Gonçalo Emanuel Palma Amaro

Licenciado em Ciências de Engenharia Eletrotécnica
e de Computadores

Desenvolvimento de uma aplicação Android para pacientes sob tratamento

Dissertação para obtenção do Grau de Mestre em
Engenharia Electrotécnica e de Computadores

Orientador: Professor Doutor Yves Rybarczyk, Professor Auxiliar, Faculdade de Ciências e Tecnologias

Júri:

Presidente: Prof. Doutor José Manuel Matos Ribeiro da Fonseca

Vogais: Prof. Doutor Yves Rybarczyk
Prof. Doutor João Almeida das Rosas

FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Setembro, 2014

Desenvolvimento de uma aplicação *Android* para pacientes sob tratamento

Copyright © Gonçalo Emanuel Palma Amaro, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa.

A Faculdade de Ciências e Tecnologia e a Universidade Nova de Lisboa têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objetivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

Aos meus pais

Agradecimentos

Todo o apoio que me foi dedicado ao longo deste projeto teve um impacto fundamental na sua realização. Deixo aqui os meus agradecimentos aos que mais me apoiaram e foram fundamentais para o meu sucesso.

Ao Professor Yves Rybarczyk, meu orientador, que esteve sempre disponível quando precisei da sua ajuda, aconselhando com as melhores abordagens ao tema, e demonstrando ser um orientador exemplar.

A todos os meus amigos que me acompanharam desde que tenho memórias, Ricardo Ribeiro, Vera Bejinha, Isabel Botelho, João Neves, Ana Júlia, que sempre me fizeram acreditar de que era capaz, durante o meu percurso académico.

Aos meus colegas e amigos que tive o maior prazer de conhecer e trabalhar, Carlota Maçaneiro, Bruno Rodrigues, Filipa Matias, Flávio Pascoa, e Júlio Oliveira.

Agradeço a duas pessoas que apareceram na minha vida recentemente e desempenharam um papel fundamental, à Ana Teresa Santos, colega de casa durante este projeto, que me deu a estabilidade e conforto que precisava, e ao André Martins, que me mostrou as minhas capacidade, persuadiu-me a continuar este percurso académico, mostrando as vantagens que me viria a trazer.

Agradeço especialmente à minha família, Maria João Amaro, Vítor Amaro e Mónica Amaro, pois sem eles nunca teria seguido o ensino superior, completando esta etapa da tão importante minha vida.

Resumo

Nos dias de hoje, as tecnologias deixaram de ser tabus para qualquer faixa etária, tornando-se uma necessidade emergente. Os *smartphones* estão entre os dispositivos mais utilizados pela população em geral no quotidiano. Adicionando aos novos estilos de vida atuais, a medicação diária também tem vindo a crescer nos últimos tempos, sendo esta utilizada em casos de hipertensão, colesterol elevado, diabetes ou outras doenças perpétuas, como também na toma de multivitamínicos, medicação para aumento de produtividade, contracetivos ou antibióticos e anti-inflamatórios em casos de constipações ou gripes. Aliado a um estilo de vida célere, a toma de medicação no momento indicado facilmente é esquecida, tornando os tratamentos ineficientes.

A aplicação "Hora do comprimido", desenvolvida para dispositivos *Android*, permite auxiliar os utilizadores na toma da sua medicação, contínua ou temporária, facilitando o quotidiano do utilizador, viabilizando assim um tratamento completo, sem falhas nas tomas ou feitas fora de horas.

Palavras-chave: medicamento, *smartphone,* aplicação *Android,* toma de medicação, base de dados SQL

Abstract

Nowadays, technologies are no longer a taboo for any age group, making it a clear market need. Smartphones are among the most used devices in the general population every day. Adding to the new current lifestyles, daily medication has also been growing in recent times, being used in cases of hypertension, high cholesterol, diabetes or other perpetual diseases, but also in taking multivitamins , medication to increase productivity, contraceptives or antibiotics and anti-inflammatory drugs in cases of colds or flus. Combining with stressful life, take the necessary medication in the right moment is easily forgotten, making the treatments less inefficient. The "Hora do comprimido" application, developed for Android devices, help users to take their medication, ongoing or temporary, everyday, establishing a continuous treatment without flaws both in doses and takeout of correct time.

Keywords: medicine, smartphone, Android application, taking medication, SQL database

Conteúdo

1.INTRODUÇÃO 1
1.1 MOTIVAÇÃO E OBJETIVOS 1
1.2 ORGANIZAÇÃO DA TESE 2
2. ESTADO DA ARTE 5
2.1. TOMA DE MEDICAÇÃO NO QUOTIDIANO 5
2.2. PLATAFORMAS MOBILE 5
2.2. DESENVOLVIMENTO DE APLICAÇÕES ANDROID 9
2.2.1. GOOGLE PLAY 9
2.2.2. ANDROID 9
2.2.3. CAMADAS DE SOFTWARE 9
2.2.4. ARQUITETURA DE UMA APLICAÇÃO 10
2.2.5. CICLO DE VIDA 12
2.2.6. ECLIPSE E ANDROID SDK 15
2.2.7. MYSQL 15
2.2.8. APLICAÇÕES EXISTENTES 16
3. IMPLEMENTAÇÃO 20
3.1. REQUISITOS FUNCIONAIS 22
3.2. BASE DE DADOS 22
3.2.1. SQLITE DATABASE 22
3.2.2. PHPMYADMIN 28
3.3. CLASSES 28
3.3.1. ALARM 28
3.3.2. ALARM.ALERT 29

3.3.3. ALARM.DATABASE 29
3.3.4. ALARM.PREFERENCES 30
3.3.5. ALARM.SERVICE 30
3.3.6. ALARM.TELEPHONY 30
3.3.7. ONLINE 30
3.3.8. PILLS_CLOCK 31
3.3.9. PERFIL.DATABASE 32
3.3.10. VIEW.VIEWGROUP 32
3.4. INTERFACE DO UTILIZADOR 33
3.3.1. MENU PRINCIPAL 35
3.3.2. GESTOR DE MEDICAMENTOS 36
3.3.3. GESTOR DE ALARMES 38
4. RESULTADOS E CONCLUSÃO 41
4.1. RESULTADOS GRÁFICOS 41
4.2. ANÁLISE DE RESULTADOS 42
4.3. CONCLUSÃO E PERSPETIVAS 45
4.4. TRABALHO FUTURO 45
5. REFERÊNCIAS 47

LISTA DE IMAGENS

FIGURA 1 - CAMADAS DE SOFTWARE 10
FIGURA 2 - DALVIK VIRTUAL MACHINE 10
FIGURA 3 - CICLO DE VIDA DE UMA *ACTIVITY*, ADAPTADO DE [11] 14
FIGURA 4 - FLUXOGRAMA DA ESTRUTURA DA APLICAÇÃO 21
FIGURA 5 - BASE DE DADOS DE MEDICAMENTOS 23
FIGURA 6 - BASE DE DADOS DE ALARMES 24
FIGURA 7 - IMAGEM CARREGADA AUTOMATICAMENTE CASO O MEDICAMENTO ADICIONADO NÃO POSSUA IMAGEM 24
FIGURA 8 - BASE DE DADOS DE MEDICAMENTOS *ONLINE* 25
FIGURA 9 - TABELAS PRESENTES NA BASE DE DADOS REMOTA 26
FIGURA 10 - DIAGRAMA UML DA CLASSE PILL_CLOCK 26
FIGURA 11 - DIAGRAMA UML DA APLICAÇÃO 27
FIGURA 12 - INTERFACE DE UTILIZADOR DISPONIBILIZADA NO PHPMYSQL 28
FIGURA 13 - REPRESENTAÇÃO DA HIERARQUIA QUE DEFINE UM *LAYOUT* 33
FIGURA 14 – DIAGRAMA REPRESENTATIVO DAS INTERFACES DA APLICAÇÃO 34
FIGURA 15 – (A) REPRESENTAÇÃO DO MENU OCULTO NA PÁGINA INICIAL DA APLICAÇÃO,
(B) REPRESENTAÇÃO O MENU INICIAL 35
FIGURA 16 - REPRESENTAÇÃO DO MENU DEFINIÇÕES 36
FIGURA 17 – (A) REPRESENTAÇÃO DO MENU "ESCOLHA DE MEDICAMENTOS",
(B) REPRESENTAÇÃO DA LISTA DE MEDICAMENTOS DISPONÍVEL *ONLINE* 37
FIGURA 18 - REPRESENTAÇÃO DO MENU "ADICIONAR MEDICAMENTO" 37
FIGURA 19 - (A) - REPRESENTAÇÃO DO MENU "ADICIONAR MEDICAMENTO"
(B) - REPRESENTAÇÃO DO MENU DESCRITIVO DO MEDICAMENTO SELECIONADO
(C) - REPRESENTAÇÃO DO MENU DE CONFIGURAÇÃO DO ALARME 38
FIGURA 20 – REPRESENTAÇÃO DO MENU APRESENTADO NO DISPARO DE UM ALARME 39
FIGURA 21 – (A) – REPRESENTAÇÃO DA LISTA DE ALARMES
(B) – EXEMPLO DA ELIMINAÇÃO DE UM ALARME 39
FIGURA 22 - TOTAL DE INSTALAÇÕES ENTRE ABRIL DE 2014 E AGOSTO DE 2014 41
FIGURA 23 - INSTALAÇÕES ATUAIS POR DISPOSITIVO 42
FIGURA 24 - INSTALAÇÕES ATUAIS POR DISPOSITIVO POR VERSÃO DO ANDROID 42
FIGURA 25 - COMENTÁRIOS REGISTADOS NO GOOGLE PLAY À APLICAÇÃO DESENVOLVIDA 43
FIGURA 26 - REGISTO DE FALHA JAVA.LANG.NULLPOINTEREXCEPTION ENTRE 24/05/2014 E 29/06/2014 44
FIGURA 27 - RELATÓRIO DE ERRO DE FALHA *JAVA.LANG.NULLPOINTEREXCEPTION* 44

Lista de Tabelas

TABELA 1 - VANTAGENS E DESVANTAGENS ENTRE A PLATAFORMA *ANDROID* E *IPHONE* 8
TABELA 2 - *MYPILLBOX* - APLICAÇÃO TRANSFERIDA DO *GOOGLEPLAY* PARA ANÁLISE 16
TABELA 3 - *MEDISAFE* - APLICAÇÃO TRANSFERIDA DO *GOOGLEPLAY* PARA ANÁLISE 16
TABELA 4 - *PILLBOX* - APLICAÇÃO TRANSFERIDA DO *GOOGLEPLAY* PARA ANÁLISE 17
TABELA 5 - *MED MINDER* - APLICAÇÃO TRANSFERIDA DO *GOOGLEPLAY* PARA ANÁLISE 17
TABELA 6 - SEU REMEDIO - APLICAÇÃO TRANSFERIDA DO *GOOGLEPLAY* PARA ANÁLISE 17
TABELA 7 - *WHATPILLS* - APLICAÇÃO TRANSFERIDA DO *GOOGLEPLAY* PARA ANÁLISE 18
TABELA 8 - *MEDICAL DROID* - APLICAÇÃO TRANSFERIDA DO *GOOGLEPLAY* PARA ANÁLISE 18
TABELA 9 - *MEDILOG* - APLICAÇÃO TRANSFERIDA DO *GOOGLEPLAY* PARA ANÁLISE 18
TABELA 10 - PONTOS FORTES E FRACOS DAS APLICAÇÕES ANALISADAS 19

Acrónimos

ADT **A**ndroid **D**eveloper **T**ools

ANR **A**pplication **N**ot **R**esponding

AOSP **A**ndroid **O**pen **S**ource **P**roject

API **A**pplication **P**rogramming **I**nterface

APK **A**ndroid **P**ackage

EDGE **E**nhanced **D**ata rates for **G**SM **E**volution

GPRS **G**eneral **P**acket **R**adio **S**ervice

GSM **G**lobal **S**ystem for **M**obile Communications

HTML **H**yperText **M**arkup **L**anguage

HTTP **H**ypertext **T**ransfer **P**rotocol

IBM **I**nternational **B**usiness **M**achines

iOS **i**Phone **O**perating **S**ystem

JSON **J**ava**S**cript **O**bject **N**otation

OEM **O**riginal **E**quipment **M**anufacturer

PHP **H**ypertext **P**reprocessor, originalmente **P**ersonal **H**ome **P**age

RIM **R**esearch **I**n **M**otion

SDK **S**oftware **D**evelopment **K**it

SO **S**istema **O**perativo

SQL **S**tructured **Q**uery **L**anguage

UML **U**nified **M**odeling **L**anguage

URI **U**niform **R**esource **I**dentifier

URL **U**niform **R**esource **L**ocator

VM **V**irtual **M**achine

XML **E**xtensible **M**arkup **L**anguage

1. Introdução

1.1 Motivação e Objetivos

Com a evolução da ciência ao longo dos últimos anos, a expectativa de vida aumentou drasticamente, a população comum sabe quais os melhores métodos e estilos de vida para manter a sua saúde, os antibióticos têm provado ser uma ferramenta indispensável na luta contra doenças, assim como o aparecimento de novos medicamentos e mais eficazes. Com o aumento de novas soluções, a medicação diária tem vindo a crescer nos últimos tempos [1], sendo esta utilizada em casos de hipertensão, colesterol elevado, diabetes ou outras doenças perpétuas, como também na toma de multivitamínicos, medicação para aumento de produtividade, contracetivos ou antibióticos e anti-inflamatórios em casos de constipações ou gripes.

