

Universidade do Minho - Escola de Engenharia

Mestrado Integrado em Engenharia Informática

Sistemas Autónomos

Trabalho prático

Sensorização Ambiente com Smartphone

Carlos José Gomes Campos a74745

José Pedro Ferreira Oliveira a78806

Vitor José Ribeiro Castro a77870

1 de Abril de 2019

Resumo

Este relatório pretende descrever os passos necessários à resolução do desafio proposto pela equipa docente. Todo o processo de escolha de sensores, o seu mapeamento, *storage* e partilha na rede, no âmbito da unidade curricular de Sistemas Autónomos, do perfil de Sistemas Inteligentes, do Mestrado Integrado em Engenharia Informática.

Conteúdo

1	Introdução	3
1.1	Contextualização	3
1.2	Caso de Estudo	3
1.3	Estrutura do Relatório	3
2	Arquitetura e Funcionamento	4
2.1	Sensorização	5
2.2	<i>Backend - Firebase</i>	7
2.3	<i>Frontend</i> - Gráficos e Sugestões	10
3	Conclusão	14

1 Introdução

1.1 Contextualização

O *smartphone* é um dispositivo usado por grande maioria da população mundial, em países desenvolvidos. Com esta disponibilidade, o grupo concluiu que, com vista à obtenção de dados ambiente, este podia ser utilizado com eficácia.

Para assistir o utilizador na manutenção de condições propícias à sua atividade e saúde física, os dados recolhidos serão disponibilizados online, através da plataforma *Firebase*. Esta permite, por exemplo, a partilha dos dados para ambientes inteligentes, como casas, escritórios, ou outros ambientes relacionados. Serão também apresentadas sugestões através de notificações.

1.2 Caso de Estudo

O projeto tem por base a conceção de uma solução para a resolução do desafio proposto, que pretende implementar um sistema capaz de monitorizar e recolher leituras de sensores físicos e/ou virtuais, possibilitando a geração de avaliações/sugestões em função do estado do ambiente em análise, no desenvolvimento e programação de sistemas autónomos, no campo de ambientes inteligentes. Numa primeira fase a recolha de dados, seguida de envio desses para um *backend*, com posterior serviço de visualização e sugestão ao utilizador.

O grupo pretende conseguir recolher fatores relativos ao local onde o utilizador se encontra, com auxílio do *smartphone*, facilitando o ajuste do local onde este se encontra às condições ideais.

1.3 Estrutura do Relatório

O relatório está dividido em três partes, correspondentes às três tarefas principais do problema, a saber, a recolha de dados, seguida do seu envio para a *cloud* e, por fim, representação no *smartphone* e sugestão. Estes constituem, respetivamente, *backend*, *backend* e *frontend*.

Na secção sobre a sensorização, vão ser descritos os sensores recolhidos, bem como o que levou o grupo a optar por estes.

Na secção referente ao *backend*, vai-se mostrar como se guardaram os dados relativos aos sensores guardados, bem como as estratégias necessárias para permitir o sistema ser usado por vários utilizadores, simultaneamente.

Na secção relativa ao *frontend*, apresentam-se as sugestões e interface que permite ver o histórico de recolhas. Esta secção é de extrema importância pois permite conhecer que tipo de dados são recolhidos e tem um impacto mais claro e direto em relação ao utilizador da aplicação.

2 Arquitetura e Funcionamento

A arquitetura escolhida passou pela implementação de uma aplicação *Android* que seria o centro da recolha de dados. O *smartphone*, sendo um dispositivo comumente usado pela maioria dos utilizadores nos países desenvolvidos, permite conhecer o local onde estes se encontram devido aos seus sensores.

Cada telemóvel tem um conjunto de sensores físicos e virtuais, dos quais se podem recolher dados frequentemente. Estes dados, quando recolhidos, seriam guardados localmente (no dispositivo) e na cloud (pelo *Firebase*).

Guardar estes dados localmente permitia, por exemplo, executar verificações instantâneas aos valores dos parâmetros recolhidos, oferecendo gráficos e sugestões úteis ao utilizador. Com recurso aos gráficos, o utilizador conseguia acompanhar o seu uso diário do *smartphone* e as condições em que o usava. Por outro lado, através de sugestões em intervalo de tempo determinado, eram dadas recomendações de melhoria do ambiente, por exemplo, o aumento da luminosidade do espaço presente. Adicionalmente, graças à manutenção dos dados no *Firebase*, que disponibiliza métodos de obtenção de dados, por exemplo, através de REST, é possível que os locais onde estará instalado um sistema de ajuste ambiente, o faça de forma rápida e dinâmica, com recurso a GETs à base de dados em tempo real.

