

Relatório técnico -- Módulo Android

Living City



Data: Aveiro, 14 de Junho de 2016

Autores: [68779] Rui Oliveira (ruipedrooliveira@ua.pt)
[68129] Tomás Rodrigues (tomasrodrigues@ua.pt)

Resumo projecto: Gerir, monitorizar e observar em tempo real a temperatura, humidade, qualidade do ar, e pontos turísticos de qualquer cidade de Portugal.

Lista de Conteúdos:

[1. Introdução](#)

[2. Âmbito da aplicação](#)

[Requisitos da aplicação](#)

[3. Interface do utilizador \(UI\)](#)

[4. Técnicas e opções de design](#)

[5. Implementação técnica](#)

[6. Sincronização de dados](#)

[7. Tecnologias utilizadas](#)

[8. Backend information](#)

[9. WorkFlow da aplicação](#)

[10. Organização dos módulos do projeto](#)

[11. Melhorias efectuadas](#)

[12. Conclusão](#)

[13. Referências e recursos](#)

2. Âmbito da aplicação

Os utilizadores podem utilizar a aplicação sempre que pretenderem visualizar a temperatura do local onde estão ou para onde se deslocam. A temperatura disponibilizada é medida por todos os utilizadores que possuam ou passem perto de um sensor BLE (do inglês *Bluetooth Low Energy*). As temperaturas medidas são disponibilizadas diariamente. Para além disto existe um *backoffice* que permite consultar graficamente a evolução da temperatura e da humidade numa determinada cidade e num determinado dia, bem como a concentração de um determinado poluente. Na própria aplicação pode ainda ser visto um gráfico com duração diária para os valores e variações de temperatura e humidade numa determinada localização.

Saber rapidamente a qualidade do ar em Portugal ou os pontos turísticos a visitar numa nova cidade, por exemplo, quando se está em viagem são outros dos benefícios que oferecemos.

De modo a completar um pouco mais a aplicação a nível de funcionalidades tendo por base a localização atual do utilizador, tínhamos pensando em utilizar a API do panoramio (<http://www.panoramio.com/api/widget/api.html>) de modo a apresentar as fotografias também de pontos turísticos em redor do utilizador.

Na figura seguinte encontra-se representado o diagrama *use case* da nossa aplicação.

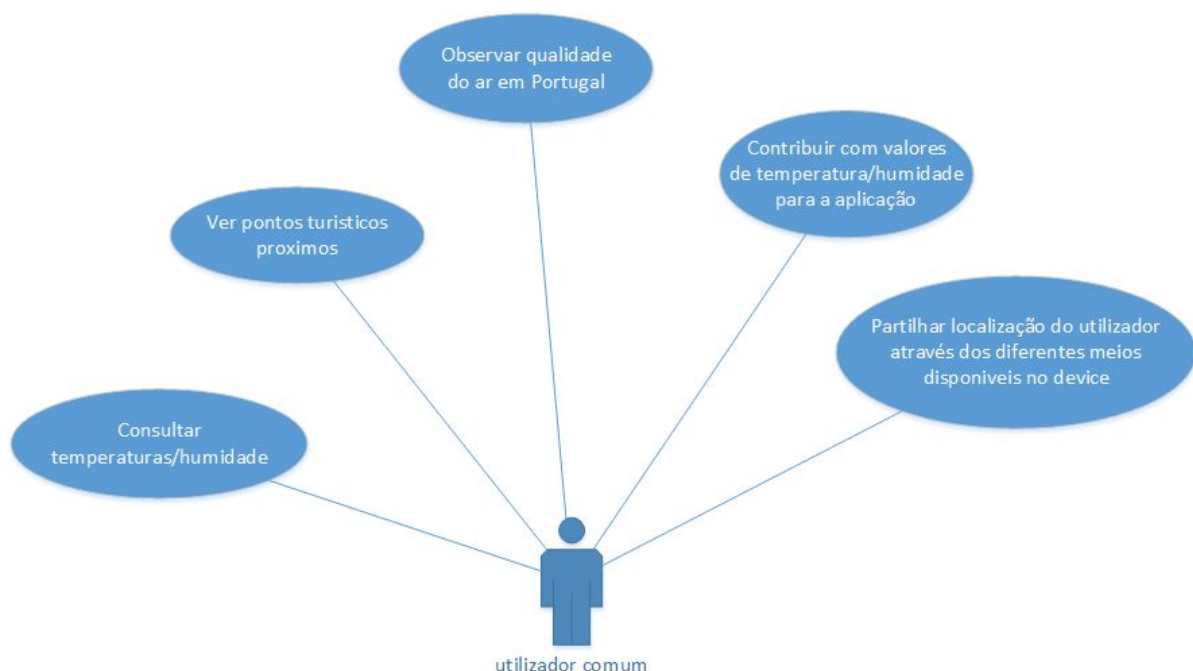


Figura 3: UML use-cases principais da aplicação

Requisitos da aplicação

Para o utilizador usar a aplicação basta ter um Android acima da versão 4.4 e acima da API 21, e:

- Para ver a qualidade do ar em Portugal e os pontos turísticos nas proximidades o utilizador terá de ter conectividade à Internet;
- Para fazer medições reais com o sensor BLE o utilizador terá de ter ativado o Bluetooth e GPS do seu dispositivo. No nosso caso, o sensor BLE utilizado foi o sensor tag CC2541 da Texas (<http://www.ti.com/tool/cc2541dk-sensor>)

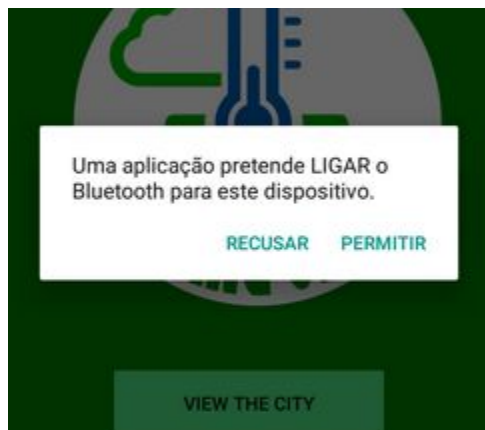


Figura 4: Permissão Bluetooth

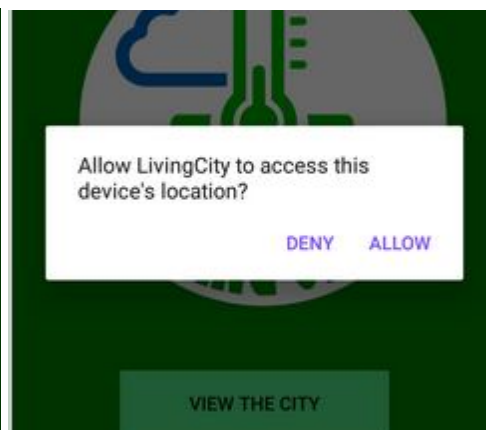


Figura 5: Permissão GPS

3. Interface do utilizador (UI)

Toda a interface do utilizador aplica os conceitos *user friendly* como o símbolo do menu no canto superior esquerdo do ecrã já familiar aos utilizadores ou simples *checkbox*'s.