Assim como a evolução da ciência, paralelamente a evolução das tecnologias cresceu exponencialmente [2], sendo um exemplo a popularidade dos *smartphones*, que encontra-se continuamente a cresce, todos os dias. Devido ao seu formato portátil e acrescendo das suas capacidades de processamento atuais, cada vez mais são utilizados para organização do dia-a-dia, tanto pessoal como profissional do utilizador. A comunicação banal que existia através de uma chamada telefónica ou por uma mensagem escrita foi substituída por outros meios disponibilizados por inúmeras aplicações que existem atualmente, como o *skype*, o *gtalk*, *messenger* do *Facebook*, *Viber*, *WhatsApp* entre outros. A utilização das redes sociais ou mesmo passar algum tempo de distração com um jogo aquando se espera um transporte ou se permanece numa fila de espera, tem sido cada vez mais uma rotina diária dos utilizadores de *smartphones*. Esta tendência evolutiva da utilização dos *smartphones* também se estendeu à gestão pessoal do utilizador, como uma lista de tarefas a realizar diariamente. Uma vez que a gestão pessoal

de saúde por vezes pode-se tornar difícil de gerir, pretende-se utilizar os *smartphones* como solução para a gestão de toma de medicamentos diária. Com o aumento do número de medicamentos que são tomados diariamente por uma grande parte da população, a introdução desta necessidade na rotina torna-se por vezes difícil de gerir. Uma vez que os *smartphones* foram completamente banalizados no quotidiano [3], a aplicação apresentada, "Hora do comprimido", desenvolvida para dispositivos *Android*, permite auxiliar os utilizadores na toma da sua medicação, contínua ou temporária. Contem várias opções como registar um medicamento, sendo possível adicionar uma fotografia do mesmo através da câmara do dispositivo, assim como o seu nome e descrição, ou consultar uma lista de medicamentos que se encontra numa base de dados externa disponível *online,* e também gerir os alarmes de medicação como o utilizador preferir, podendo definir a hora de toma, assim como o intervalo de tempo entre tomas, e personalizar um tom de alarme para cada medicamento. Uma outra função permite que a cada disparo de alarme, avise outro utilizador, através do envio de uma mensagem escrita, possibilitando o acompanhamento do utilizador por parte de outrem. Esta função foi desenhada a pensar nos adultos mais velhos que vivem de forma independente em suas casas particulares, sendo muitas vezes responsáveis pela própria ingestão diária de medicamentos, tornando-se por vezes difícil de gerir fora do ambiente clínico.

1.2 Organização da tese

Esta tese encontra-se dividida em seis capítulos, sendo eles *Introdução, Estado da Arte, Implementação, Resultados, Conclusão e perspetivas* e *Referências bibliográficas,* seguindo-se por esta ordem.

O primeiro e presente capítulo deste documento apresenta o paradigma da toma de medicação na atualidade, e como a aplicação desenvolvida inserida nas tecnologias existentes se enquadram, bem como o seu acrescento de valor e resolução dos problemas que se propõe a resolver.

O segundo capítulo pretende situar o leitor no estado atual da toma de medicação por parte população em geral, bem como das tecnologias utilizadas no desenvolvimento de aplicações *mobile*.

No terceiro capítulo apresenta-se o modelo proposto para a aplicação, todas as funcionalidades e estrutura do sistema, documentado sob a forma de diagramas *UML* de modo a elucidar graficamente. Neste capitulo também é apresentada a interface da aplicação, do ponto de vista do utilizador.

No quarto capítulo analisam-se os resultados obtidos de modo a retirar conclusões e críticas, assim como o *feedback* dado pelos utilizadores que testaram a aplicação. Neste capítulo também se encontra uma síntese do projeto desenvolvido, e abordam-se futuras atualizações que possam ser realizadas na aplicação.

O quinto capítulo finaliza esta tese, apresentando as referências bibliográficas, de contribuição fundamental para a conceção deste projeto.

2. Estado da Arte

2.1. Toma de medicação no quotidiano

Várias investigações demonstram que a gestão da toma de medicação pode ser um desafio [4] quando se trata de doenças específicas, como hipertensão ou diabetes, ou em situações práticas, como quando um paciente se encontra em viagem. No mercado dos medicamentos, um dos aspetos relevantes tem como base estratégias que levam em conta um conjunto de atividades do quotidiano dos pacientes, de modo a facilitar os seus tratamentos [5].

Com a banalização das novas tecnologias no quotidiano, nomeadamente os *smartphones*, várias formas de gerir esta prática têm vindo a surgir, seja no registo de anotações, ou o uso de aplicações específicas para este fim [6].

2.2. Plataformas Mobile

Os mais recentes *smartphones* são dispositivos portáteis que combinam as melhores características de um telemóvel e de um computador. A classificação de um *smartphone* é determinada de acordo com o seu sistema operativo (SO) presente no dispositivo. Os SO's mais proeminentes incluem a *iPhone* (iOS), *Google* (*Android*), *Blackberry* (RIM) e *Microsoft* (*Windows Mobile*). O *Android* é atualmente líder no mercado, com 44% das vendas de *smartphones* no primeiro trimestre de 2014 [7], registando um crescimento anual de 24%. Os projetos *Android Open Search (AOSP)* albergam uma participação no mercado de 13% registada no 1º trimestre de 2014, perfazendo o valor de 57% no mer-

cado. O colapso da *BlackBerry Ltd* em 2013 está a provocar a sair do mercado global de *smartphones* mantendo-se apenas *a Android, iOS* e *Windows Phone*.

Foram registados cerca de 187 milhões de dispositivos *Android*, vendidos no 1 º trimestre de 2014, um aumento de 150.620 mil dispositivos durante o mesmo trimestre de 2013, com um crescimento anual de 24%. Prevê-se que o *Android* irá continuar a dominar com o aumento de *smartphones* nos mercados em desenvolvimento, ampliando a sua quota de mercado, que regista 80% de todos os *smartphones* e 65% dos *tablets* no mundo . A *Google* está a produzir os maiores lucros, com milhões de utilizadores registados mundialmente, que abandonaram os antigos telemóveis para adotar novos *smartphones android*.

A Apple vendeu cerca de 43,7 milhões de *iPhones* no 1 º trimestre de 2014, uma queda de 11% em relação a 2013, onde foram vendidos 51.040 mil *iPhones*. No entanto, a Apple registrou um forte crescimento anual de 17%, devido ao bom desempenho do *iPhone 5S* . Com o lançou do *iPhone 5C* de baixo preço no ano anterior, pretendia-se atrair mais clientes de mercados emergentes, mas o dispositivo fez pouco sucesso a aumentar o volume de vendas. Nos próximos anos prevê-se uma queda do *iOS* no mercado, apesar da popularidade nos EUA, Japão e Austrália.

A parcela de mercado do *Windows Phone* ganhou um bom crescimento de 16% no 1 º trimestre de 2014 através da venda de 13,27 milhões de dispositivos. Registou um crescimento anual de 119% duplicando assim comparativamente a 2013, no entanto, a sua quota de mercado de sistemas operacionais é de apenas 3% a nível mundial. O crescimento de 1% desde o último trimestre mostra a aceitação do *Windows Phone*. A marca *Nokia*, propriedade atual da *Microsof*t, lançou vários modelos *low-end* e *high-end* para atrair todos os segmentos de clientes. A Microsoft ainda investiu no fabrico de *tablets* e *smartphones*, para aumentar a participação no mercado. A atualização do *SO Windows Phone* 8.1 tem recebido uma boa resposta, graças às modificações árias e melhorias.

A *Blackberry* foi oficialmente eliminada do mercado consumidor, com a sua quota de mercado em 0%. Apenas 750.000 dispositivos *Blackberry* foram vendidos no 1 º trimestre de 2014, comparativamente aos 5.4 milhões no ano anterior. A nova versão do SO *BlackBerry* chamado *BlackBerry 10* também foi um fracasso colossal, uma redução de 44% em termos homólogos de quota de mercado. A empresa considerou sair do negócio de *smartphones* e concentrar-se no fornecimento de *software* de defesa e outras aplicações.

As vendas de dispositivos móveis básicos caíram de 44% durante um ano e 24% desde o último trimestre. Os telemóveis mais básicos perderam a sua quota de mercado em 5% em 2013, permanecendo em 30%. Durante o mesmo período do ano passado, as

vendas de telemóveis mais simples foram muito mais elevadas do que de dispositivos *Android*. No 4 º trimestre de 2013, as vendas de telemóveis simples foram de 167,3 milhões, superando os dispositivos *Android* pela primeira vez. Estes números provam que o *Android* vence no mercado móvel simples, que se encontra em declínio.

A popularidade do *Android* não vai ser superada por um longo tempo. Pela primeira vez os utilizadores de *smartphones* mundialmente estão a utilizar dispositivos *Android* para uma ampla gama de aplicações e interface de usuário simples. Enquanto isso, o *Windows Phone* é uma opção atraente para as OEMs, de modo a lucrar em países onde o mercado *Android* está saturado. No entanto, o momento da entrada é demasiado tarde, pois a quota de mercado do *Android* é bastante elevada. Os utilizadores também permanecem cautelosos com as restrições de interface do utilizador e da falta de aplicações. No ritmo atual, a *Microsoft* poderá esperar ganhar quota de mercado de 12% em 2017 e competir com a *Apple* para o segundo lugar.

Perante a situação atual do mercado, as duas melhores opções de investimento demonstradas restringem-se à *Google Android* e *Apple iPhone*, procedendo assim a uma análise geral a ambas, de modo a determinar qual a melhor plataforma para desenvolver a aplicação pretendida, concluindo que a *Android* revela ser mais vantajosa para os programadores. Na tabela 1 registou-se as vantagens e desvantagens da plataforma *Android* e *IOS,* comparativamente.

Tabela 1 - Vantagens e Desvantagens entre a plataforma *Android* e *iPhone*

Android	iPhone IOS
Linguagem de programação Java, sendo uma tecnologia *open-source* e orientada a objetos	Linguagem de programação *Objective-C,* única e mais complexa comparativamente a *Java,* sendo menor a quantidade de documentação existente
Multi-tasking, suporta processamento em paralelo	Os dispositivos mais recentes já suportam *Multi-tasking,* processamento em paralelo
Grande variedade de dispositivos, o que trás maior competitividade no mercado, tornando os equipamentos mais económicos	Poucos dispositivos, apenas da *Apple*, tornando-se bastante dispendiosos
As aplicações não são validadas, o que pode levar a problemas de segurança e desempenho	Aplicações disponíveis mais fidedignas e seguras
Grande variedade de aplicações disponibilizadas	Número de aplicações gratuitas mais restrito
O desempenho do dispositivo diminui bastante com o aumento de aplicações instaladas	Os processos *Multi-tasking* são limitados, assegurando a fluidez do dispositivo
A aquisição de uma licença vitalícia no valor de 25 Dólares permite disponibilizar aplicações de forma ilimitada no *Google Play*	É necessário adquirir uma licença anual, no valor de 99 Dólares, para testar as aplicações desenvolvidas nos dispositivos da *Apple* e disponibilizar na *App Store*
Bastante documentação oficial, guias bem exemplificados e explícitos	Documentação disponibilizada após aquisição da licença de *Developer*

2.2. Desenvolvimento de aplicações Android

2.2.1. Google Play

A plataforma *Google Play* [8] permite a venda e distribuição de aplicações exclusivas para *Android.* É possível um utilizador registar-se gratuitamente e usufruir de todas as aplicações que se encontram na plataforma, mas também caso pretenda divulgar as suas próprias aplicações como programador, bastando adquirir uma licença vitalícia paga para o efeito. Como programador, a plataforma oferece funcionalidades, como sugestões de otimização das aplicações, traçar dados estatísticos das aplicações submetidas, como o número de *downloads*, quais os dispositivos em que foram instalados assim como registo de comentários e classificação, e também disponibilizar a aplicação por um custo determinado pelo programador.

2.2.2. Android

O sistema *Android,* atualmente, possui a maior percentagem de dispositivos móveis em mais de 190 países [9] vindo a registar um crescimento acentuado desde a sua comercialização até aos dias de hoje. Sendo um sistema baseado em *Linux open source*, permitiu um progresso rápido, registando atualmente uma vasta gama de *hardware, software* e operadoras distintas, tornando-se um dos favoritos para consumidores e programadores. Esse crescimento também é visível pela plataforma *Google Play*, onde o aumento de aplicações tem registado um forte crescimento, assim como os *downloads* de aplicações.

2.2.3. Camadas de Software

A plataforma *Android* é o produto do *Open Handset Alliance*, um grupo liderado pelo *Google* [10] que inclui operadores de telefones móveis, fabricantes de aparelhos portáteis, fabricantes de componentes, provedores de plataformas e soluções de *software* e empresas de marketing.

O sistema *Android é* um ambiente em camadas baseado em *Linux*, que inclui uma vasta biblioteca de funções, tornando-se bastante completo, incluindo janelas, visualizações e *widgets* para a exibição de elementos como botões, caixas de edição ou listas de itens, como se encontra representado na figura 1. Ao nível da aquisição de dados, o *Android* inclui várias opções de conectividade como *WiFi*, *Bluetooth* e conexão GPRS, EDGE, 3G ou 4G, como também o suporte para serviços de localização (como GPS) e ainda sensores de posicionamento como acelerômetros e giroscópios, capazes de detetar os movimentos efetuados com o dispositivo.