A figura abaixo apresenta uma descrição visual do referido anteriormente. De seguida, explica-se, ponto a ponto, os mais importantes pontos do sistema.

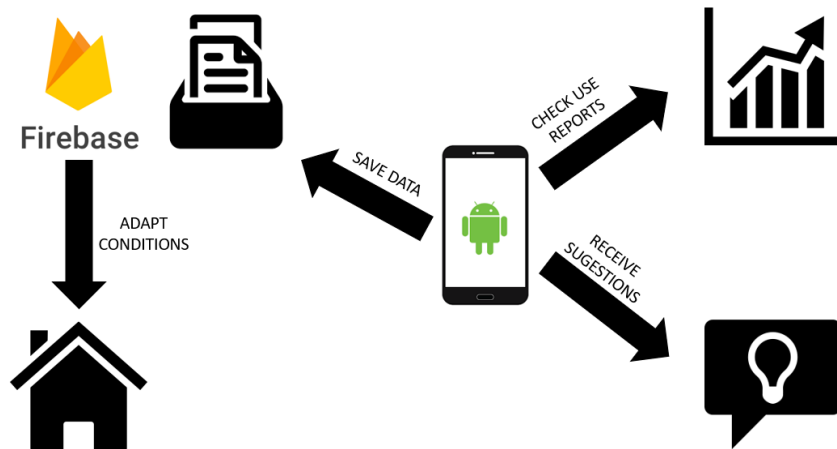


Figura 1: Arquitetura da solução

2.1 Sensorização

Com o objetivo de recolher dados suficientes para detetar o ambiente do utilizador a usar o telemóvel, foi necessário recolher dados de vários sensores. Apesar de nem todos estarem diretamente relacionados com o ambiente em que o utilizador se encontra, eles permitem detetar quando, por exemplo, um utilizador poderá encontrar-se aborrecido ou em consumo de multimédia. Em qualquer um dos casos podem ser feitos ajustes na condição do espaço em que este se localiza, de modo a melhor acomodar a sua atividade (um consumo de multimédia exige, à partida, um ambiente com luz cinematográfica (podia implicar, por exemplo, ligar apenas as fitas LED de uma divisão de uma casa). Tendo em conta estas considerações, foram recolhidos os seguintes dados:

- Nível de bateria (0% a 100%);
- Rotação do ecrã (1=*Landscape* ou 0=*Portrait*);
- Número de notificações (sociais, chatting, outras) recebidas;
- Número de notificações não lidas;
- Número de ativações de ecrã;
- Número de chamadas efetuadas;
- Número de chamadas recebidas;
- Proximidade (0=*Near* ou >0=*Far*);
- Número de SMS recebidas;
- Luminosidade do ambiente (valor em Lx);
- Número de pressões dos botões HOME e RECENT;
- Estado da ligação ao Wi-Fi (1=Ligado ou 0=Desligado);
- Estado da ligação a Redes Móveis (1=Ligado ou 0=Desligado);

Não é demais reforçar que, apesar de nem todos os sensores importarem diretamente informação sobre o ambiente, estes podem contribuir para um enquadramento do utilizador no espaço e, conseqüentemente, permitir adequar as alterações no ambiente inteligente de melhor forma. Para que tal aconteça, foi necessário abrir vários *Listeners* e *Services* disponibilizados em *Android*, algo que exigiu bastante esforço dos membros do grupo. Na figura 2 pode-se verificar a construção de vários elementos necessários à manutenção destes componentes.

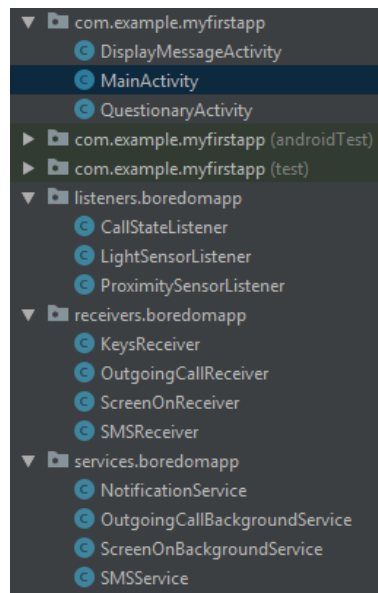


Figura 2: *Packages* da aplicação móvel

2.2 Backend - Firebase

Para dar suporte à aplicação em backend, foi decidido usar o *Firebase*, visto que permite integrar uma base de dados em *Real-Time* de forma muito intuitiva numa aplicação *Android*.