Temos também para ajudar o utilizador uma tab/fragmento com uma legenda simples que explica tudo o que precisa de saber acerca a informação apresentada. O menu lateral (sidebar android) também é extremamente intuitivo e possui ícones ilustrativos do que cada tab/fragmento apresenta (Figura 6).

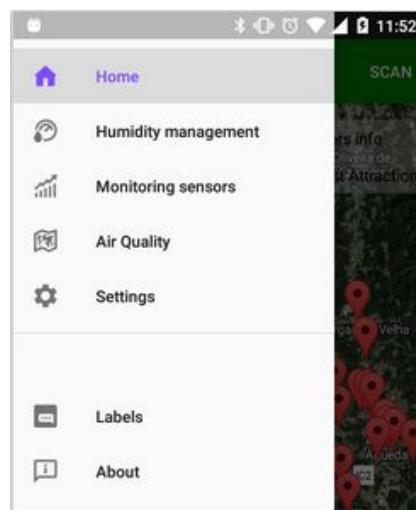


Figura 6: Sidebar menu da aplicação

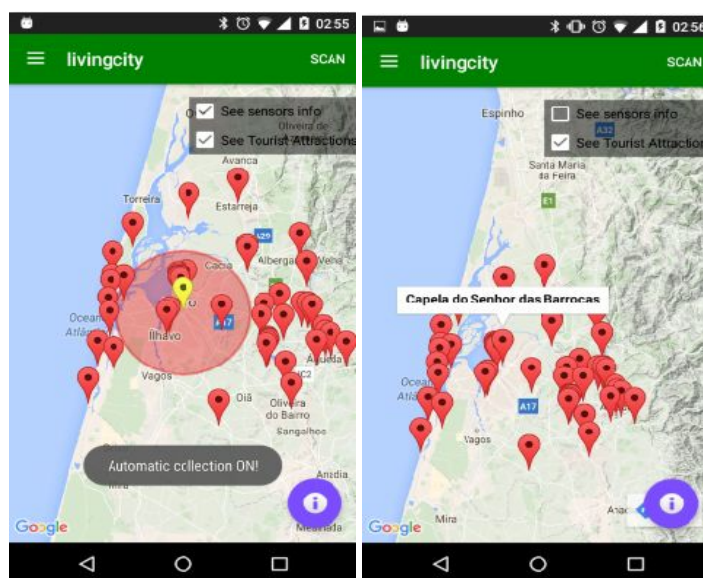
Nos fragmentos que contêm mapas, a interação com o utilizador é feita com simples cliques nos círculos e nos markers que dão toda a informação que o utilizador pretende saber. No fragmento que contém o gráfico é possível ao utilizador saber o valor das temperaturas/humidades medidas com simples cliques na hora desejada.

Pensamos que a experiência do utilizador está bastante simples e intuitiva. Temos mecanismos de feedback para todos os cliques e notificações para dar informações acerca de algum requisito necessária, por exemplo, permissões quando algo é tentado executar fora da normalidade como foi visto acima.

4. Técnicas e opções de *design*

Temos uma animação usando o logótipo da nossa aplicação quando o utilizador a inicia que, posteriormente passará para a *Activity* principal através de um botão. Em relação às cores como pode ser observado temos à volta dos markers dos sensores círculos que variam do vermelho ao azul, indicando se em volta daquela localização está mais ou menos quente, respetivamente. O fragmento respetivo à qualidade do ar e humidade segue o mesmo modelo de gradiente mas com escalas de verde e azul, respetivamente.

Para além disso temos duas *checkbox*'s que podem estar ou não ativas permitindo ao utilizador filtrar a informação que quer ver no mapa de modo a ter uma personalização a seu gosto.



Figuras 8 e 9: Filtragem de informação no mapa

O logótipo criado para esta aplicação baseia-se na sobreposição de três componentes: um termómetro, uma nuvem e um plano que representa um mapa. A cor principal do logótipo é o verde, uma vez que transmite um simbolismo de uma cidade limpa e pura. Ao fundo, é apresentado o nome da aplicação em formato elíptico.

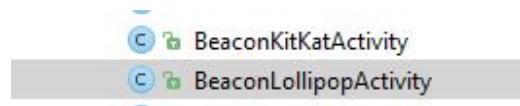


Figuras 10: Proposta de logótipo

5. Implementação técnica

Nesta aplicação temos duas Activities, uma inicial com uma pequena animação com o logótipo da aplicação alterando as cores, e outra Activity principal que contém um *sidebar* (menu) e vários fragmentos.

O fragmento que é visto inicialmente pelo utilizador na Activity principal chama-se *SensorTouristicFragment* e contém um mapa da Google (API Google Maps) com a informação das temperaturas medidas ao longo dos dias e pontos turísticos próximos da localização de onde o utilizador está a usar a aplicação. Acrescentamos ainda que para as medições com o sensor foram implementadas no código duas versões de leitura, uma para o Android Kitkat e outra para o Android Lolipop sendo que estas se efetuam de forma diferente em cada versão Android.



No segundo fragmento, *HumidityFragment*, temos o mapa de Portugal dividido em distritos com a informação dos valores da humidade num gradientes de tons de azul. Nos distritos em que não existem valores para os dados da humidade a área do distrito respetivo mantêm-se translúcida.



Figura 9 - Fragmento AirQualityFragment

O terceiro fragmento visto na figura 9 - AirQualityFragment contém as estações de medições da qualidade do ar em Portugal bem como a informação desta última utilizando para isso o mesmo algoritmo utilizado pela Agência Portuguesa do Ambiente que pode ser visto na Figura 10.

Poluente em causa / Classificação	CO		NO ₂		O ₃		PM ₁₀		SO ₂	
	Min	Máx	Min	Máx	Min	Máx	Min	Máx	Min	Máx
Mau	10000	-----	400	-----	240	-----	120	-----	500	-----
Fraco	8500	9999	200	399	180	239	50	119	350	499
Médio	7000	8499	140	199	120	179	35	49	210	349
Bom	5000	6999	100	139	60	119	20	34	140	209
Muito Bom	0	4999	0	99	0	59	0	19	0	139

NOTA: Todos os valores anteriormente indicados estão em µg/m³

Independentemente de quaisquer factores de sinergia entre diferentes poluentes, o grau de degradação da qualidade do ar estará mais dependente da pior classificação verificada entre os diferentes poluentes considerados, pelo que o IQar será definido a partir do poluente que apresentar pior classificação (ex: valores médios registados numa dada área: SO₂ - 35 µg/m³ (Muito Bom), NO₂ - 180 µg/m³ (Médio); CO - 6000 µg/m³ (Bom), PM₁₀ - 15 µg/m³ (Muito Bom) e O₃ - 365 µg/m³ (Mau) o IQar será **mau**, devido às concentrações observadas para o **ozono**).