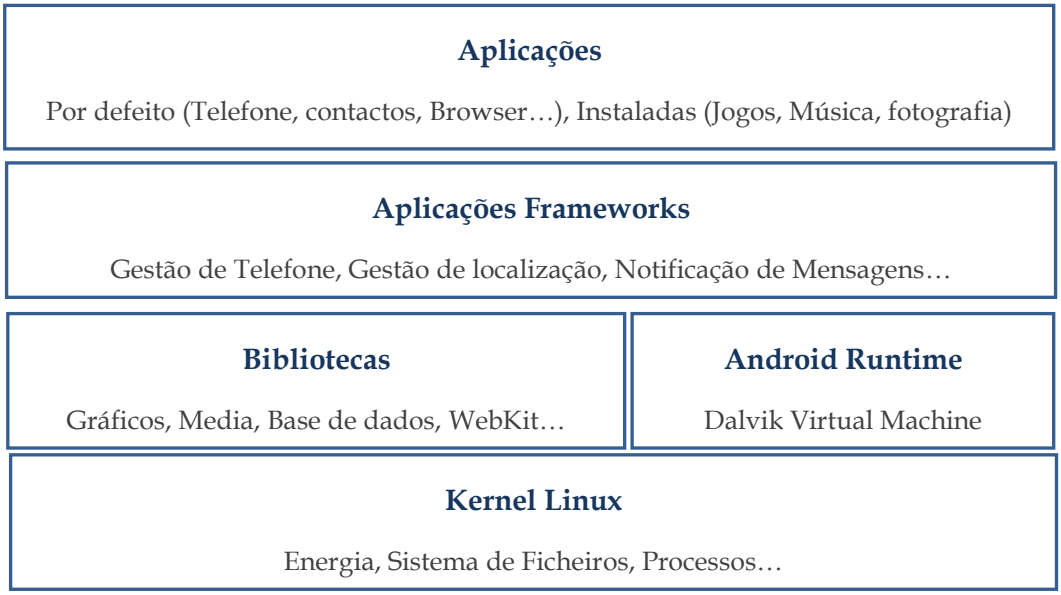


Figura 1 - Camadas de Software

2.2.4. Arquitetura de uma aplicação

A plataforma *Android* é executada sobre um *kernel Linux* e as suas aplicações são desenvolvidas na linguagem de programação Java, sendo executadas numa máquina virtual (*Virtual Machine*). É importante observar que a *VM* não é uma *Java VM*, mas sim uma *Dalvik Virtual Machine*, uma tecnologia de *software* livre. Cada aplicação *Android* é executada numa instância da *Dalvik VM*, que por sua vez, reside num processo gerido pelo *kernel Linux*, representado na figura 2.

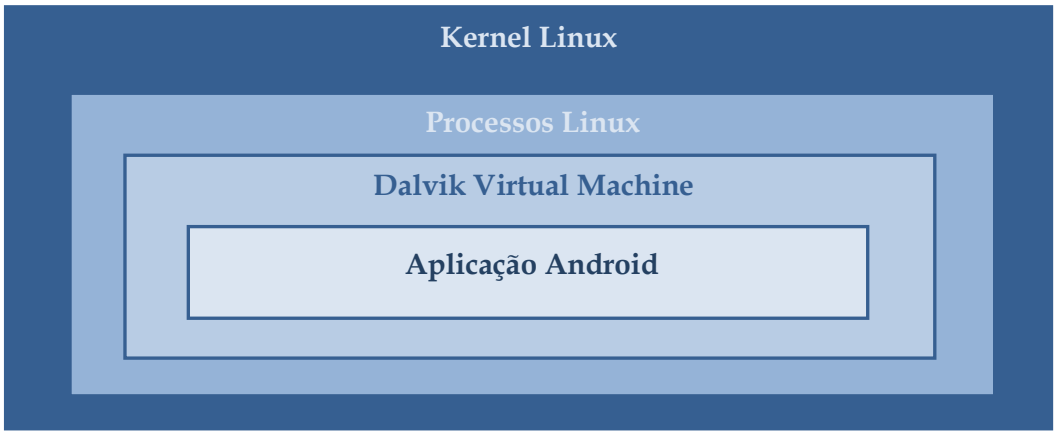


Figura 2 - Dalvik Virtual Machine

Uma aplicação pode ter as seguintes classificações:

- Atividades (*Activity*) **-** Uma aplicação é implementada com uma atividade, ou seja, quando um utilizador seleciona uma aplicação no ecrã inicial, ou da lista de aplicações, a atividade é iniciada.

- Serviços (*Service*) - Um serviço deve ser utilizado por qualquer aplicação que necessite manter-se ativo por um longo período de tempo, como um monitor de rede ou uma aplicação que verifique atualizações.

- Provedores de conteúdo (*Content Provider*) – A função de um provedor de conteúdo é gerir o acesso aos dados que persistem, como uma base de dados *SQLite*. Se a aplicação for muito simples, não é necessário criar um provedor de conteúdo. No caso de uma aplicação de maior dimensão, ou que disponibilize dados para várias atividades ou aplicativos, o provedor de conteúdo será a interface para acesso aos dados.

Processos e *Threads* - Uma aplicação pode ser iniciada para processar um elemento de dados ou para responder a um evento. Um exemplo prático deste tipo de acontecimentos é quando ocorre a receção de uma chamada ou uma mensagem de texto.

2.2.5. Ciclo de vida

Na figura 3 pode-se observar o ciclo de vida de uma aplicação. Uma das mais importantes classes de uma aplicação é a classe *Activity*, sendo esta responsavel pela gestão da interface com o utilizador, pois recebe os pedidos, trata-os e processa-os. A classe *Activity* é constituida pelos seguintes métodos:

onCreate() - É a primeira função a ser executada numa *Activity*. Geralmente é responsável por carregar os layouts XML e outras operações de inicialização. É executada apenas uma vez.

onStart() - É executada imediatamente após a onCreate() e quando uma *Activity* que se encontra em background volta a ter foco.

o*nResume()* - Assim como a onStart(), é invocada na inicialização da *Activity* e quando uma Activity volta a ter foco. A diferença para o metodo *onStart()* encontra-se na sua chamada, esta reinicia-se nas "retomas de foco", ao contrário do metodo anterior, que é invocado quando a *Activity* não se encontra visível, e retorna a ter o foco.

onPause() – É um metodo executado quando uma nova *Activity* é iniciada, ou seja, quando a *Activity* atual perde o foco.

onStop() – É invocado quando a *Activity* fica suprimida por outra *Activity*.

onDestroy() – É o último método a ser executado. A *Activity* é desativada, ou seja, nao pode mais ser relançada. Se o utilizador voltar a requisitar essa *Activity*, um novo objeto será construído.

onRestart() - Executado imediatamente antes do metodo onStart(), quando uma *Activity* volta a ter o foco depois de estar em background.

Em conjunto, estes métodos definem o ciclo de vida de uma atividade, sendo possível analisar três principais fazes de aplicação:

- uma atividade é iniciada entre o método *onCreate()*, onde deve ser realizada a configuração do *layout* e o método *OnDestroy(),* onde se libertam todos os recursos utilizados;
- o tempo de vida de uma atividade acontece entre o método *OnStart()* e *onStop().* Durante este tempo, são mantidos os recursos necessários para mostrar a atividade ao utilizador possibilitando a interação com a aplicação;
- uma atividade pode transitar entre vários planos, estando por cima de todas as outras atividades que se encontram no *display*. A duração de tempo do primeiro plano de uma atividade ocorre entre a invocação do método *onResume()* e do método *OnPause(),* sendo possível que o utilizador interaja com a mesma.

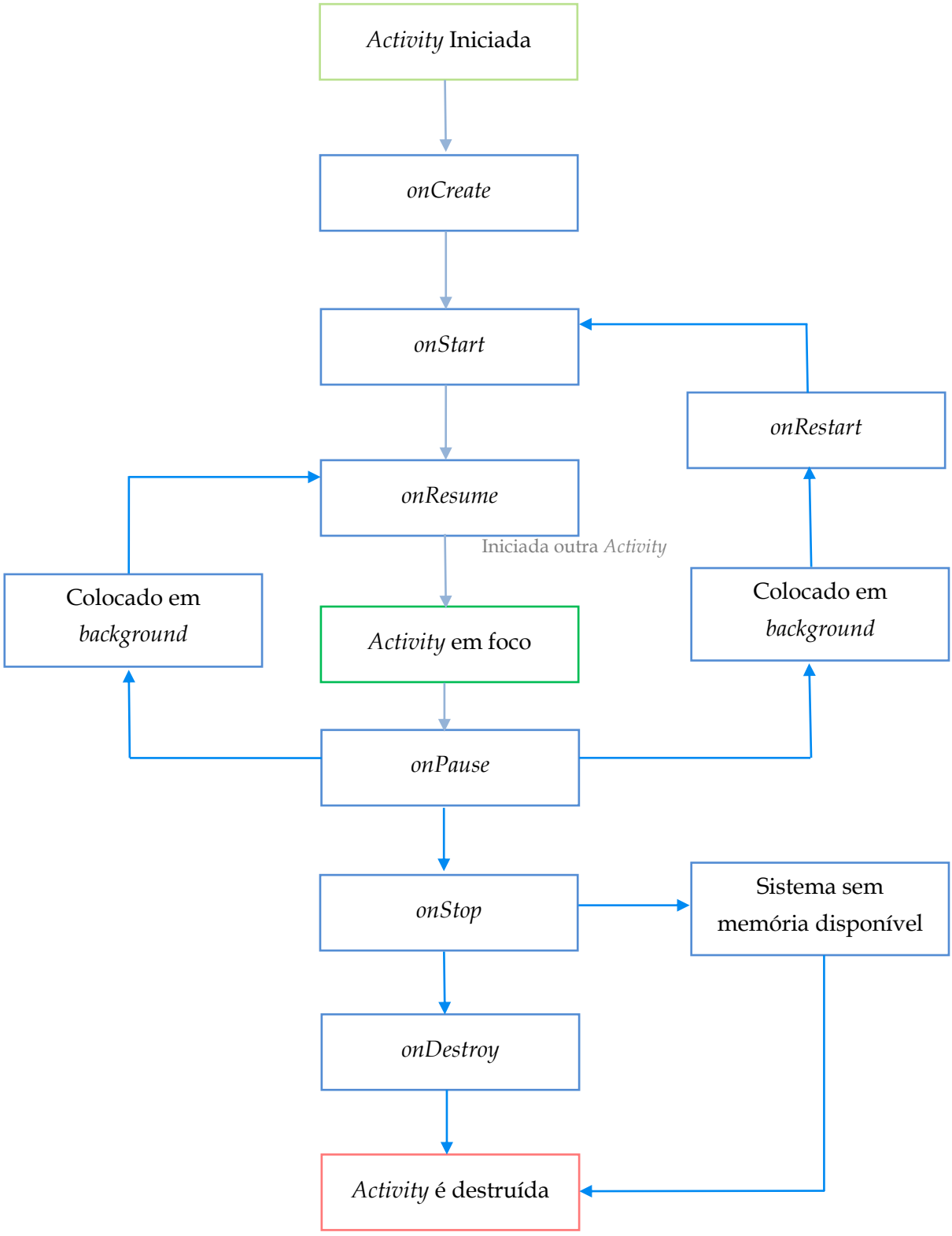


Figura 3 - Ciclo de vida de uma *activity*, adaptado de [11]

Por fim, o ficheiro *AndroidManifest.xml* contém as informações de configuração para executar a instalação no dispositivo. Este inclui os nomes das classes utilizadas ao longo do seu desenvolvimento, e os tipos de eventos que a aplicação irá processar, assim como as permissões necessárias que o utilizador precisa conceder, como o acesso a dados pessoais, lista telefónica ou utilização de sensores de movimento ou as câmaras do dispositivo.

2.2.6. Eclipse e Android SDK

Eclipse é uma plataforma para desenvolvimento de *software open source* que inicialmente foi criada pela IBM [12] nascendo posteriormente a fundação *Eclipse* sem fins lucrativos, onde são hospedados projetos de utilizadores, membros de equipas ou simples curiosos, permitindo criar um fornecedor neutro e *open source.*

O *Android SDK* [13] é um pacote que fornece as bibliotecas de *API* e todas as ferramentas necessárias para desenvolver, testar e fazer *debug* de aplicações para *Android.* De modo a facilitar a instalação destas ferramentas, existe um *plugin* designado de *ADT* [14], que amplia os recursos fornecidos no eclipse, permitindo criar novos projetos ou construir uma interface de um modo mais prático, fazer *debug* num dispositivo físico e exportar a aplicação criada através de um pacote APK.

2.2.7. MySql

MySQL é o sistema de gestão *open source* mais conhecido e utilizado, disponibilizado pela *Oracle Corporation* [15].

Uma base de dados é uma estrutura de dados organizada de forma a facilitar a sua consulta e manipulação, que pode ser constituída por uma simples lista de nomes, um conjunto de imagens ou uma rede que contenha um grande conjunto de informações de uma empresa. Uma estrutura adequada para uma base de dados deve dividir a sua informação por tabelas, sendo estas um conjunto de dados dispostos num número finito de colunas e número ilimitado de linhas. Cada linha corresponde a uma instância de uma determinada atividade ou conceito, neste caso concreto, um medicamento ou alarme são exemplos práticos. Os componentes de cada linha correspondem aos atributos de cada instância, como o nome ou descrição de um medicamento. De forma a referenciar uma linha única, tornando-a distinta das outras, deve-se definir um dos seus atributos como chave primária, funcionando como um índice, sendo este um valor único não repetido.