Esta base de dados é do tipo *NoSQL*, sendo que toda ela se trata de um ficheiro *JSON*, tendo uma coleção de *Users*, em que cada entrada se trata da coleção de dados correspondente a cada utilizador da aplicação, representados pelo *IMEI* do seu telemóvel. Por sua vez, esta é uma coleção em que cada entrada corresponde ao *Snapshot* tirado ao smartphone no momento em que o utilizador responde à pergunta "Are you bored?".

A inserção nesta base de dados só acontece porque a aplicação tem uma classe específica, correspondente a essa entrada, denominada por **Entry.java**, em que as suas variáveis de instância estão apresentadas na figura 3, sendo que cada variável corresponde ao valor do sensor representado pelo nome.

```
private String bateria;  
private String nAppSociais;  
private String nAppChatting;  
private String nOutrasApps;  
private String notificacoesAppsAtuais;  
private String nAtivacoesEcra;  
private String nChamadasFeitas;  
private String getnChamadasRecebidas;  
private String proximidade;  
private String nSMSRecebidas;  
private String luminosidade;  
private String orientacao;  
private String nClicksHome;  
private String nClicksRecentes;  
private String wifi;  
private String dadosMoveis;  
private String bored;
```

Figura 3: Variáveis de instância da classe Entry.java

Sempre que o botão *CHOOSE* é pressionado no questão inicial "Are you bored?", como visível na figura 4, a função **question**, da **MainActivity.java** é invocada, e insere os dados dos sensores através deste comando:

```
myRef.child(imei).child(System.currentTimeMillis()+"")  
    .setValue(entrada);
```

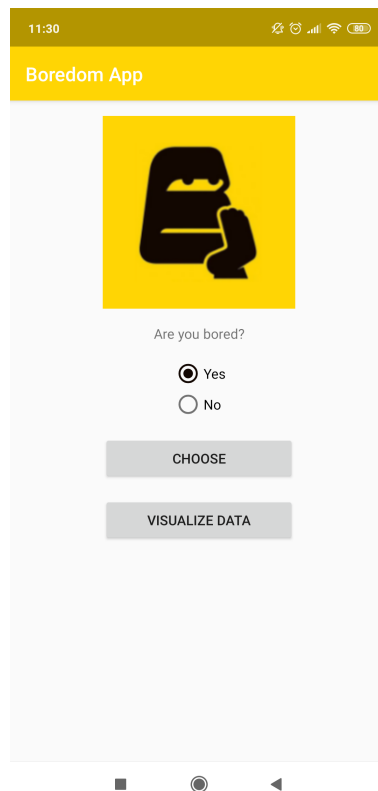


Figura 4: Questão sobre aborrecimento do utilizador

Sendo que o **myRef**, é a referência que aponta para a base de dados, o **imei** corresponde ao *IMEI* do telemóvel, e cada entrada é dada como chave, o **currentTimeMillis** do sistema, garantindo assim unicidade da mesma.

Na figura 5, pode se ver a estrutura da base de dados anteriormente descrita.

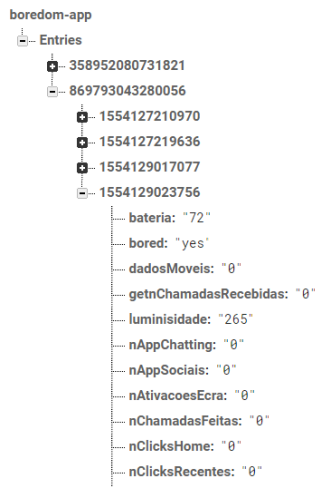


Figura 5: Estrutura da base de dados

Por fim, nesta secção convém referir que os dados também são escritos em um ficheiro interno, denominado por *data.txt*, caso a conexão ao *Firebase* esteja impossibilitada de leitura, e assim sempre que o utilizador consultar os gráficos de dados, fornecidos pela **ViewListActivity.java**, pode o fazer em offline.

2.3 *Frontend* - Gráficos e Sugestões

Qualquer programa, por maior que seja o trabalho desenvolvido no seu *backend*, necessita de um *frontend* que permita, no mínimo, dar informações relevantes ao utilizador. No caso em específico, e tendo em conta que são registados os valores guardados em cada *snapshot*, que acontece quando pressionado o botão **CHOOSE**, como se mostra na figura 4, seria interessante mostrar um gráfico com a evolução dos valores recolhidos. Para isso, o grupo desenvolveu a atividade relativa à figura 6.