Figura 10: Método do cálculo do índice da qualidade do ar

Informações mais específicas sobre os valores de poluentes num dado dia ou hora podem ser vistas pelo administrador do sistema utilizando para isso um *backend* desenvolvido por nós que vai ser falado mais à frente neste relatório.

Animação inicial com logotipo da aplicação - LivingCity

Para a criação da animação da activity inicial foi usada uma **animation-list** com diferentes item's. Mais detalhes sobre a implementação desta funcionalidade estão disponíveis através da ligação presente na secção *Referências e Recursos*.

```
<?xml version="1.0" encoding="utf-8"?>
<animation-list android:layout_width="match_parent" android:layout_height="match_parent"
xmlns:android="http://schemas.android.com/apk/res/android" android:oneshot="true">
    <item android:drawable="@drawable/verdeall" android:duration="500" />
    ...
</animation-list>
```

Obtenção dos pontos turísticos em torno da localização do utilizador

Para obtenção dos pontos turísticos foi utilizado o seguinte *webservice*: <http://www.tixik.com/api/nearby?lat=XX&lng=YY&limit=50&key=demo>, em que **XX** e **YY** correspondem, respetivamente, à latitude e à longitude onde o utilizador se situa.

Através deste *webservice* tivemos que criar uma *AsyncTask* que permite a recolha dos dados e o seu processamento em formato XML. Mais ainda, este *webservice* devolve 50 pontos turísticos em torno do utilizador, número este definido através do parâmetro **limit**.

Obtenção do distrito através das coordenadas do utilizador

Para obtenção do distrito foi utilizado o seguinte webservice: <https://maps.googleapis.com/maps/api/geocode/xml?&latlng=XX,YY>, em que **XX** e **YY** correspondem, respetivamente, à latitude e à longitude onde o utilizador se situa.

Através dele tivemos que criar uma **AsyncTask** que permite a recolha dos dados e o seu processamento dos dados em formato XML.

Recolha automática de valores (temperatura/humidade) pelo dispositivo móvel

Com o objetivo tornar automática a recolha de dados pela aplicação decidimos criar um **IntentService** que é executado em *background* na aplicação. Este serviço permite que num intervalo de três minutos sejam recolhidos novos dados através do dispositivo. Os dados recolhidos são os seguintes:

- Temperatura: *Device temperature* (**TYPE_TEMPERATURE**)
- Humidade: *Ambient relative humidity* (**TYPE_RELATIVE_HUMIDITY**)

As entradas guardadas na firebase são exatamente as mesmas caso a temperatura e humidade fossem recolhidas pelo sensor Tag. Por padrão, a recolha automática de dados é ativada quando a aplicação é iniciada, contudo através da tab *settings* é possível desactivar esta recolha automática.

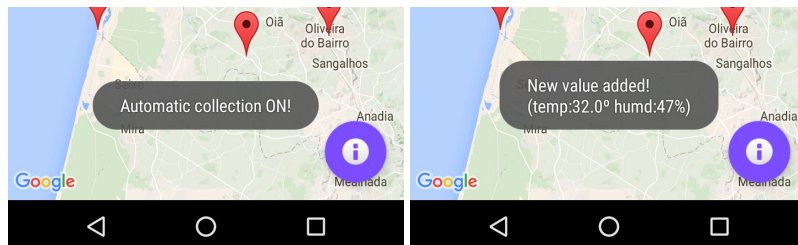


Figura 11: Feedback dado ao utilizador após/durante a recolha automática

Tal como dito anteriormente, ao iniciar a aplicação a recolha de dados é ativada automaticamente e fornecido o seguinte feedback ao utilizador (figura anterior à esquerda). De 3 em 3 minutos são adicionados novos dados, o feedback desta operação também é transmitida ao utilizador (figura anterior à direita).

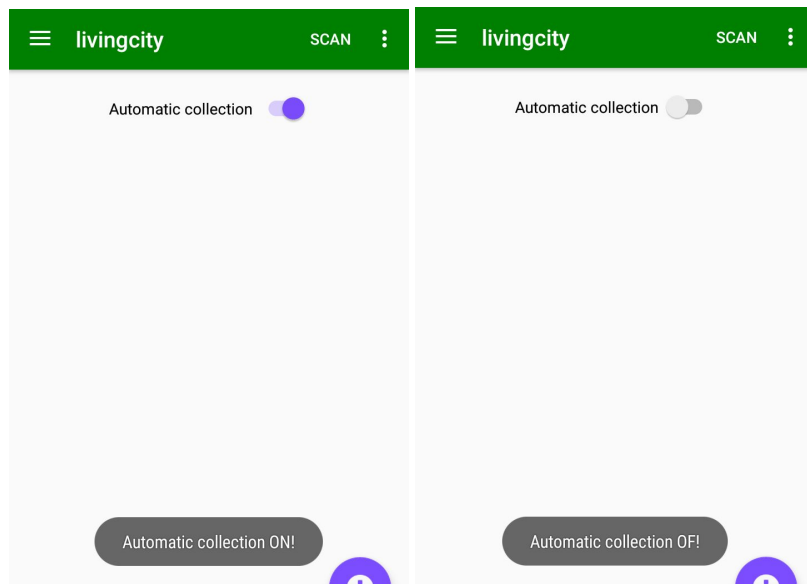


Figura 12: Tab *Settings* que permite ativar/desativar recolha automática de dados

Através do fragmento *settings* é possível desativar a recolha automática de dados utilizando um **Switch**. Sempre esta ação é ativada ou desativada é transmitido um feedback ao utilizador através de um **Toast**, tal como é possível observar na figura 12.

Divisão dos diferentes distritos de Portugal (apresentação de valores de humidade e qualidade do ar)

As coordenadas dos vários distritos foram criadas manualmente através do site [geojson](http://geojson.io) (geojson.io) e posteriormente guardadas na firebase. As mesmas podem ser observadas através de jsondistricts.firebaseio.com.

Através da ligação das diferentes coordenadas é possível criar cada um dos distritos de Portugal, através destes são criados Polygons (polígonos) e sobrepostos no mapa (Google Maps) existente na nossa aplicação.