Para aceder, processar ou adicionar dados a uma base de dados é necessário ter um sistema de gestão de base de dados, como o *MySQL Server.* Estes sistemas são progra-

mas capazes de criar ou modificar bases de dados, através de operações como inserir, editar ou remover registos, definir critérios de visualização dos registos, ou acesso à informação através de outras tecnologias. É possível ainda definir regras que regem as relações entre os vários campos de dados, de forma a obter uma estrutura bem organizada e adaptada às necessidades do utilizador.

2.2.8. Aplicações existentes

Através da plataforma *Google Play* é possível encontrar-se algumas aplicações semelhantes à que se pretende desenvolver. Como tal, de forma a obter um conjunto de opções fundamentais, tornando a aplicação o mais completa possível, estudou-se aplicações semelhantes, permitindo observar pontos positivos e negativos das mesmas, representados nas tabelas 2, 3, 4, 5, 6, 7, 8 e 9. Na tabela 10 é possível observar tanto os pontos fortes, como funcionalidades a evitar em cada aplicação estudada.

Tabela 2 - *Mypillbox* - Aplicação transferida do *GooglePlay* para análise

Aplicação	Mypillbox (versão testada a 11 de outubro de 2013)
Complexidade	Bastante completo, mantem um registo de medicação, no qual permite mencionar se a medicação está a fazer efeito e se foi tomada, como uma opção para introduzir notas do médico
Apresentação	Na seleção de medicamento, é possível atribuir uma cor, a forma do medicamento (Comprimido, injeção, penso...)
Língua	Inglês

Tabela 3 - *Medisafe* - Aplicação transferida do *GooglePlay* para análise

Aplicação	Medisafe (versão testada a 11 de outubro 2013)
Complexidade	Bastantes completo, tornando-se pouco intuitivo. Permite manter um registo para vários utilizadores, criar lembretes para adquirir mais medicamento. Ao introduzir-se o nome do medicamento aparecem sugestões. (aplicação paga, com 30 dias para teste)
Apresentação	Na seleção dum medicamento, é possível atribuir uma cor, assim como a forma do medicamento (Comprimido, injeção, penso...). Apresenta uma ilustração do dia em 4 fatias (madrugada, manha, tarde e noite)

Língua	Português

Tabela 4 - *Pillbox* - Aplicação transferida do *GooglePlay* para análise

Aplicação	Pillbox (versão testada a 12 de outubro 2013)
Complexidade	Muito simples, permite introduzir medicamentos e formas de toma de medicação
Apresentação	Permite escolher o nome do medicamento, a dosagem e comentários
Língua	Inglês

Tabela 5 - *Med Minder* - Aplicação transferida do *GooglePlay* para análise

Aplicação	Med Minder (versão testada a 12 de outubro 2013)
Complexidade	Muito simples, permite apenas introduzir medicamentos e definir o seu alarme
Apresentação	Permite escolher o nome do medicamento, e a dosagem diária e a hora da mesma.
Língua	Inglês

Tabela 6 - Seu Remedio - Aplicação transferida do *GooglePlay* para análise

Aplicação	Seu Remédio (versão testada a 12 de outubro 2013)
Complexidade	Bastantes completo, tornando-se pouco intuitivo. Permite manter um registo para vários utilizadores, criar lembretes para adquirir mais medicamento. Ao introduzir-se o nome do medicamento aparecem sugestões.
Apresentação	Composto por três menus, para introduzir uma nova receita, ordenar as listadas e adicionar anotações. Vários tipos de receita para escolha
Língua	Português

Tabela 7 - *WhatPills* - Aplicação transferida do *GooglePlay* para análise

Aplicação	WhatPills (versão testada a 13 de outubro 2013)
Complexidade	Muito simples de utilizar. É colocado um adesivo NFC (Near Field Communication) no medicamento e, por proximidade do dispositivo Android, é registado na aplicação. Para consultar ou tomar basta aproximar do dispositivo
Apresentação	Apenas 3 opções no menu, bem identificados e intuitivos
Língua	Português

Tabela 8 - *Medical Droid* - Aplicação transferida do *GooglePlay* para análise

Aplicação	Medical Droid (versão testada a 13 de outubro 2013)
Complexidade	Muito simples, contem uma lista com o nome de medicamentos que permite ser descarregada para a aplicação.
Apresentação	Composto por duas opções, para carregar a base de dados de medicamentos, e definir um novo alarme para um medicamento. Permite introduzir o nome da medicação a tomar, a hora que deve ser tomado, e se a medicação é contínua ou temporária. Não emite sinal sonoro, apenas informa como notificação.
Língua	Português

Tabela 9 - *MediLog* - Aplicação transferida do *GooglePlay* para análise

Aplicação	MediLog (versão testada a 14 de outubro 2013)
Complexidade	Simples e intuitivo, permite introduzir o nome do medicamento, definir a altura a que deve ser tomado e definir a hora a que deve notificar.
Apresentação	Mostra um calendário mensal, e premindo um dia permite adicionar um medicamento, no qual se destacam as opções de escolher qual a altura do dia a tomar (pequeno almoço, almoço…). Não emite sinal sonoro, apenas informa como notificação.
Língua	Inglês

Tabela 10 - Pontos fortes e fracos das aplicações analisadas

Aplicação	Opção viável	Evitar
Mypillbox	Regista se a medicação foi ou não tomada	Configuração do medicamento muito complexa
Medisafe	Ao pesquisar um medicamento surgem sugestões de nomes	Pouco intuitivo
Pillbox	Permite editar o nome do medicamento e adicionar comentários	Em inglês
Med Minder	Permite introduzir novos medicamentos	Em inglês
Seu Remédio	Lembretes para adquirir mais medicamento em falta	Introdução de receitas
WhatPills	Intuitivo	Trabalhar através de NFC
Medical Droid	Lista de medicamentos disponível online	Transfere toda a base de dados online
MediLog	Mostra calendário mensal	Não emite sinal sonoro

Após a analise das aplicações existentes no mercado, pretende-se desenvolver uma aplicação intuitiva e que reúna todos os pontos positivos que as aplicações estudadas demonstraram, contornando as suas lacunas para obter uma aplicação solida, funcional e que grupe apenas as principais funcionalidade, tornando-a assim mais simples de usar e com um funcionamento também mais intuitivo.

3. Implementação

No desenvolvimento do projeto é necessário estruturar de uma forma organizada todo o comportamento do sistema. Deste modo optou-se pela utilização de diagramas *UML*, permitindo organizar as ideias que se pretendem projetar, sendo possível visualizar graficamente toda a arquitetura do projeto.
Os diagramas de *UML* são uma das linguagens de modelação mais utilizadas para documentação e especificação de sistemas de *software* [16] uma vez que é uma linguagem predominantemente visual, onde os modelos são construídos por diagramas, tornando mais fácil a compreensão da arquitetura e tornando a sua construção mais rápida. Uma vez que os diagramas ~~de~~ *UML* utilizam princípios de linguagem orientada a objetos, tornam-se particularmente adequados para linguagens deste tipo, como C++, C# e Java.
Para representar a arquitetura da aplicação foram utilizados diagramas UML, de modo a facilitar a sua compreensão. Na figura 4 encontra-se representado o fluxograma de toda a aplicação. Ao executar a aplicação, é possível prosseguir por quatro ramos, nomeadamente, listar os medicamentos armazenados, listar os alarmes criados, e no caso do utilizador selecionar um dos alarmes, o próximo menu apresentado será o de configuração de alarme, adicionar um medicamento e por fim encerrar a aplicação. Selecionada a opção de listar medicamentos, é permitido escolher um dos medicamento apresentados, que levará também para o menu de configuração de alarme, voltar para o menu anterior ou procurar um medicamento *online*, onde será apresentada uma outra lista de medicamentos armazenada num servidor remoto. O menu adicionar um medicamento apresenta as opções de camara, que inicia a aplicação da camara fotográfica do dispositivo, confirmar a criação do medicamento, ou voltar ao menu anterior.

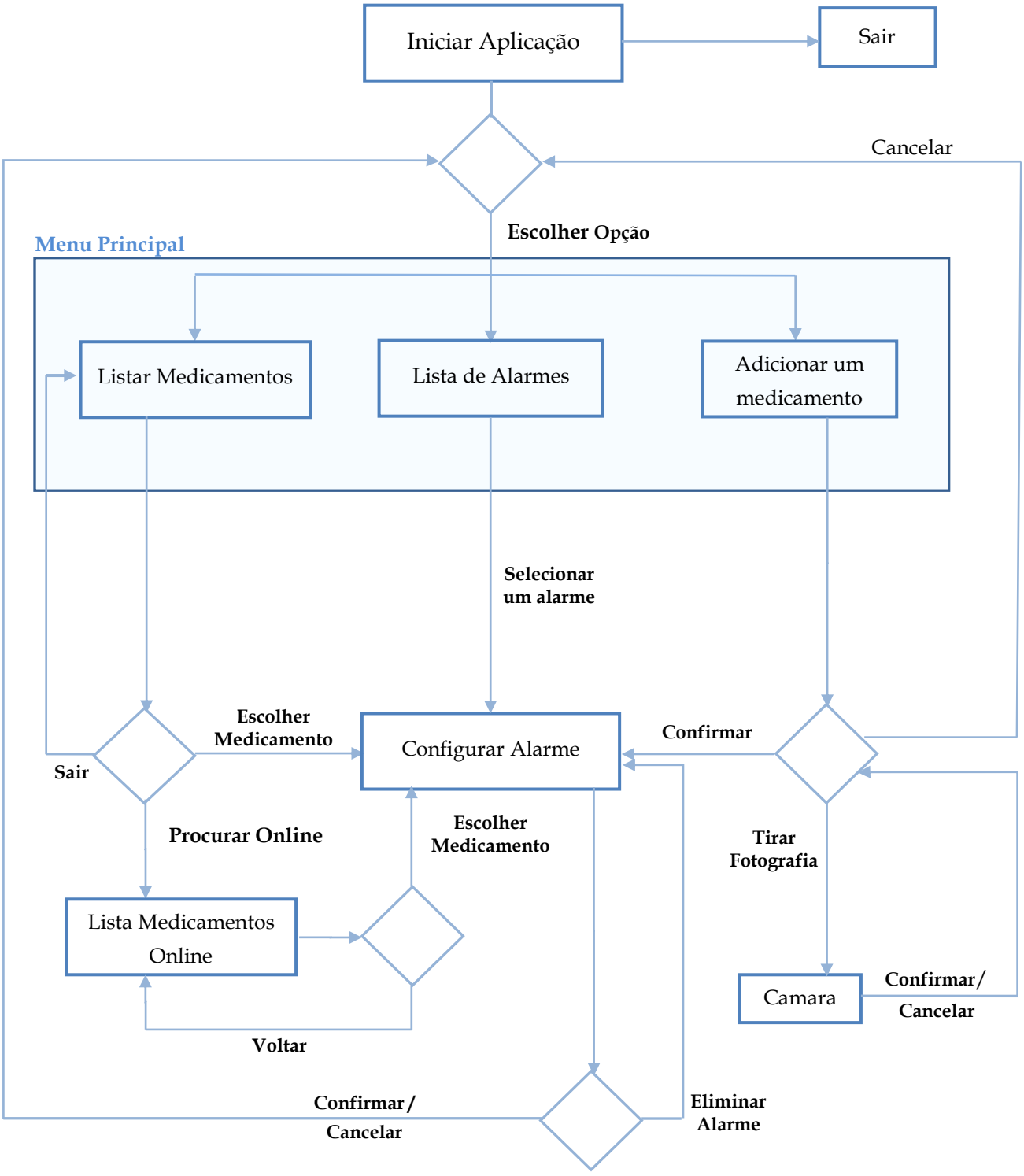


Figura 4 - Fluxograma da estrutura da aplicação

3.1. Requisitos Funcionais

Após uma pesquisa e análise de aplicações semelhantes, as características observadas e retidas para implementação basearam-se nas seguintes funcionalidades:

- Adicionar um medicamento personalizado, com o seu nome e uma breve descrição do mesmo, e adicionar uma sua fotografia, capturada através da camara do dispositivo,

- Aceder a uma base de dados disponibilizada num servidor *online*, que contem uma lista de medicamentos, sendo possíveis guardar na lista pessoal do utilizador,

- Permitir criar vários alarmes para a hora de tomar a medicação, definindo o intervalo de tempo entre cada toma,

- Adicionar um contacto telefónico de outem, para que seja enviada uma mensagem escrita quando um dos alarmes dispara, e não é desativado, com o intuito de outra pessoa poder monitorizar a toma dos medicamentos do paciente.

3.2. Base de dados

Para o registo dos medicamentos foram desenvolvidas duas bases de dados, uma que permite ao utilizador armazenar os medicamentos criados pelo mesmo, ou guardar os medicamentos que são disponibilizados *online*. Para que a aplicação não ocupe demasiada memória nos dispositivos, foi criada uma base de dados *online*, contendo vários medicamentos com as respetivas descrição e imagem, sendo que o utilizador poderá guardar os que pretender na sua lista pessoal.