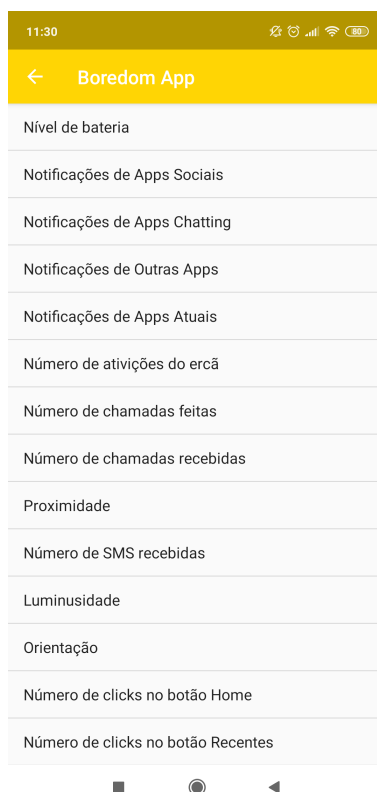


Figura 6: Gráficos disponíveis para consulta do utilizador

No caso, para exemplo representativo, usa-se os valores de luminosidade, que podem bem representar a mudança de ambientes de um utilizador. É bastante interessante de verificar a evolução conseguida, sendo que neste caso específico se passou de um local exterior para uma sala de aula, entre locais mais próximos da saída e diretamente abaixo de iluminação artificial. Quando seleccionado um dos gráficos, recebemos a leitura dos seus valores dos últimos 10 registos, caso este número exista, ou todos os existentes até ao momento, quando não temos 10 registos.

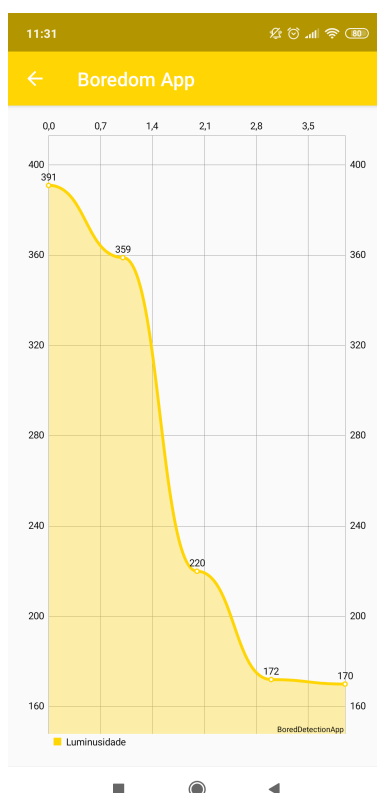


Figura 7: Gráficos referente à luminosidade

Adicionalmente, como o objetivo do trabalho é personalizar o ambiente e ajustar o mesmo tendo em conta as necessidades do utilizador que usa a aplicação, seria importante dar algum *feedback* direto. A solução que se encontrou para promover uma alteração de comportamento ou chamada de atenção ao utilizador, para além de disponibilizar dados que alterariam o ambiente em que o utilizador se encontra, foram as notificações. Determinados certos parâmetros, num intervalo de tempo especificado, apenas em momentos de uso do telemóvel, poderia ser sugerido que o utilizador está apenas a perder tempo e que devia fazer outra coisa ou até que a luminosidade do local onde se encontra pode ser prejudicialmente baixa. As figuras 8 e 9

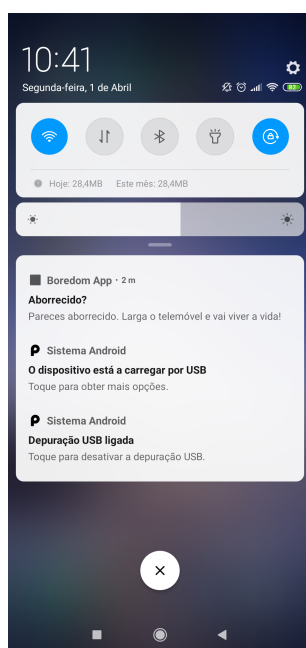


Figura 8: Sugestão de aborrecimento detetado

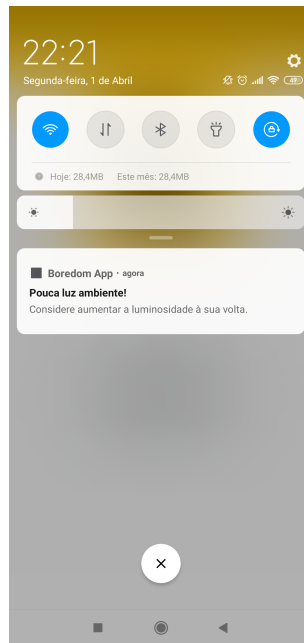


Figura 9: Sugestão de condições de luz prejudiciais

Assim, conseguiu-se fornecer dados que podem apresentar-se relevantes para o utilizador, através de gráficos, bem como promover alterações na sua atividade atual de forma dinâmica, através de notificações.