Partilha da localização atual do utilizador

Através da nossa aplicação o utilizador poderá partilhar a sua localização recolhida pela mesma. Esta localização é enviada através de um link do Google Maps **<http://maps.google.com/maps?q=XX,YY>**, em que **XX** e **YY** correspondem, respetivamente, à latitude e à longitude onde o utilizador se situa. O utilizador poderá partilhar a sua localização através de qualquer aplicação existente no seu dispositivo disponível para o efeito.

```
Intent sendIntent = new Intent();
sendIntent.setAction(Intent.ACTION_SEND);
sendIntent.putExtra(Intent.EXTRA_TEXT, "My location sends via
```

```

LivingCity\nhttp://maps.google.com/maps?q="+gps.getLatitude()+","+gps.getLongitude());
sendIntent.setType("text/plain");
startActivity(sendIntent);

```

Gráficos para monitorização dos dados de humidade e temperatura

De modo a permitir uma visualização gráfica dos dados medidos pela nossa aplicação criámos um fragmento que permite incluir dois gráficos: um da temperatura e outro da humidade. Ambos os valores são apresentados apenas para o presente dia do distrito onde se localiza o utilizador.

Para a criação dos gráficos utilizámos a biblioteca android chamada EazeGraph. Esta biblioteca disponibiliza várias opções de gráficos, nomeadamente:

- Bar Chart
- Stacked Bar Chart
- Pie Chart
- Line Chart

Para a presente aplicação utilizámos duas *Line Chart*. A sua utilização é bastante simples tal como é possível observar no excerto de código seguinte e através do seguinte link: <https://github.com/blackfizz/EazeGraph>

```

ValueLineSeries seriesTemp = new ValueLineSeries();
seriesTemp.setColor(Color.GREEN);
for (int i=0; i<sensorValue.size(); i++){
    Log.e("ce", sensorValue.get(i).getDistrict());
    if (sensorValue.get(i).getDistrict().equals(DistrictsModule.getDistrict())){
        String temp =
sensorValue.get(i).getTemperature().split("°")[0].replace(",",".");
        seriesTemp.addPoint(new ValueLinePoint(sensorValue.get(i).getHora(),
Float.parseFloat(temp)));
    }
}
chartTemp.addSeries(seriesTemp);
chartTemp.addStandardValue(27);
chartTemp.addStandardValue(-5);
chartTemp.addStandardValue(55);
chartTemp.setOnPointFocusedListener(new IOnPointFocusedListener() {
    @Override
    public void onPointFocused(int _PointPos) {
        Log.d("Test", "Pos: " + _PointPos);
    }
});

```

6. Sincronização de dados

Os dados dos pontos turísticos obtidos através de um XML de uma API online são adquiridos imediatamente com recurso à implementação de uma **AsyncTask** na Activity inicial. Também nesta activity inicial é recolhido o distrito da localização do utilizador, como visto anteriormente. Os dados da informação lida através dos sensores é obtido no *SensorTouristicFragment* para que no momento duma nova leitura de temperatura e/ou humidade se possa atualizar este fragmento e este actualize imediatamente as informações com a nova medição lida incluída o mapa.

Por último os dados das estações são obtidos de um acesso à Firebase também na *SensorTouristicFragment* não havendo problemas com a sincronização de dados, pois o fragmento estando associado a ela quando é chamado tem toda a informação já pronta a ser mostrada corretamente.

Para cada medição efectuada, seja ela através do sensor tag ou por recolhada automática, são guardados os seguintes dados na firebase:

- Distrito, Hora, Humidade, Temperatura, Longitude e Latitude

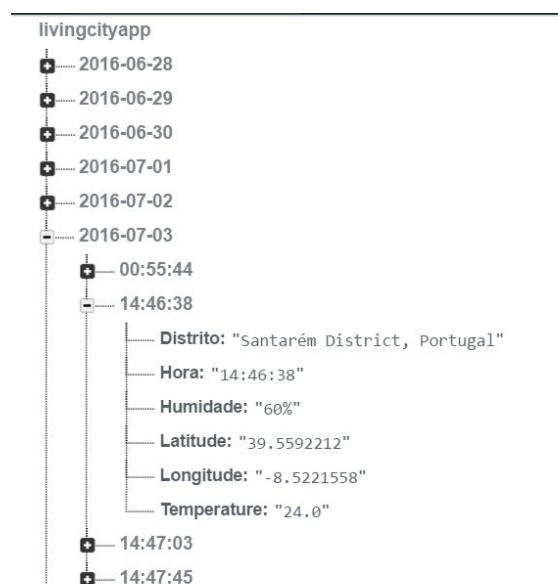


Figura 13: Estrutura para guardar dados na firebase (livingcityapp.firebaseio.com)

7. Tecnologias utilizadas

- Android sdk
- XML
- JSON
- API Google Maps
- Firebase (<https://www.firebase.com/>)
- Backoffice (HTML, JS, CSS) + gráficos (Chartist.js)
- Bluetooth BLE
- Sensor Tag

Leitura XML através de uma API online tratado totalmente numa AsyncTask garantindo a sincronização de dados quando o Fragmento inicial inicia na segunda Activity após a animação inicial.

Dados dos poluentes e estações estão em JSON são obtidos apartir de uma API REST criada na disciplina de Engenharia de Software localizada num servidor local, para além disso toda a Firebase guarda e modela os dados em objetos JSON que são depois modelados e tratados para apresentar a informação de forma concisa.

Firebase guarda todos os nossos dados. Informação dos sensores guardada num único nó (por dia) ficando as informações/medições relacionadas diariamente, exemplo disto pode ser visto acima na figura 13. Para além disso a informação de tudo relativo à informação das estações e dos poluentes também é lá guardada e a informação dos pontos/coordenadas necessários para dividir Portugal em distritos estará também guardada nesta mesma firebase no futuro.

Por último mas não menos importante trabalhar com o bluetooth e o sensor Tag da Texas foi outro dos desafios ultrapassados neste trabalho. Relativamente ao backoffice tudo é falado mais abaixo e o logótipo foi feito trabalhando com Photoshop.



Figura 13: Tecnologias usadas

8. Backend information

Na disciplina de Engenharia de Software já tínhamos contruído uma API REST para obter os dados da Agência Portuguesa do Ambiente relativos às estações e aos poluentes medidos nessas estações ao longo do dia (de hora a hora). Um gráfico exemplo para um determinado poluente numa dada estação pode ser visto na imagem seguinte.