3.2.1. SQLite database

A aplicação é constituída por duas bases de dados, desenvolvidas a partir do pacote *android.database.sqlite,* incluindo classes que permitem criar e gerir bases de dados individuais [17]. A base de dados com o nome "DatabaseMedicamentos", representada na figura 5, é constituída por uma tabela "medicamentos" que por sua vez contem quatro campos:

- ID, onde é armazenado o número do medicamento, com a função de o identificar na base de dados;
- Nome, onde é armazenado o nome do medicamento;
- Descrição, onde é armazenado a descrição sucinta do medicamento
- Foto, onde é armazenado uma imagem do medicamento, geralmente da caixa deste.

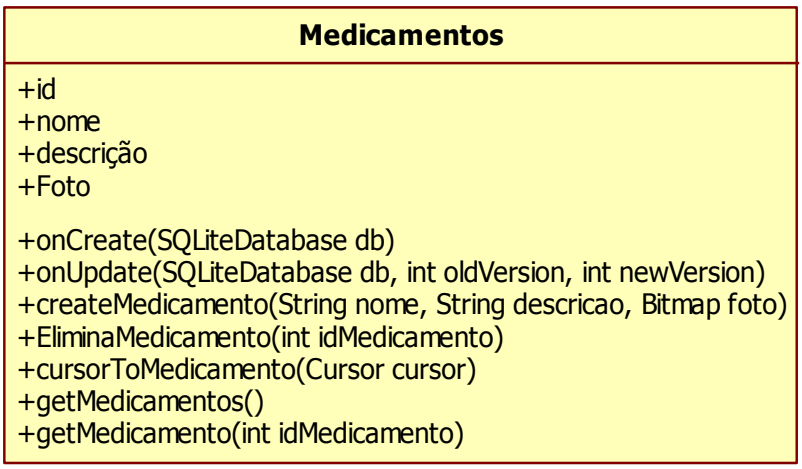


Figura 5 - Base de dados de medicamentos

Paralelamente existe uma outra tabela local, representada na figura 6, encarregue de registar os alarmes criados, constituída pelos seguintes campos:

- Column_Alarm_Id, onde é armazenado o número do alarme, com a função de o identificar na base de dados;
- Column_Alarm_Active, que permite definir o estado do alarme, se ativo ou inativo;
- Column_Alarm_Time, onde é armazenada a hora a que deve disparar o alarme;
- Column_Alarm_Time_Repet, onde é armazenado o intervalo de tempo entre a repetição do alarme;
- Column_Alarm_Days, onde é registado a que dias da semana deve tocar o alarme;
- Column_Alarm_Tone, onde é registado qual o toque que deve soar aquando o alarme dispara;
- Column_Alarm_Vibrate, que permite definir se o alarme vibratório está ativo ou desativo;
- Column_Alarm_Name, onde é armazenado o nome do medicamento;
- Column_Alarm_Foto, onde é armazenada a imagem do medicamento.

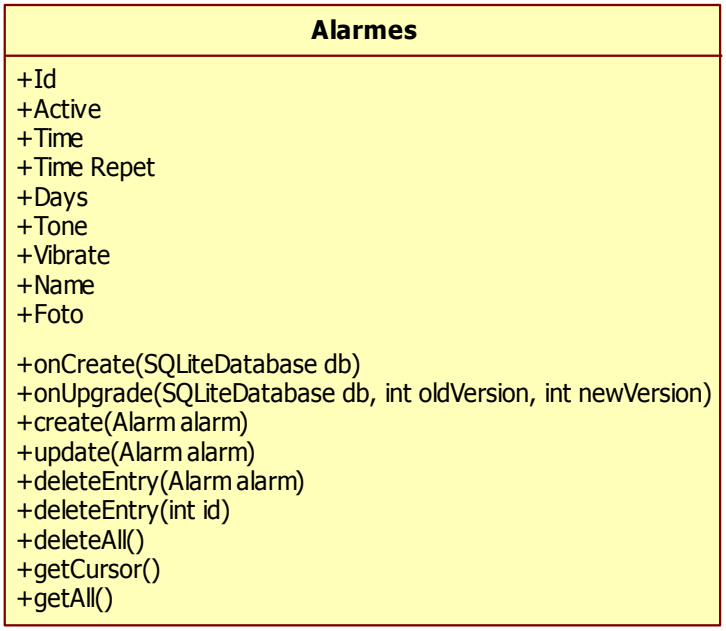


Figura 6 - Base de dados de alarmes

Quando se pretende adicionar um novo medicamento à lista pessoal é necessário introduzir o nome e uma descrição do medicamento em questão, sendo que se o utilizador não pretender adicionar uma imagem do medicamento, será carregada uma imagem padrão com o símbolo de imagem em falta, representada na figura 7. No caso de consultar a lista de medicamentos alojada *online*, as informações serão copiadas automaticamente para a lista pessoal, possibilitando a edição de todos os campos posteriormente.

Figura 7 - Imagem carregada automaticamente caso o medicamento adicionado não possua imagem

A base de dados remota, representada na figura 8, está alocada num servidor contendo uma lista de medicamentos com as respetivas descrições e imagens, sendo possível ao utilizador guardar os que pretender na sua lista pessoal. Esta base de dados tem uma estrutura semelhante à que se encontra alocada na aplicação. Porem, a descrição do medicamento está dividida em várias colunas, tais como: DCI, ou denominação comum internacional, forma farmacêutica, embalagem, titular AIM ou Autorização de Introdução no Mercado e data em que foi introduzido no mercado.

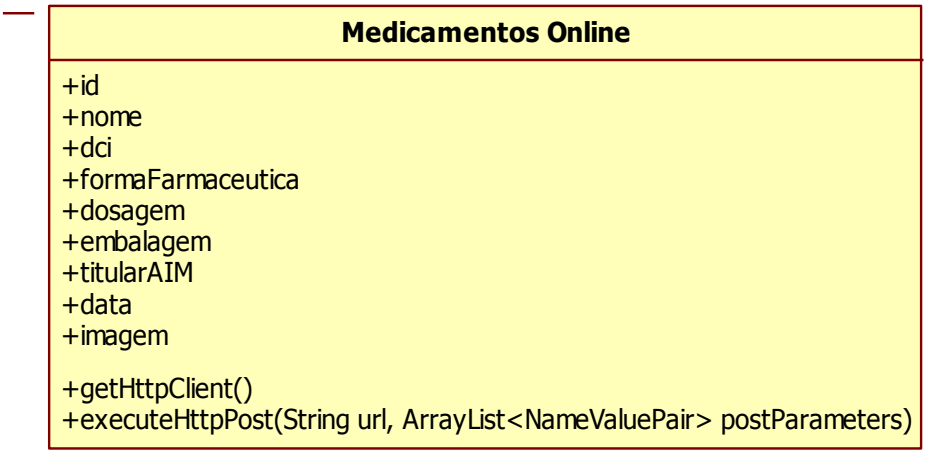


Figura 8 - Base de dados de medicamentos *online*

Para consultar a base de dados remota é necessário efetuar um pedido *http*, utilizando o método *httpPost*. O protocolo *http* é um *application-level protocol*, ou seja um protocolo de comunicação entre sistemas de informação que permite a transferência de dados entre estes sistemas [18].

O método *POST* é utilizado para solicitar ao servidor de origem para que aceite os dados enviados para serem processados por si posteriormente, sendo estes identificados pelo *Request-URI*. Neste caso é enviado no método *POST* um *url* que contem um script em PHP responsável pela seleção dos dados pretendidos na base de dados remota. Caso o método *POST* seja bem sucedido é devolvido um *array* do tipo *JSONArray* [19]. *JSON* é um formato de envio simples de dados entre sistemas baseado em *JavaScript*, em forma de texto independente do idioma utilizado, que utiliza convenções familiares aos programadores de linguagem *C-family*. O *JSONArray* é devolvido numa lista de valores organizados sendo separados por uma vírgula. Através do comando *getJSONObject* é possível obter cada objeto que compões o *JSONArray*.

Na figura 9 é possível observar uma representação gráfica da tabela de medicamentos que se encontra disponível no servidor.

Sort by key: None

id	nome	dci	formaFarmaceutica	dosagem	embalagem	titularAIM	data	imagem
1	A-A-S	Acido acetilsalicilico	Comprimido	500 mg	Blister - 20 unidade(s)	Omega Pharma Portuguesa, Unipessoal, Lda.		[BLOB - 0 B]
2	Acarilbial	Benzoato de benzilo	Solucao cutanea	277 mg/ml	Frasco - 1 unidade(s) - 200 ml	Bial - Portela & Cº, S.A.	17/12/1957	[BLOB - 0 B]
3	Acetilcisteina Azevedos	Acetilcisteina	Comprimido efervescente	600 mg	Recipiente para comprimidos - 20 unidade(s)	Laboratorios Azevedos - Industria Farmaceutica, S....	15/07/2005	[BLOB - 0 B]
4	Acetilcisteina Bluval 600 mg Comprimidos efervesce...	Acetilcisteina	Comprimido efervescente	600 mg	Saqueta - 20 unidade(s)	Sandoz Farmaceutica, Lda.	28/07/2006	[BLOB - 0 B]
5	Acetilcisteina Generis	Acetilcisteina	Comprimido efervescente	600 mg	Recipiente para comprimidos - 20 unidade(s)	Generis Farmaceutica, S.A.	22/03/2007	[BLOB - 0 B]
6	Acetilcisteina Mepha	Acetilcisteina	Po para solucao oral	600 mg	Saqueta - 20 unidade(s)	Mepha - Investigacao, Desenvolvimento e Fabricacao...	26/04/2012	[BLOB - 0 B]
7	Acetilcisteina Pharmakern	Acetilcisteina	Comprimido efervescente	600 mg	Recipiente para comprimidos - 20 unidade(s)	Pharmakern Portugal - Produtos Farmaceuticos, Soci...	30/10/2009	[BLOB - 0 B]
8	Acetilcisteina Sandoz	Acetilcisteina	Comprimido efervescente	600 mg	Recipiente para comprimidos - 20 unidade(s)	Sandoz Farmaceutica, Lda.	28/07/2006	[BLOB - 0 B]
9	Acetilcisteina ToLife	Acetilcisteina	Comprimido efervescente	600 mg	Recipiente para comprimidos - 10 unidade(s)	ToLife - Produtos Farmaceuticos, S.A.	18/12/2008	[BLOB - 0 B]

Figura 9 – Tabela de medicamentos presentes na base de dados remota

Na figura 10 é possível observar o diagrama UML da classe *pill_clock* com todas as classes dependentes, que permitem construir os menus e navegação entre eles, como se poderá verificar mais à frente.

Na figura 11 está representado o diagrama UML de toda a aplicação, dividindo-se no acesso remoto e acesso local. Os objetos circulares representam as ações disponibilizadas na aplicação, que permitem aceder ou modificar as tabelas da base de dados. A tabela de cor azul representa a base de dados remota, que contem uma lista de medicamentos disponibilizada ao utilizador para consultar e guardar no seu dispositivo, e a amarelo é possível observar as tabelas que armazenam os medicamentos do utilizador no dispositivo e os respetivos alarmes configurados pelo mesmo.