3 Conclusão

O grupo apresentava-se expectante em relação a este trabalho. Se, por um lado, um dos elementos já tinha alguma experiência em programação *Android*, por outro os outros elementos ainda precisavam de ser contextualizados. Inicialmente, foi proposto um trabalho de integração de interação com *beacons*, em que na proximidade de um destes dispositivos o utilizador da aplicação seria notificado sobre, por exemplo, oportunidades de compra (caso estivesse dentro de um shopping) ou informações espaciais (sobre a ocupação de uma biblioteca da universidade, caso estivesse na mesma). No entanto, devido a novas restrições causadas pelo **RGPD** e implementadas pelo *SDK 28* do *Android*, esta ideia teve que ser abandonada. Das duas semanas projetadas inicialmente para a resolução do projeto, restaria então apenas uma para a implementação do trabalho que se encontra descrito neste relatório. De facto, a equipa tentou solucionar o problema que havia tido com a deteção dos *beacons* durante uma semana.

Tendo em conta a experiência adquirida nesta de semana de tentativa de resolução de problemas, em que a equipa partilhou de várias ideias e conhecimentos, decidiu-se por seguir um diferente rumo e escolher utilizar um dispositivo que utilizamos todos os dias, o *smartphone*, como um super sensor ambiente, uma vez que anda sempre com cada utilizador, para onde quer que ele vá. O tempo para realização deste projeto era agora escasso e, impreterivelmente, exigiu uma forte coordenação da equipa. Enquanto um membro tratou de fazer a recolha dos dados dos sensores que se consideraram necessários, um segundo tratou da conexão e guarda desses dados ao *Firebase* e localmente, enquanto que o último implementava os gráficos e as sugestões a ser apresentadas.

O grupo encontrou algumas dificuldades na obtenção de alguns sensores, tendo-as superado com trabalho conjunto. Também encontro dificuldades em conseguir um identificador única para cada dispositivo móvel utilizado, tendo optado por usar o IMEI num momento final. O IMEI é único para cada célula de comunicação e todos os telemóveis o possuem. Por fim, na ordem das dificuldades, os gráficos tornaram-se desafiantes pela necessidade de *parsing* de dados que exigiam, tendo alguns problemas sido solucionados na obtenção e registo dos dados para a memória interna do *smartphone*, enquanto outros com métodos de *parsing* próprios para cada caso.

No geral, o grupo encontra-se orgulhoso do trabalho coordenado que conseguiu, tendo implementado todas as funcionalidades estabelecidas inicialmente, tendo em conta o prazo limitado, e superando até as mesmas. Exigiu dedicação e trabalho mas, no fim, é perceptível a utilidade destes dados e reconhece-se que é possível estender a lista de informação recolhida e melhorar a sensorização ambiente conseguida. Foi um trabalho extremamente útil do ponto de vista de perceção dos dados que são passíveis de ser recolhidos, bem como na importância dos mesmos para uma vida em ambiente inteligente, algo que pode estar incrivelmente mais perto do que o cidadão comum pode pensar.

Referências

- [1] Tutorials Point Android Notifications, https://www.tutorialspoint.com/android/android_notifications.htm
- [2] Tutorials Point Event Handling, https://www.tutorialspoint.com/android/android_event_handling.htm
- [3] Android Developers Notifications, <https://developer.android.com/training/notify-user/build-notification>
- [4] Android Developers Broadcasts, <https://developer.android.com/guide/components/broadcasts.html>
- [5] Android Developers Services, <https://developer.android.com/guide/components/services>
- [6] Android Developers UI Events, <https://developer.android.com/guide/topics/ui/ui-events>
- [7] Android Developers Graphics, <https://developer.android.com/reference/android/graphics/package-summary>
- [8] Medium Mastering Android Context, <https://medium.freecodecamp.org/mastering-android-context-7055c8478a22>
- [9] Firebase Android Read and Write, <https://firebase.google.com/docs/database/android/read-and-write>
- [10] Codepath Basic Event Listeners, <https://guides.codepath.com/android/Basic-Event-Listeners>
- [11] Codepath Creating Custom Listeners, <https://guides.codepath.com/android/Creating-Custom-Listeners>