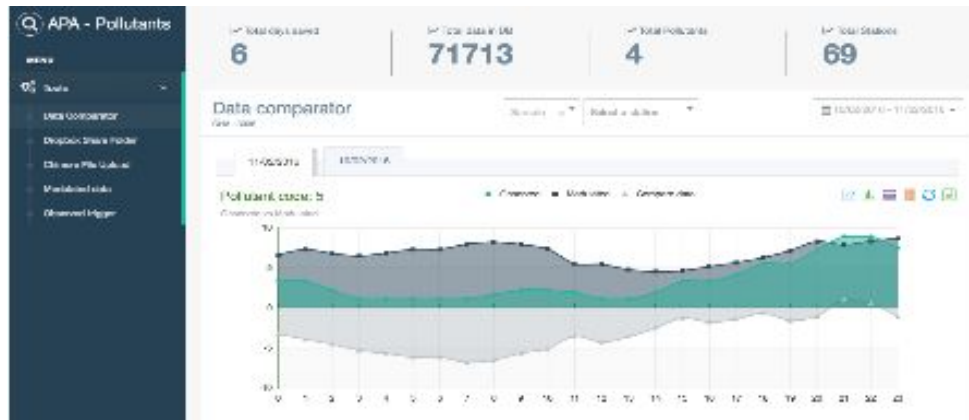


Figura 14: Exemplo de gráfico observado no backend para um poluente

No âmbito da disciplina de Computação Móvel, foi desenvolvido um pequeno *backend* que permite observar uma evolução da temperatura e humidade ao longo de um dia, para um determinado dia do ano (em que exista medições) e para uma determinada cidade. Um exemplo disso pode ser observado nas imagens seguintes:

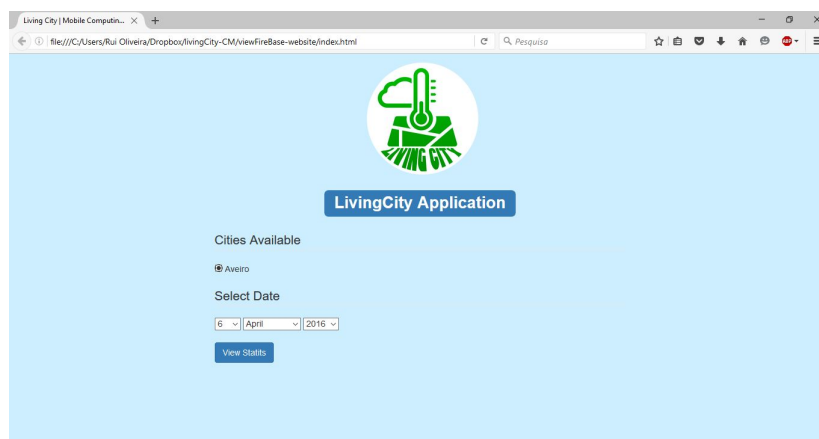


Figura 15: Escolha da data

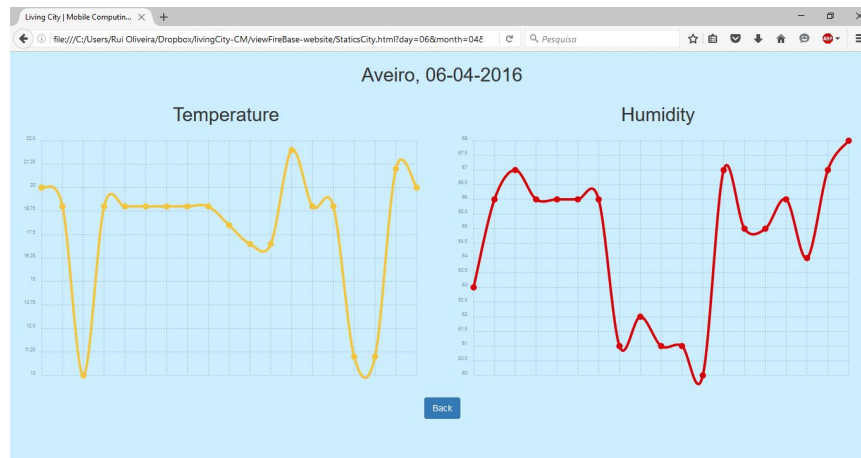


Figura 16: Visualização de gráficos diários

9. Workflow da aplicação

O workflow da nossa aplicação é apresentado no esquema seguinte. Seguidamente iremos especificar cada vista que aqui se apresenta.