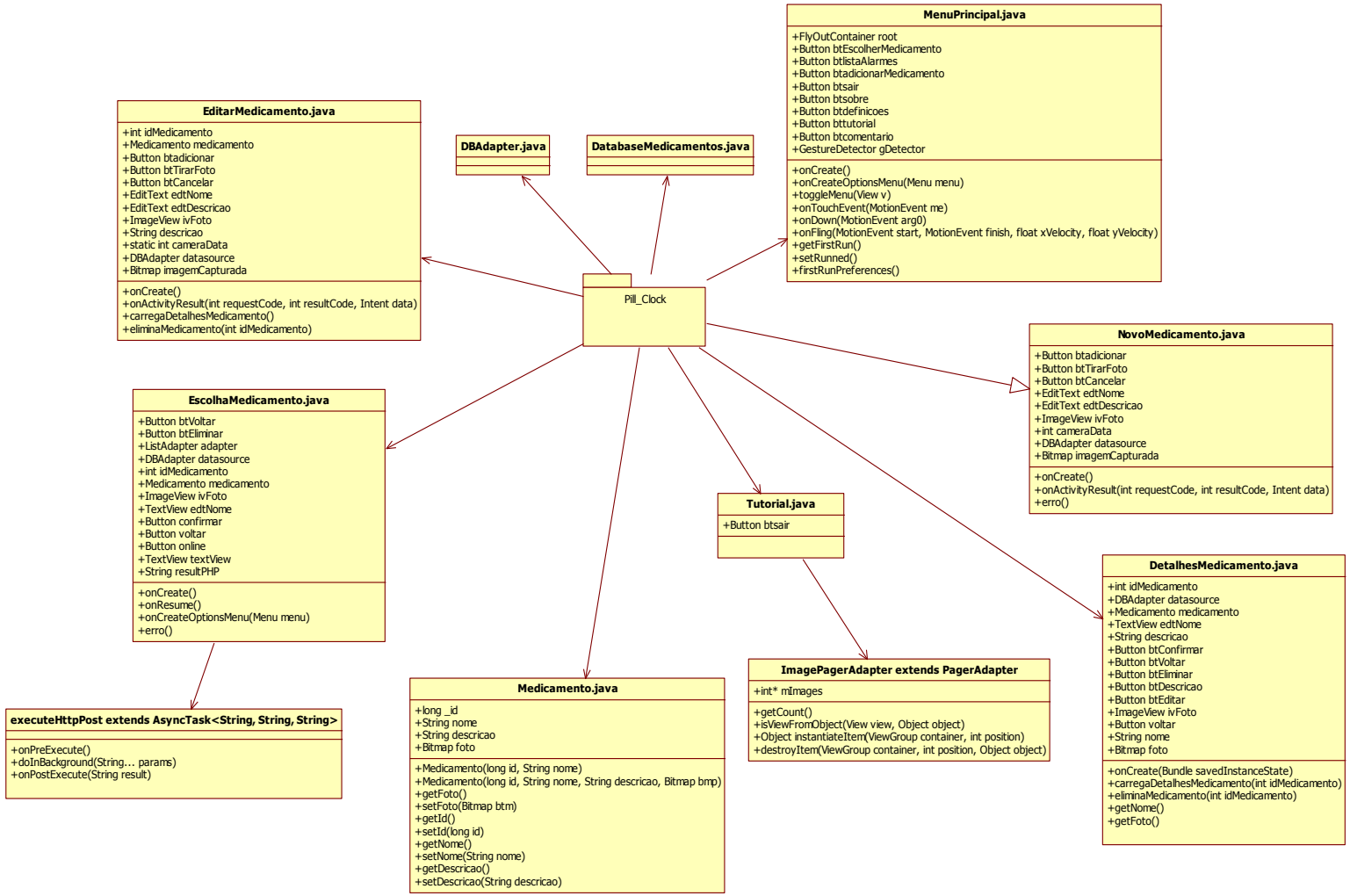


Figura 10 - Diagrama UML da classe Pill_Clock

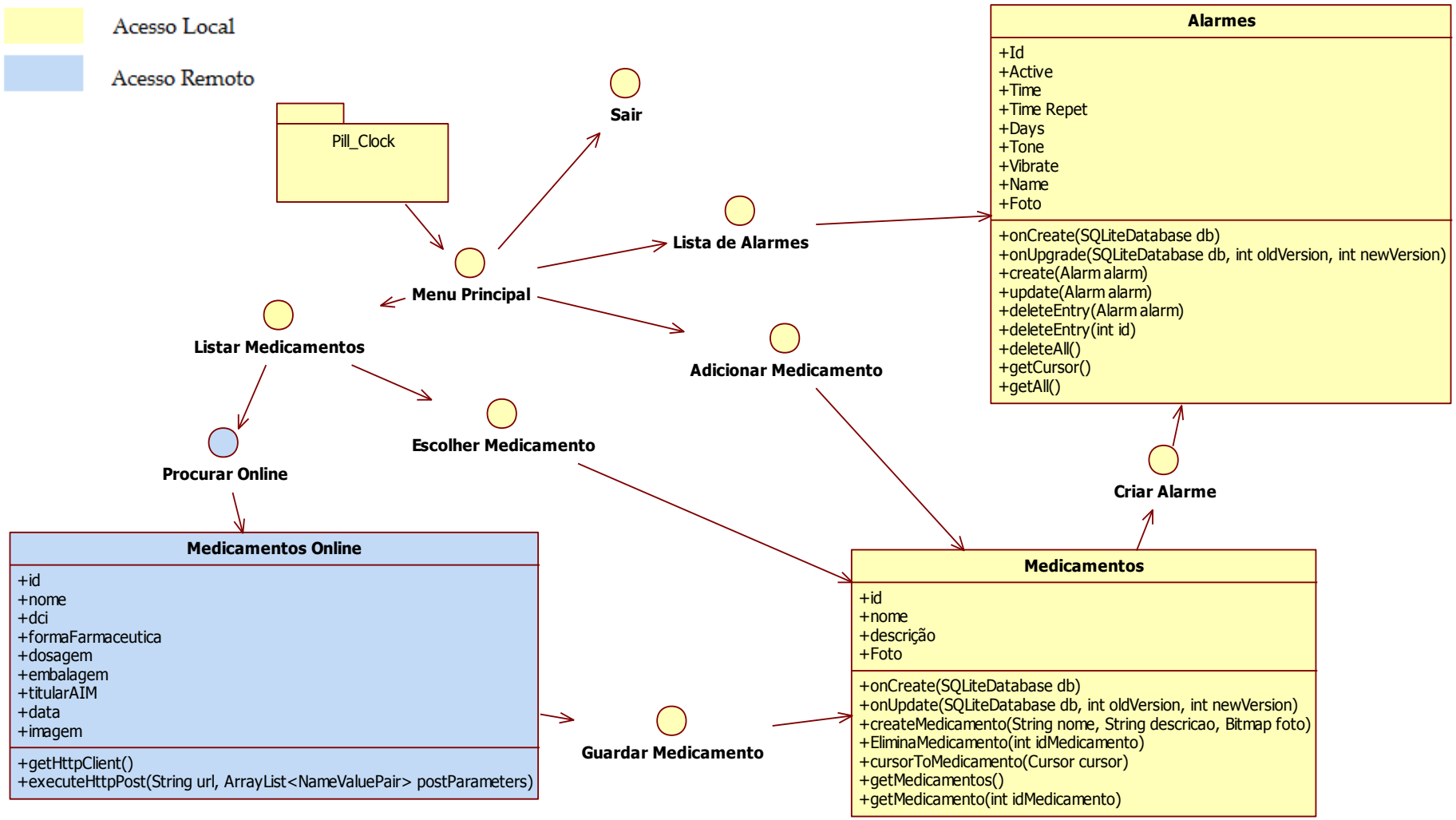


Figura 11 - Diagrama UML da aplicação

3.2.2. PhpMyAdmin

phpMyAdmin [20] é uma ferramenta de software gratuita, escrita em *PHP*, destinada à modelação de *SQL* através da web. Na figura 12 encontra-se representado um exemplo da interface da ferramenta. Operações como gestão de bases de dados, tabelas, colunas, relações, índices, utilizadores ou permissões podem ser realizadas através da interface de utilizador disponibilizada, oferecendo ainda a capacidade de executar diretamente qualquer *query* em *SQL*.

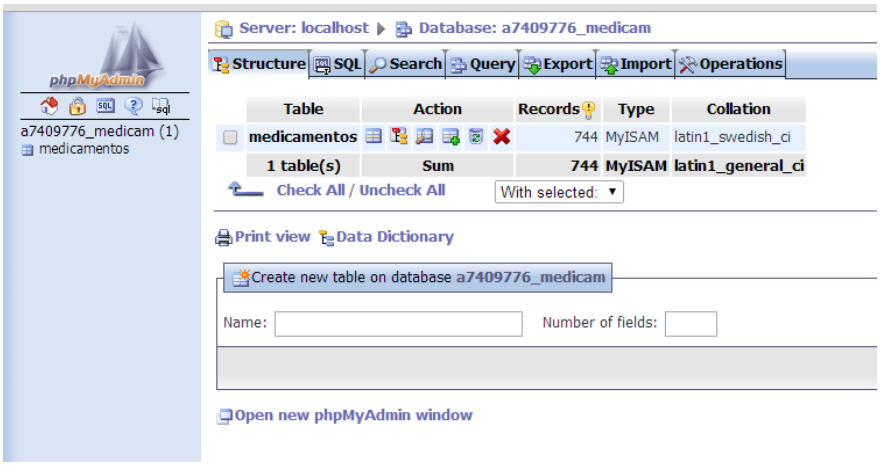


Figura 12 - Interface de utilizador disponibilizada no phpMySQL

3.3. Classes

Uma classe é um *template* utilizado para criar objetos e definir os seus tipos de dados, assim como os seus métodos. Duma forma geral, os objetos são caracterizados por atributos, ou seja as propriedades destes, e métodos, que representam as ações que um objeto pode tomar. Uma classe é um conjunto de objetos que têm propriedades em comum e podem realizar as mesmas ações.

A aplicação desenvolvida é constituída por 10 pacotes de classes distintas, diferenciadas pelas suas funcionalidades e descriminadas seguidamente.

3.3.1. alarm

O pacote *alarm* tem como objetivo gerir os alarmes criados, assim como as suas características, sendo constituído por quatro classes:

- alarm.java – Esta classe é constituída por funções que permitem definir e consultar as opções disponíveis no alarme, como o nome do medicamento, hora do alar-

me, intervalo de repetição ou alarme vibratório. Esta classe é estendida à classe *Serializable*. Em *Android* não é possível passar referências de objetos para as atividades, sendo necessário enviar através de um *Intent*. Esta classe permite representar um objeto como uma sequência de bytes [21], onde estão incluídos os seus dados, assim como a informação sobre o seu tipo. Depois do objeto estar escrito num ficheiro, este pode ser lido a partir do mesmo, e a mesma sequência de bytes permite recrear o objeto em memória, ou seja, modificar a informação do objeto.

- alarmListAdapter.java e ListaAlarmes.java – Estas duas classes funcionam em paralelo, com o objetivo de criar o menu que lista os alarmes criados. A classe alarmListAdapter.java é estendida à classe *ListActivity,* e permite exibir uma lista de *items* [22] que despertam eventos quando o utilizador seleciona um dos mesmos. Paralelamente, a classe ListaAlarmes.java é estendida à classe *BaseAdaper*, que contem um conjunto de métodos [23] que permitem manipular a lista criada, tais como verificar se um item está ativo ou se a lista está vazia.

- ShakeEventListener.java – Contem funções que obtêm dados a partir do giroscópio do dispositivo, para quando um alarme é disparado possa ser desativado agitando o dispositivo. Esta classe é estendida à classe *SensorEventListener*, que permite receber valores dos sensores quando estes são alterados [24].

3.3.2. alarm.alert

O pacote alarm.alert tem como função executar o disparo de um alarme, gerindo todas as necessidades de gestão desencadeadas pelo evento, constituído pelas seguintes classes:

- AlarmAlertBroadcastReciever – tem como função verificar quando um alarme é disparado. Esta classe é estendida à classe *BroadcastReceiver* [25], que permite iniciar o alarme, dando foque principal ao alarme, seja qual for o estado atual do dispositivo.

- AlarmAlertActivity – esta classe permite obter as definições do alarme que irá ser executado, de modo a preparar a sua execução. Esta atividade é iniciada pela classe AlarmAlertBroadcastReciever.

3.3.3. alarm.database

O pacote alarm.database contém a base de dados que permite armazenar e gerir os alarmes criados, constituído pela classe DataBase. Esta classe é estendida à classe *SQLiteOpenHelper* [26], que permite implementar a subclasse *onCreate e onUpdate*, que por sua vez possibilitam a criação e atualização da base de dados.

3.3.4. alarm.preferences

O pacote alarm.preferences tem com o objetivo gerir as características dos alarmes criados, como a hora de disparo, intervalos entre disparos, tom de alarme, entre outros.

- Opcoes.java – Esta classe contém as estruturas necessárias para armazenar as características dos alarmes, assim como os métodos de *get* e *set* correspondentes.

- OpcoesAlarme.java e OpcoesAlarmeListAdapter.java – Estas duas classes funcionam em paralelo, com o objetivo de criar o menu que lista as opções dos alarmes. A classe OpcoesAlarme.java é estendida à classe *ListActivity,* e permite exibir uma lista com todas as opções configuráveis de um alarme. Paralelamente, a classe OpcoesAlarmeListAdapter.java é estendida à classe *BaseAdaper*, permitindo alterar as opções do alarme.

3.3.5. alarm.service

O pacote alarm.service, tem como função despertar um alarme no momento para o qual se encontra programado. Este pacote é constituído por uma classe com o mesmo nome, que é estendida à classe *Service* [27], permitindo fazer a gestão dos alarmes. Um serviço é um componente da aplicação que permite executar operações em *background* sem exibir uma interface. A classe *Service* permite uma execução *multi-threading*. Ao invés de processar solicitações por ordem, executa vários pedidos ao mesmo tempo, utilizando uma *thread* para executar e processar apenas um pedido de cada vez. Esta implementação permite criar vários alarmes, possibilitando que cada um seja executado de forma independente.

3.3.6. alarm.telephony

O pacote alarm.telephony, tem como função verificar quando uma chamada é recebida no dispositivo. Este pacote é constituído por uma classe com o mesmo nome, estendida à classe *BroadcastReceiver,* que permite receber a ação *PHONE_STATE.* Esta ação indica quando o estado do dispositivo muda, ou seja, quando o dispositivo recebe ou termina uma chamada esta ação é despertada [28].

3.3.7. online

O pacote online contém classes encarregues pela consulta da base de dados que se encontra alocada num servidor, constituído pelas seguintes classes:

- CustomHttpClient.java – Esta classe contém os métodos necessários para realizar um pedido *http*, utilizando o pacote *org.apache.http*, como a representação da mensagem ou definir os *timeout* [29].

- MedicamentosOnline.java – Esta classe, estendida à classe *ListActivity,* lista todos os medicamentos encontrados na base de dados externa. Após o pedido *http* ser bem sucedido, é retornado uma *string* do tipo *JSONarray* [30]. Este tipo de *string* é constituído por uma sequência de *JSONObjects* [31], tendo ainda a capacidade de retornar outros *JSONarray*, *strings, booleans, integers, longs* e *doubles.* Um *JSONObjects* é constituído por uma *string*, designada por "nome", seguida de um "valor" que poderá ser do tipo *JSONarray*, *strings, booleans, integers, longs* ou *doubles.* Neste caso concreto, cada *JSONObjects* retornado é constituído pelo id do medicamento, seguido do seu nome, DCI, ou denominação comum internacional, forma farmacêutica, embalagem, titular AIM ou Autorização de Introdução no Mercado e data que foi introduzido no mercado, e por fim a imagem do medicamento. A imagem é devolvida com uma codificação *base64* [32].

