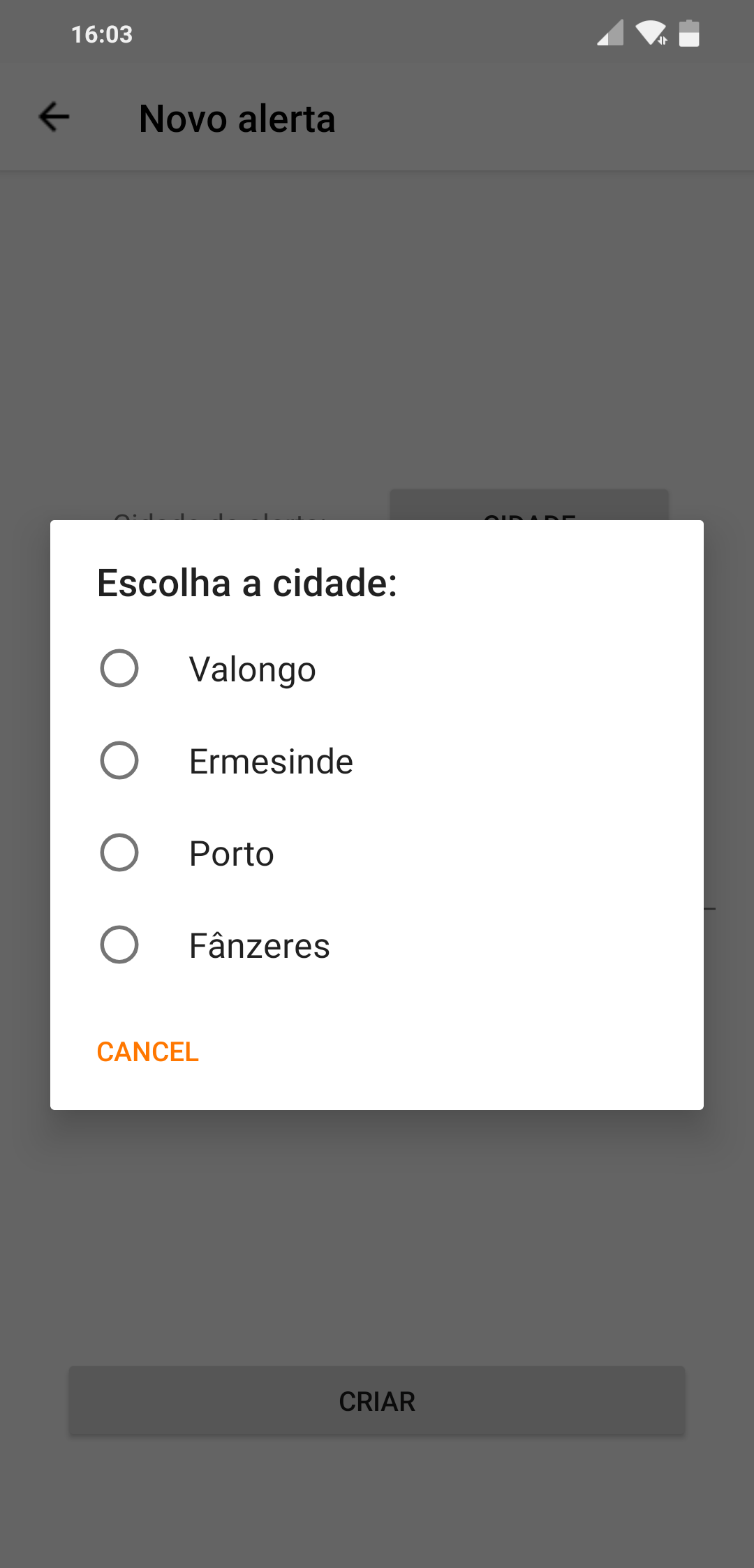


Figura 12 - Novo alerta

Figura 14 - Escolher o tipo de alerta

Figura 13 - Escolher a cidade para o alerta

**Nota:** Os alertas apenas podem ser definidos para uma cidade favorita do utilizador.

## 3.2. Requisitos de implementação obrigatória

Nesta secção será apresentado o modo de como os requisitos obrigatórios foram implementados no projeto.

• **Uso do GitLab e Git para gestão e versionamento do código do projeto (git e GitLab)**

* Uso de GitHub como repositório
* Uso de Git para gestão do código

**• Suporte para ecrãs de diferentes dimensões (Telemóvel e Tablet)**

* Aplicação de landscape views para as atividades existentes

**• Uso das guidelines do material design**

* Icones e estilos da aplicação

**• Uso de listas (RecyclerView e Adapters)**

* Lista de cidades favoritas
* Resultados da pesquisa de cidades
* Lista de alertas

**• Uso das bibliotecas de suporte (Android Support Library)**

* Glide - Usado para facilitar o download de imagens de fontes externas, a sua representação, e otimização através de cache.
* Retrofit - Usado para fazer pedidos às diversas API’s de forma idiomática e recomendada pela plataforma Android.
* Butterknife - Permite reduzir a complexidade do código, injetando as dependências necessárias para interagir com os elementos definidos no xml.
* BubbleSeekBar - Elemento usado para permitir ao utilizador selecionar a hora para a qual deseja ver a previsão metereológica.
* Android-Jobs - Usado para facilitar a utilização de tarefas agendadas em Android. Necessário para agendar as verificações dos alertas e, caso os limites de um alerta sejam ultrapassados, ativar uma notificação.

**• Uso de notificações**

* Alertas

**• Uso de base de dados (Room)**

* Guardar cidades favoritas e alertas ativos

**• Uso de operações assíncronas (AsyncTask/ Thread/ IntentService)**

* Adicionar e eliminar alertas

**• Uso de sensores de localização e disponibilização de informação em mapas**

* Obter a localização atual
* Mapa da localização atual e regiões circundantes

**• Integrações com API’s de serviços externos via pedidos REST (Retrofit)**

* Obtenção de dados e previsões metereológicas da API OpenWeatherMap
* Obtenção das cidades através da API Google Place e Google Geocoding

## 3.3. Requisitos de bonificação

Nesta secção serão apresentados os requisitos de bonificação implementados

* **Uso webservices adicionais via pedidos REST (Retrofit)**
  + Obtenção das cidades através da API Goolgle Place
* **Interação com elementos do Android**
  + Sensores de localização
  + Background scheduled tasks