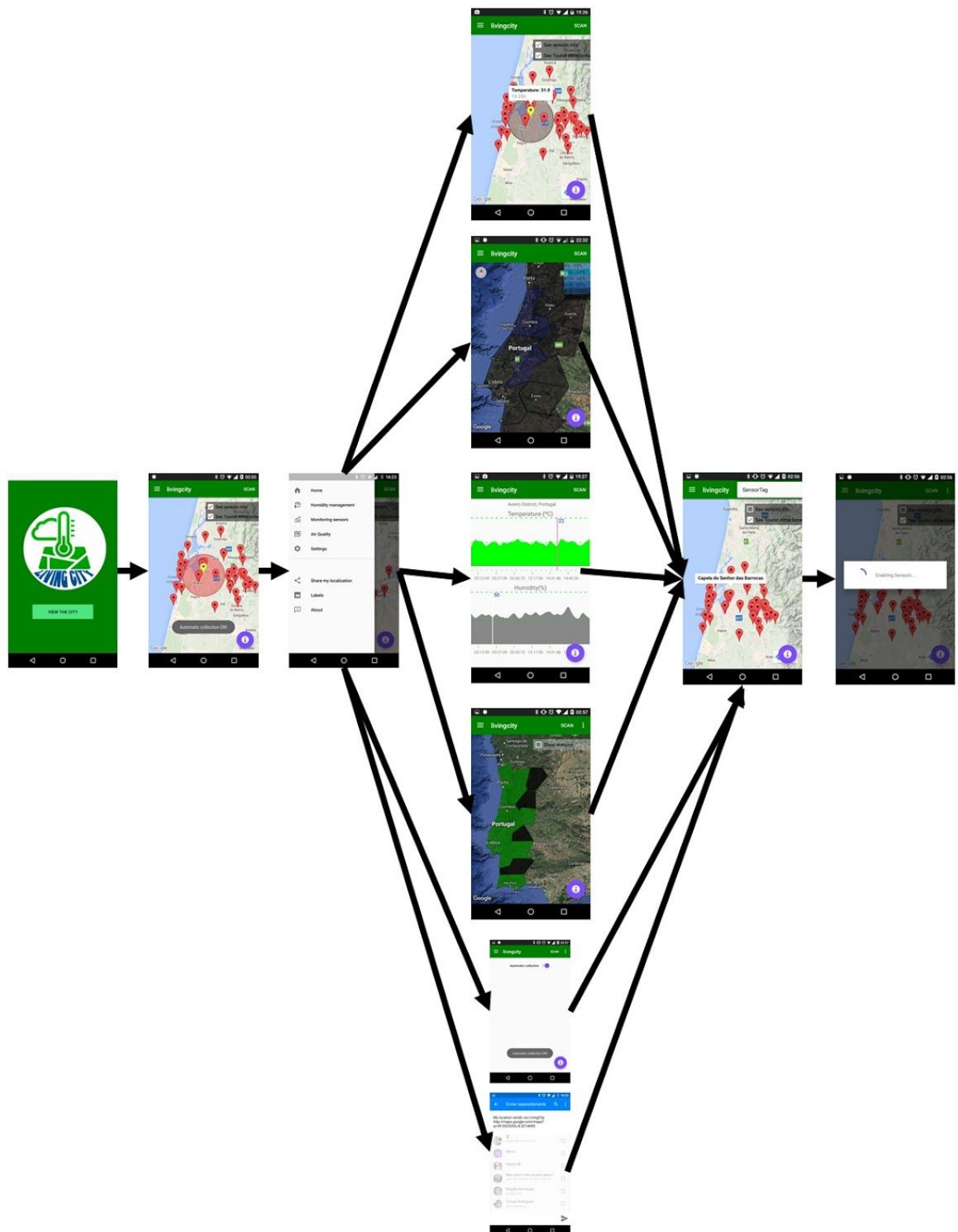




Figura 17

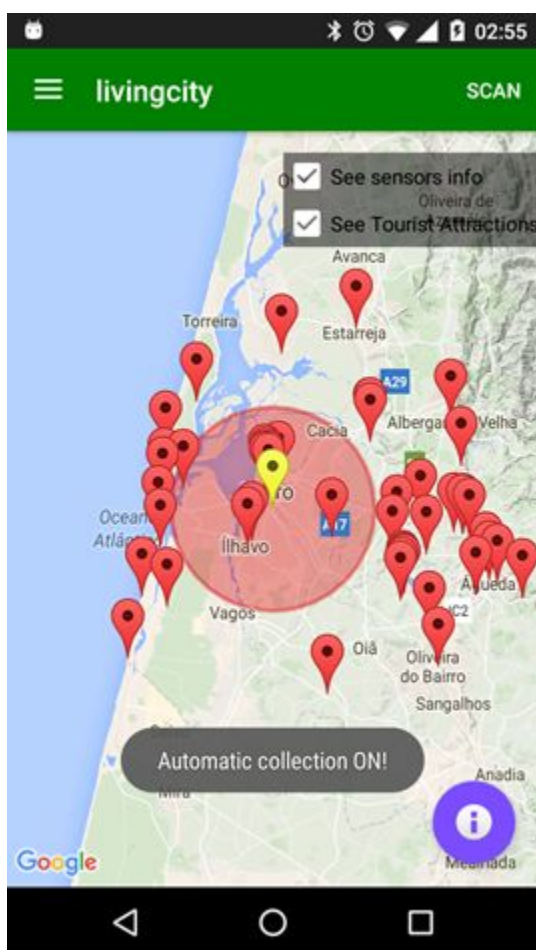
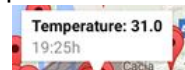


Figura 18

O utilizador após instalar a aplicação irá aceder-lhe a ser-lhe-á apresentada a seguinte vista (*FristActivityAnimation*). Nesta *activity* existe uma animação construída através da sobreposição de várias imagens. Na mesma *activity* existe um botão que permite prosseguir o processo.

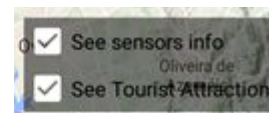
Após o cenário anterior, o utilizador irá ser redirecionado para uma nova *activity* constituída por vários fragmentos, o primeiro fragmento possui um mapa centrado na localização do utilizador. Os marcadores apresentados a vermelho dizem respeito aos pontos turísticos em redor da localização do utilizador, por outro lado, os representados a verde representam todas as medições efectuadas pelos utilizadores via sensor BLE ou recolhidos automaticamente pelo dispositivo móvel a cada 3 minutos.

Ao pressionar o marcador verde, será apresentado a seguinte informação de temperatura e respectiva hora de medição.



Ao pressionar um marcador vermelho será apresentado o nome do ponto turístico correspondente à localização.

Ao interagir com as *checkbox* no canto superior direito, é possível ocultar cada um dos tipos de marcadores existentes.



Os círculos de diferentes cores representam a variação de temperatura existente (cores mais azuis correspondem a temperaturas mais baixas enquanto que as temperaturas mais vermelhas correspondem a temperaturas mais altas).

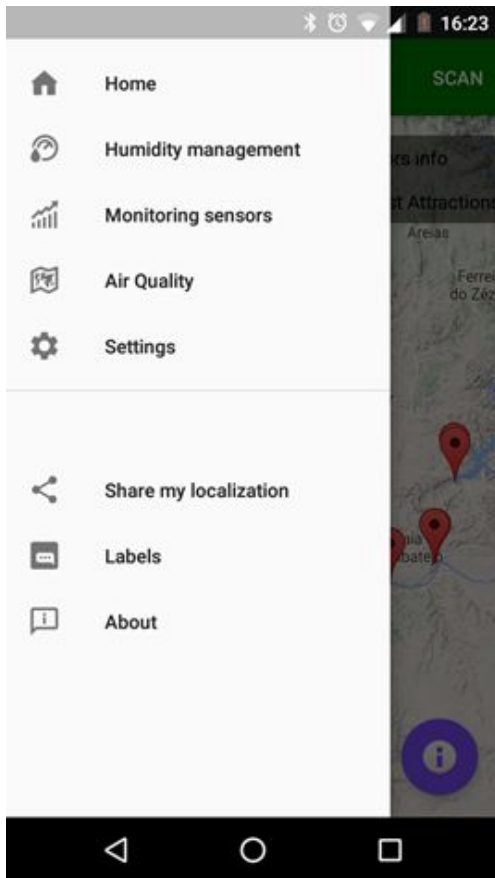


Figura 19

Através de qualquer fragmento da activity principal é possível aceder ao *side bar* lateral. Esta side bar permite navegar através de toda a aplicação e aceder a todos os fragmentos existentes e às diferentes funcionalidades subjacentes.

O modo de aceder a este menu pode ser efectuado de duas formas distintas:

- Clicando sobre o icon que de seguida se apresenta (em todos os fragmentos da activity);



- Efectuando um movimento suave da esquerda para a direita no lado esquerdo do ecrã do dispositivo.

A entrada *Home* permite aceder ao fragmento apresentado na imagem anterior.

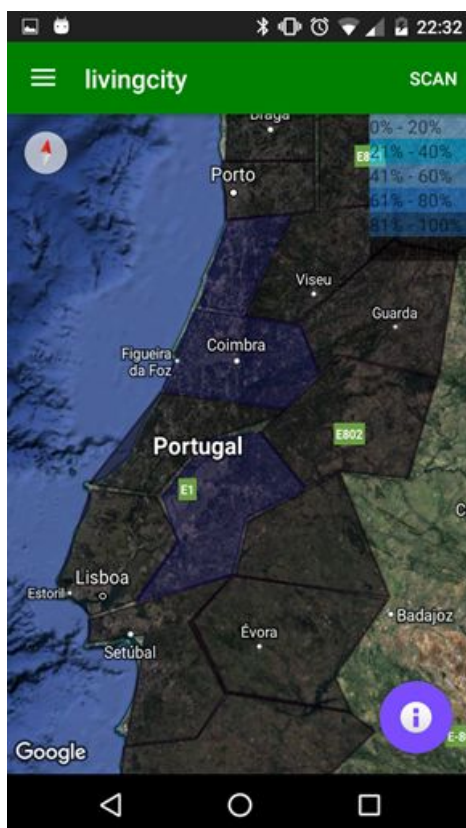
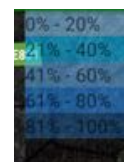


Figura 20

O presente fragmento é acedido através da entrada no menu (sidebar) com o nome *Humidity managemnt*.