- DetalhesMedicamentoOnline.java – Esta classe, estendida à classe *Activity*, mostra os detalhes do medicamento selecionado, permitindo voltar à seleção anterior ou guardar o medicamento na base de dados do dispositivo.

3.3.8. pills_clock

O pacote pills_clock, contém as classes que permitem a navegação entre menus. Neste mesmo pacote existe numa nova base de dados, semelhante ao pacote alarm.database, onde serão armazenados os medicamento criados pelo utilizador, ou retirados da base de dados externa.

- MenuPrincipal.java - Esta classe, estendida à classe *Activity*, apresenta as opções iniciais disponíveis, nomeadamente, escolher um medicamento, listar os alarmes configurados, adicionar um novo medicamento ou sair da aplicação. Inclui ainda a classe *OnGestureListener* [33] que permite identificar quando ocorre um tipo de gesto. Neste caso pretende-se visualizar ou ocultar o menu que se encontra escondido.

- EscolhaMedicamento.java – Esta classe, estendida à classe *ListActivity,* tem com objetivo listar os medicamentos criados pelo utilizador ou retirados da base de dados externa. Deste menu apresentado é possível passar para outros dois menus: DetalhesMedicamento.java, que apresenta os detalhes do medicamento selecionado, como o seu nome, a imagem e a descrição ou para o menu apresentado pela classe MedicamentosOnline.java, descrito anteriormente.

- EditarMedicamento.java - Esta classe, estendida à classe *Activity*, permite modificar todos os detalhes de um medicamento, como a sua descrição ou imagem. É possível aceder a esta opção através do menu que apresenta os detalhes de um medicamento.

- NovoMedicamento.java - Esta classe, estendida à classe *Activity*, permite adicionar um novo medicamento à base de dados, adicionando o seu nome, uma descrição e adicionar uma fotografia através da câmera do dispositivo.

- NovoMedicamento.java - Esta classe, estendida à classe *Activity*, apresenta um tutorial de como utilizar a aplicação, explicando todos os menus e funcionalidades disponíveis, através de uma galeria de imagens.

- DatabaseMedicamentos.java - estendida à classe *SQLiteOpenHelper,* e à semelhança da classe alarm.database.java, permite armazenar os medicamentos criados pelo utilizador ou retirados da base de dados externa. Em simultâneo utiliza a classe DBAdapter.java, que contem um conjunto de métodos facilitando assim a sua manipulação, como criar, eliminar ou aceder a um medicamento.

3.3.9. perfil.database

O pacote perfil.database tem como função permitir adicionar os dados do utilizador da aplicação, o seu nome, um número de telefone de outrem, com o objetivo informar uma outra pessoa, através de uma mensagem escrita, cada vez que um alarme do utilizador é disparado, e uma opção para ativar ou desativar a funcionalidade.

3.3.10. view.viewgroup

O pacote view.viewgroup, tem como função criar a animação de *slidshow* permitindo visualizar o menu oculto presente no menu principal. Este pacote é constituído pela classe *FlyOutContainer*, estendida à classe *LinearLayout* [34], que permite organizar o conteúdo existente numa única coluna ou linha. Esta classe ainda inclui o pacote *android.view.animator* que fornece mecanismos para criar animações, tais como alteração do conteúdo presente, como as suas dimensões, rotação ou posição.

3.4. Interface do utilizador

A interface de uma aplicação define-se por tudo o que o utilizador pode ver e interagir. Numa aplicação *android* uma interface é construída a partir da classe *View* [35], um objeto que se caracteriza por uma área retangular de dimensão igual ao display do dispositivo, responsável pela manipulação de eventos através de botões, caixas de texto e outros componentes. A classe *ViewGroup* é outro objeto que permite formar um conjunto de objetos do tipo *View,* formando assim o *layout* da interface. A hierarquia entre *Views* e *ViewGroup* pode ser representada pelo diagrama da figura 13.

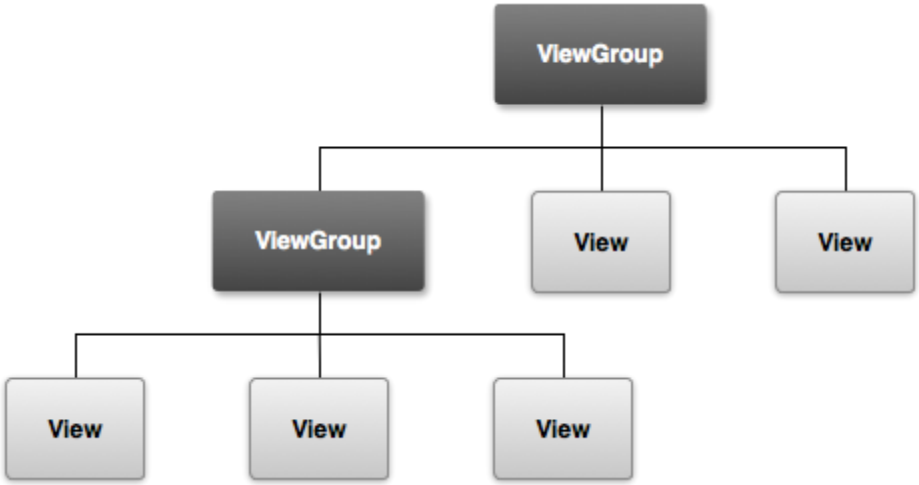


Figura 13 - Representação da hierarquia que define um *Layout*

O modo mais simples para implementação de um *layout* é a partir de um ficheiro XML, com uma estrutura de fácil interpretação, semelhante ao *HTML*. No ficheiro XML é possível configurar vários parâmetros, tais como os atributos (botões, caixas de texto…) ou os parâmetros de *layout*, ou seja, os tamanhos de cada *View* e *ViewGroup,* assim como as suas posições relativas ao *ViewGroup* hierárquico, distância de margens e posicionamento na *View*.

O diagrama da figura 14 pretende representar a interface geral da aplicação, seus menus, submenus e a interligação entre os mesmos.

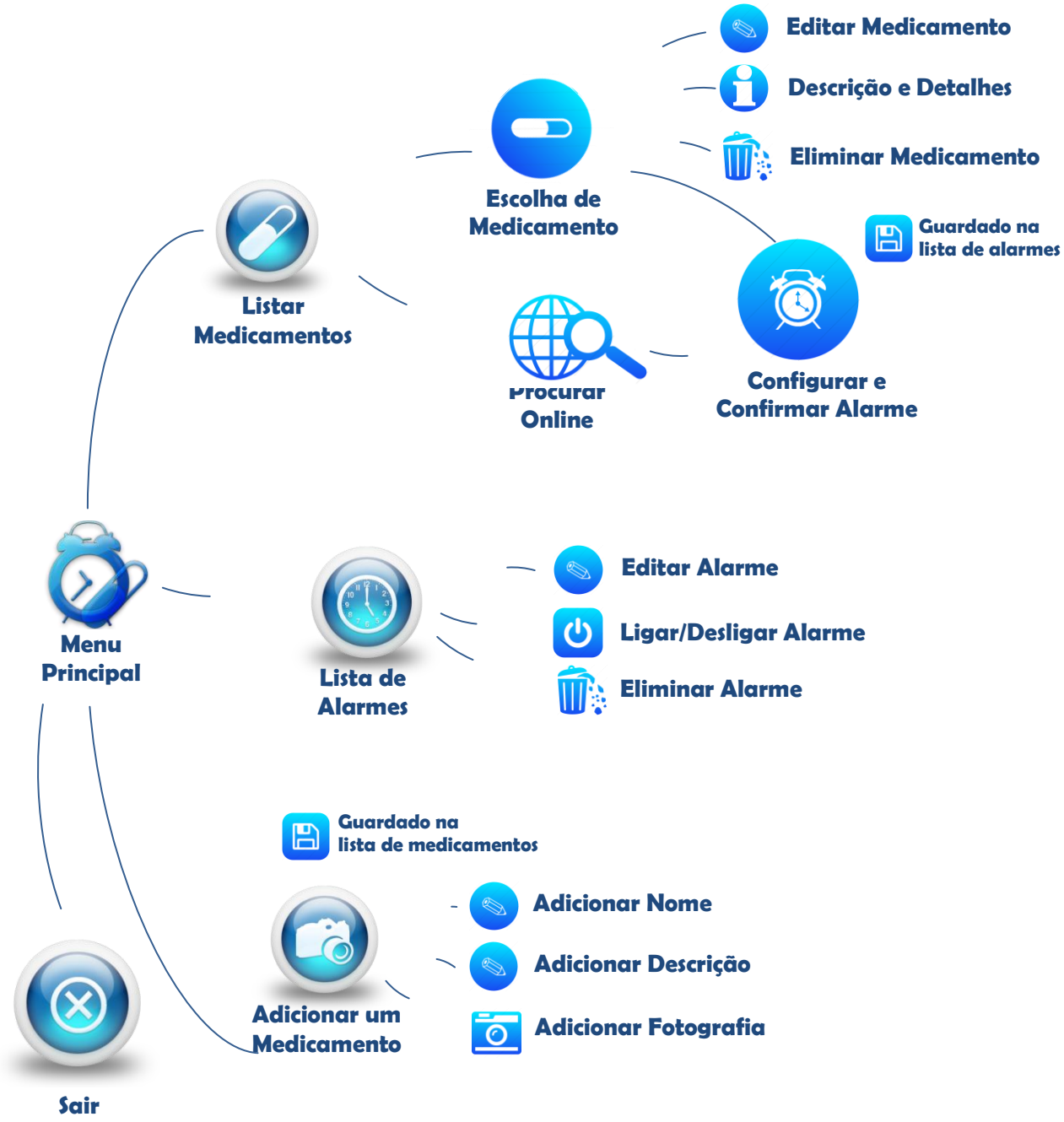


Figura 14 – Diagrama representativo das interfaces da aplicação

3.3.1. Menu Principal

A figura 15 ilustra o menu apresentado quando que se inicia a aplicação, no qual existem 4 ícones, que permitem seguir para 3 submenus:

- Escolha de medicamento;

- Consultar lista de alarmes

- Adicionar um medicamento;

- Sair da aplicação.

(a) (b)

Figura 15 – (a) Representação do menu oculto na página inicial da aplicação, (b) Representação o menu inicial

No menu inicial ainda é possível aceder a um menu oculto deslisando o dedo da esquerda para a direita ao longo do ecrã, onde surgem as opções de "Definições", "Tutorial" e "Sobre".

Na figura 16 é possível observar a opção "Definições", que permite introduzir o nome do utilizador que está a usufruir da aplicação, e adicionar um número de telefone de outra pessoa. Esta função tem como objetivo informar uma outra pessoa, através de uma mensagem escrita, cada vez que um alarme do utilizador é disparado. Caso o utilizador seja uma pessoa sénior, que necessite de acompanhamento por parte de

outrem, esta função permite acompanhar diariamente a toma da medicação do utilizador, podendo entrar em contacto com este, confirmando se a toma foi feita.

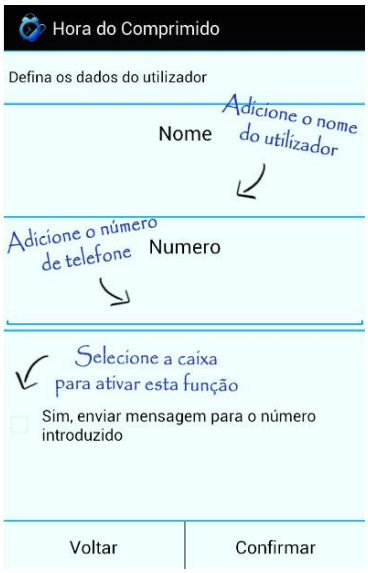


Figura 16 - Representação do menu Definições

A opção "Tutorial" permite visualizar uma sequência de imagens que tem como objetivo exemplificar a utilização geral da aplicação. O tutorial é exibido quando a aplicação é instalada pela primeira vez num dispositivo, permitindo rever caso o utilizador tenha alguma dúvida.

A opção "Sobre" permite visualizar uma descrição geral da aplicação, quais os seus objetivos e qual a versão instalada no dispositivo móvel.

3.3.2. Gestor de medicamentos

Para adicionar medicamentos à lista pessoal do utilizador existem duas formas, através de uma pesquisa *online* na base de dados fornecida, ou adicionando manualmente um novo medicamento, como exemplificado na figura 17. Para aceder-se à base de dados, basta ir ao menu de "Escolha de medicamento", e clicar em "Procura Online". Para visualizar a lista fornecida é necessário garantir que se está conectado à internet, através de *wireless* ou dados móveis.