Como é possível observar, neste fragmento o mapa de portugal encontra-se dividido nos diferentes distritos (são usados poligonos para a divisão apresentada). Para os distritos com valores registados será apresentado uma cor que pretende retratar a humidade existente naquela zona. Dada que a humidade é medida em percentagem (%) foi usada a seguintes escala de cores.



Para distritos sem medições, estes são preenchidos pela cor preta (com transparência).

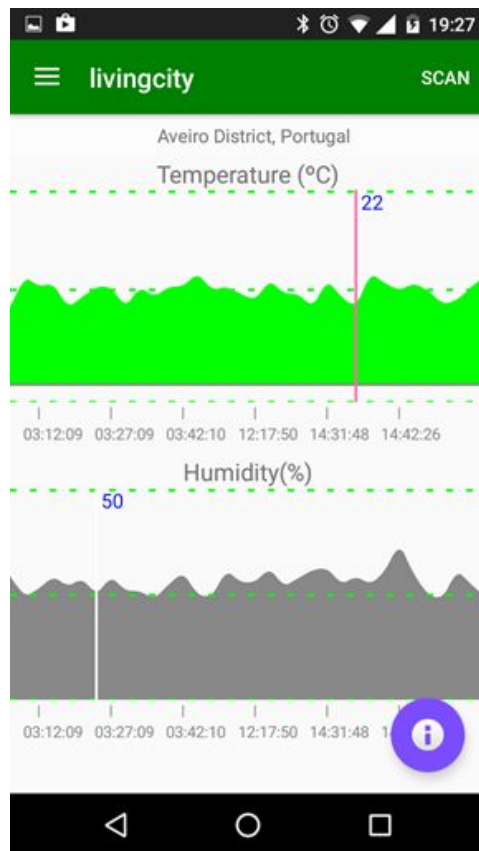


Figura 21



Figura 22

O presente fragmento é acedido através da entrada no menu (sidebar) com o nome *Monitoring sensors*.

Neste fragmento é possível observar a evolução das medições de temperatura e da humidade durante um dia para o distrito da localização do utilizador.

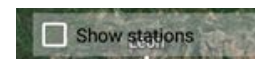
Clicando sobre o gráfico é possível observar o valor de temperatura correspondente à hora de uma determinada medição. É possível ampliar o gráfico utilizando para isso o método comum de ampliação de uma imagem num dispositivo móvel.

O presente fragmento é acedido através da entrada no menu (sidebar) com o nome *Air quality*.

Tal como no fragmento da humidade é possível observar uma divisão do mapa de Portugal em distritos.

Os distritos com cor verde corresponde a um índice de qualidade do ar *Good*, enquanto que a cor vermelha corresponde a um índice de qualidade do ar *Bad*.

Através da seguinte checkbox é possível observar a localização das estações de medição da qualidade do ar.



Os distritos sem qualquer estação de medição é apresentado a cor preta (com transparência).

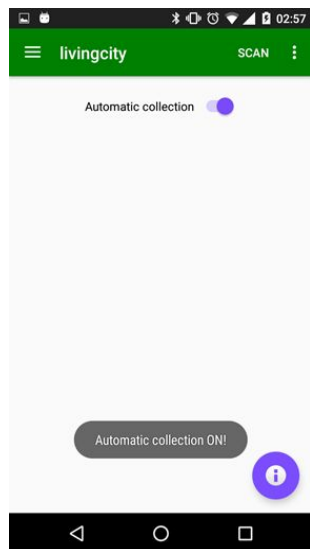


Figura 23

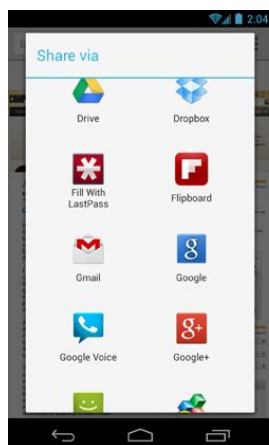
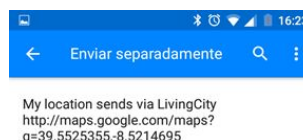


Figura 24



O presente fragmento é acedido através da entrada no menu (sidebar) com o nome *Settings*. Através deste fragmento é possível desativar ou ativar a recolha automática de dados de temperatura e humidade através do dispositivo móvel onde se encontra instalada a aplicação LivingCity. Sempre que uma destas operações é efectuada é dado feedback ao utilizador através de toast.

Ao aceder à entrada no menu (sidebar) com o nome *Share my localization* é possível partilhar a localização do utilizador da aplicação através das aplicações que o utilizador tenha instaladas no seu dispositivo. Através do link que se apresenta na figura seguinte é possível aceder à localização atual do utilizador.

Através de qualquer fragmento é possível efectuar uma medição através do sensor tag da texas. Para tal o utilizador terá que pressionar o botão SCAN para detectar os sensores disponíveis ao alcance, de seguida terá que seleccionar o sensor através do qual pretende recolher a informação. Posteriormente é dado um feedback ao utilizado em como a leitura é efectuada.



Figura 25

10. Organização dos módulos do projeto

Na figura seguinte encontram-se todos os módulos utilizados neste projeto.

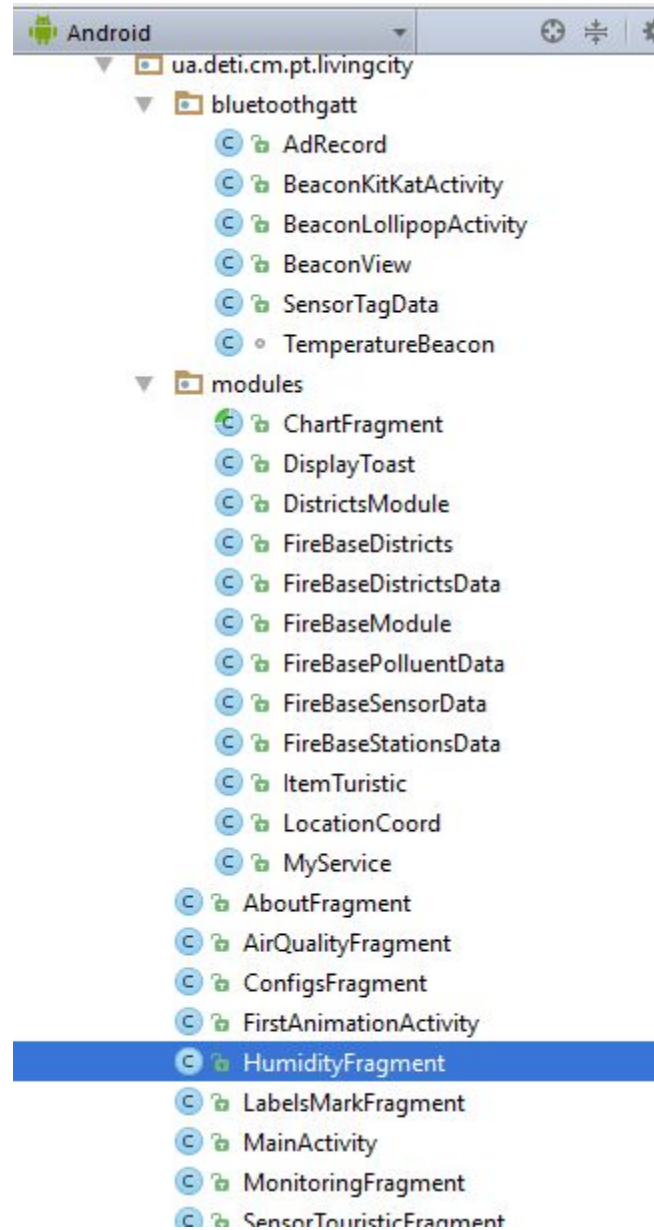


Figura 26: Módulos implementados de todo o projeto

11. Melhorias efectuadas

- A estrutura de dados foi alterada: atualmente, sempre que é inserida uma nova entrada na firebase esta contém também o distrito de onde é enviada a medição
- Foi usado a API do google maps que nos permitiu determinar o distrito tendo por base a coordenadas de localização do utilizador. É usada uma async task para execução do parser do XML
 - <https://maps.googleapis.com/maps/api/geocode/xml?&latlng=LAT,LON>
(Em que LAT: latitude e LON: longitude)
- Mapa apresentado na tab HOME apenas apresenta valores de temperatura (e local onde são medidos) e pontos turísticos
 - Apenas os valores mais recentes serão apresentados neste mapa, para cada distrito
- Valores são lidos automaticamente em background - é utilizado um intentservice. Este valor é lido de 3 em 3 minutos.
 - Na tab settings é possível desativar a recolha automática de valores
 - Os presentes dados recolhidos são respectivamente
 - Temperatura: Device temperature (TYPE_TEMPERATURE) / valores aleatórios
 - Humidade: Ambient relative humidity (TYPE_RELATIVE_HUMIDITY) / valores aleatórios
- A apresentação dos valores de humidade é agora distinto em relação à temperatura (o mesmo não acontecia na última versão deste projeto). Na tab HUMIDITY MANAGEMENT é apresentado um mapa com todos os distritos de Portugal pintados, em tons de azul quando existem valores medidos para este distrito ou preto quando não existem.
 - A divisão de humidade 0-100% foi dividida em 5 escalas de cores diferentes
- Tab MONITORING SENSORS permite visualizar gráficos evolutivos dos valores medidos para temperatura e humidade para o presente dia e para o distrito onde se encontra o utilizador
- Tab AIR QUALITY permite visualizar os diferentes distritos de Portugal e mostrar a qualidade do ar em vez de em cada marker.
 - Os círculos em volta dos markers foram também removidos.
 - Os dois graus possíveis de qualidade são: Bad(vermelho) ou Good (verde)
 - Quando não existem dados é apresentado a cor preta

12. Conclusão

Chegado ao final deste relatório, é nossa intenção efetuar uma retrospectiva da evolução do mesmo, tendo em conta os problemas com que nos deparámos, e principais conclusões retiradas.

Ao usar o conhecimento apreendido durante as aulas conseguimos começar pelos passos básicos a criar a nossa aplicação. Obviamente que ao longo de todo este trajeto as dificuldades encontradas foram inúmeras, destacando a leitura do Sensor BLE da Texas numa 1ª fase e a leitura da Firebase em alguns aspetos devido à complexidade por vezes dos objetos JSON mas também devido à implementação única desta tecnologia da Google.

Tivemos também alguma dificuldade devido à complexidade da nossa aplicação da sincronização atempada da informação vital à nossa aplicação, sendo usado depois AsyncTask e todos os métodos já acima no relatório descritos, com isto declaramos que todos estes pontos foram ultrapassados com sucesso e pensamos ter uma aplicação única, completamente funcional e user friendly para ser utilizada no quotidiano.

De modo a finalizar o presente trabalho, tivemos que ter/obter conhecimentos avançados sobre o fluxo do Android e sobre o funcionamento e componentes disponíveis no sdk para implementar e criar a aplicação Livingcity. Este conhecimento foi adquirido não apenas nas aulas Computação Móvel, mas também na exploração intensiva em casa de todo o funcionamento aplicacional Android.

No final deste projeto e após as melhorias efectuadas, estamos bastante contente e pensamos ter uma aplicação pronta para ser lançada no mercado com um ambiente gráfico agradável e simples e útil para o utilizador final.

13. Referências e recursos

Repositórios do projeto para aplicação Android:

- Repositório de código (projeto android):
<https://github.com/ruipoliveira/livingCity-Android>
- Repositório de código (backend):
<https://github.com/ruipoliveira/firebaseGraphicsViews>

Referências utilizadas neste projeto:

- Firebase Guides: <https://www.firebase.com/docs/web/guide/saving-data.html>
 - <https://www.firebase.com/docs/web/guide/retrieving-data.html>
 - YouTube tutorial vídeos: <https://goo.gl/v8tbUk>
- Animation-list:
<http://stackoverflow.com/questions/17372214/android-how-to-set-width-of-animation-list-or-items-in-it>
- Info Sharing:
 - <http://stackoverflow.com/questions/2091465/how-do-i-pass-data-between-activities-on-android>
 - <http://stackoverflow.com/questions/7149802/how-to-transfer-some-data-to-another-fragment>
- CheckBox: <http://developer.android.com/guide/topics/ui/controls/checkbox.html>
- Async Task:
<http://stackoverflow.com/questions/14250989/how-to-use-async-task-correctly-in-android>
- Google Maps: <https://developers.google.com/maps/get-started/>
 - Circles:
<https://developers.google.com/android/reference/com/google/android/gms/maps/model/Circle>
 - Marker:
<https://developers.google.com/maps/documentation/android-api/marker>
- Partilha da localização do utilizador:
<https://developer.android.com/training/sharing/send.html>
- Gráficos para monitorização: <https://github.com/blackfizz/EazeGraph>