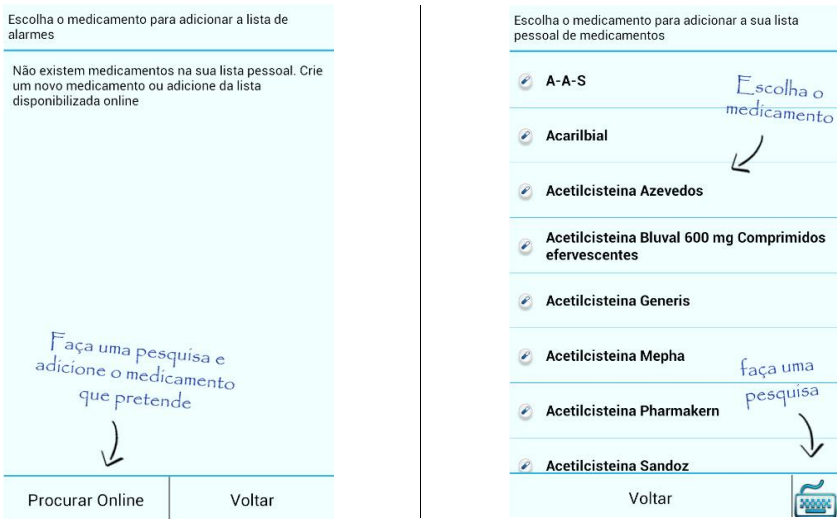


Figura 17 – (a) Representação do menu "Escolha de medicamentos", (b) Representação da lista de medicamentos disponível *online*

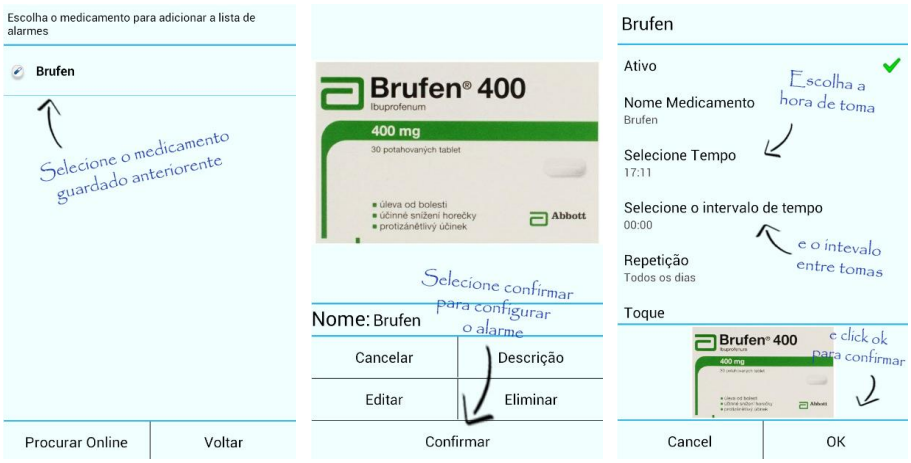
Caso o utilizador não encontre o seu medicamento, pode adicionar manualmente, como se encontra exemplificado na figura 18. Para isso, através da opção "Adicionar Medicamento" no menu inicial, o utilizador pode escrever o nome do medicamento, adicionar uma descrição do mesmo e caso pretenda, utilizando a camara do seu dispositivo móvel, pode tirar uma fotografia ao referido medicamento. Quando concluir o processo, o utilizador prime o botão confirmar e o seu novo medicamento passará a estar disponível na sua lista pessoal, no menu "Escolha de Medicamento".

Figura 18 - Representação do menu "Adicionar medicamento"

3.3.3. Gestor de alarmes

Conforme referido anteriormente, esta aplicação pretende ser um gestor de alarmes pessoal para medicamentos, permitindo ao utilizador adicionar o seu próprio medicamento, ou consultar uma lista de medicamentos fornecida a fim de criar o seu alarme.

Para criar um novo alarme é necessário aceder ao menu "Escolha de Medicamento", e selecionar o medicamento que se pretende programar, como se encontra exemplificado na figura 19.



(a) **(b)** **(c)**

Figura 19 - (a) - Representação do menu "Adicionar medicamento"
(b) - Representação do menu descritivo do medicamento selecionado
(c) - Representação do menu de configuração do alarme

Nas definições de alarme é permitido selecionar a hora a que se pretende tomar o medicamento e o intervalo de tempo entre tomas. Por exemplo, caso se inicie a medicação as 8 horas e o medicamento ter que ser tomado de 6 em 6 horas, deve-se preencher o campo "Selecione Tempo" com "8:00" e o campo "Selecione o intervalo de tempo" com "6:00". Ainda é possível personalizar quais os dias da semana em que se tem de fazer essa medicação, o toque do alarme e ativar ou desativar o alarme vibratório.

Na figura 20 está representado o menu que surge quando um alarme é disparado. O utilizador poderá desativar o alarme, premindo o botão "desligar" ou agitando o dispositivo, caso tome de imediato o medicamento, ou poderá colocar o alarme em modo "snooze" apenas uma vez, caso não possa tomar a medicação de imediato, sendo avisado novamente minutos depois.

Figura 20 – Representação do menu apresentado no disparo de um alarme

Para editar um alarme posteriormente a ser criado, acede-se ao menu "Consultar lista de alarmes", e premindo no alarme que se pretende modificar, todas as definições visualizadas na sua criação ficam disponíveis novamente. Para eliminar um alarme da lista pessoal, basta premir continuamente em cima do alarme pretendido, como se encontra exemplificado na figura 21.

(a) **(b)**

Figura 21 – (a) – Representação da lista de alarmes
(b) – Exemplo da eliminação de um alarme

4. Resultados e Conclusões

A aplicação "Hora do comprimido" foi lançada pela primeira vez no *Google Play* em abril de 2014, sendo esta uma versão protótipo funcional, com o intuito de registar a opinião geral dos utilizadores e possíveis falhas ocorridas.

4.1. Resultados gráficos

Na figura 22 é possível observar-se a representação gráfica da totalidade de instalações efetuadas em dispositivos.

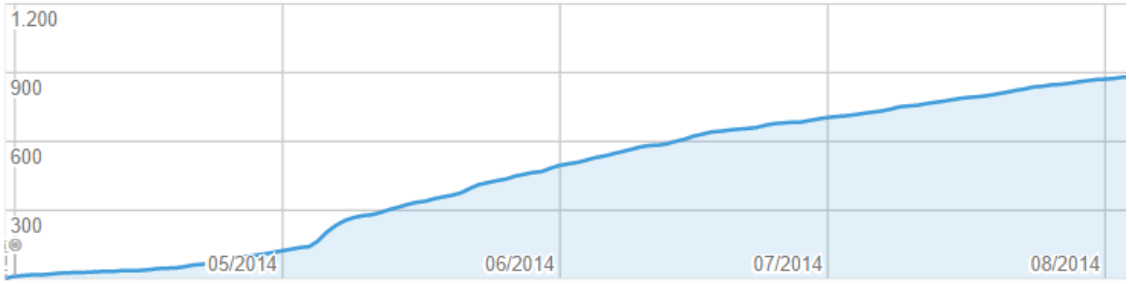


Figura 22 - Total de instalações entre abril de 2014 e agosto de 2014

Através da análise gráfica, verifica-se que existe interesse no mercado por este género de aplicação, uma vez que não existiu divulgação da mesma e contudo foi registado um número consideravelmente alto de instalações.

Uma das métricas importantes para analisar o sucesso de uma aplicação no mercado, é o número atual de instalações ativas, ou seja, o número de utilizadores que têm a aplicação instalada no seu dispositivo no presente momento. Na figura 23 é possível observar o total de aplicações instaladas no fim do mês de agosto.

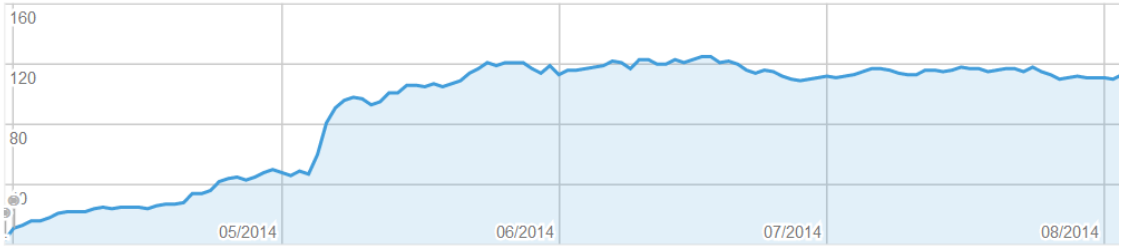


Figura 23 - Instalações atuais por dispositivo

Como se pode observar, o número de instalações atuais é bastante inferior ao valor total registado de instalações ao longo de 5 meses, com um número de aplicações instaladas atualmente de 118 contra 1012 instalações na totalidade. É possível concluir que o número de desinstalações é elevado, especulando que a versão protótipo disponibilizada pode apresentar alguma instabilidade em certos dispositivos. Na figura 24 é possível verificar em detalhe a versão de *Android* em que a aplicação se encontra instalada.

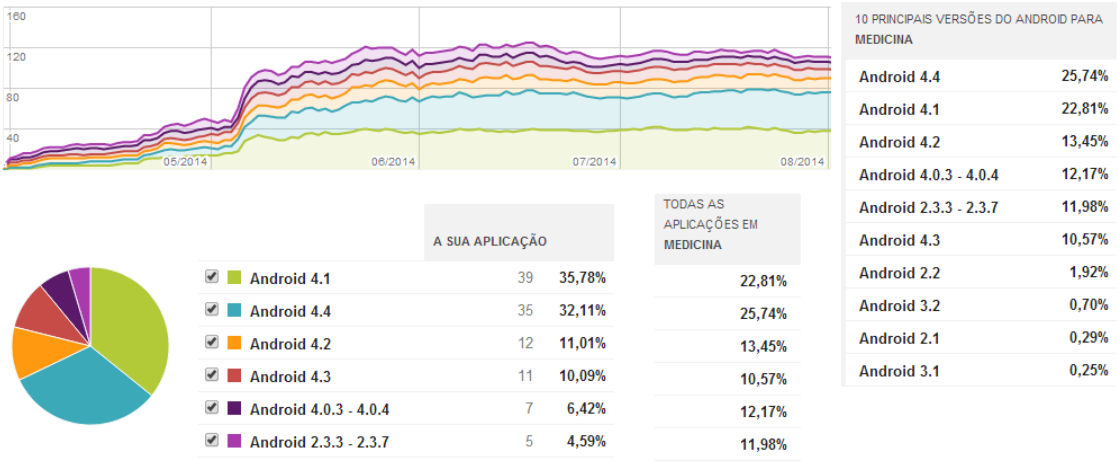


Figura 24 - Instalações atuais por dispositivo por versão do Android

4.2. Análise de Resultados

De modo a que se possa melhorar a aplicação desenvolvida de uma forma continua, é necessário existir um *feedback* por parte dos utilizadores. O *Google Play* oferece duas funcionalidades de *feedback* essenciais, a possibilidade de classificar e comentar uma aplicação, permitindo tanto aos utilizadores como aos criadores saber a opinião geral de uma aplicação, como a funcionalidade de falhas e ANRs (*Application Not Responding*), que reporta erros ocorridos num dispositivo.

O número de comentários registados, ao longo da fase de teste da aplicação, foi baixo, ao contrário do que se desejava. O registo de um comentário e classificação sobre uma

aplicação, requer que o utilizador abra a página da mesma no *Google play*, o que raramente ocorre, pois o utilizador após a instalação de uma aplicação no seu dispositivo, realiza uma primeira análise a esta, decidindo se irá usufruir da mesma ou desinstalar por não servir às suas necessidades, sem voltar à página deste para registar a sua opinião. Na figura 25 é possível observar alguns comentários positivos e negativos mencionados pelos utilizadores.

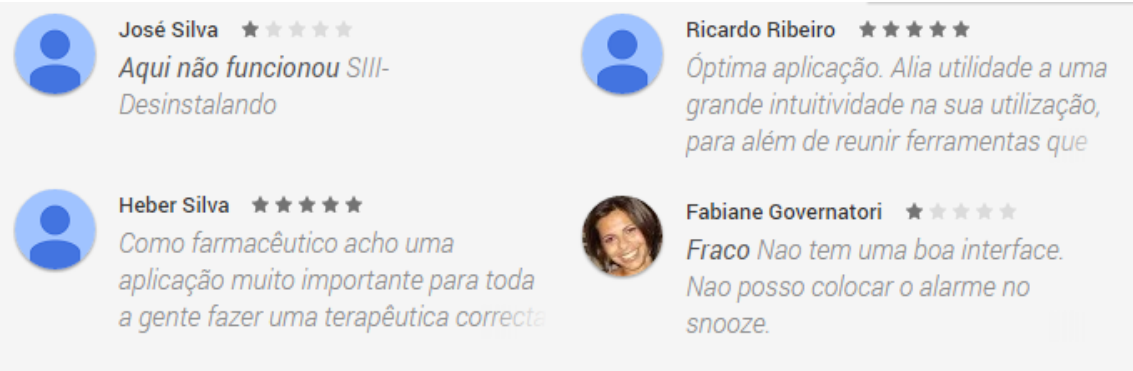


Figura 25 - Comentários registados no Google Play à aplicação desenvolvida

Nas falhas da aplicação reportadas foi possível encontrar um erro de implementação, registado 24 vezes como é possível verificar na figura 26, e um relatório mais detalhado na figura 27. Quando o utilizador efetua uma pesquisa *online* de medicamentos, é necessário previamente realizar um pedido *http*, e quando este ocorre com sucesso, é guardada a lista de medicamento devolvida, para posteriormente listar no ecrã. Durante a fase de testes, não foi detetado qualquer erro devolvido nos pedidos executados. No entanto, o relatório de erros de alguns dispositivos demonstrou que, por vezes, esta lista é devolvida vazia, e como o pedido *http* foi efetuado com sucesso, mesmo que durante a sua execução tenha ocorrido um erro na construção do vetor a devolver. Ou seja, na tentativa de listar os medicamentos obtidos, ocorre uma falha por não existir nenhum item presente, terminando assim a aplicação inesperadamente. Para solucionar este problema bastou executar uma verificação da lista de medicamentos devolvida, e caso esta se encontre vazia, é exibida uma mensagem de erro no ecrã a solicitar que o utilizador volte a tentar executar a pesquisa *online*.

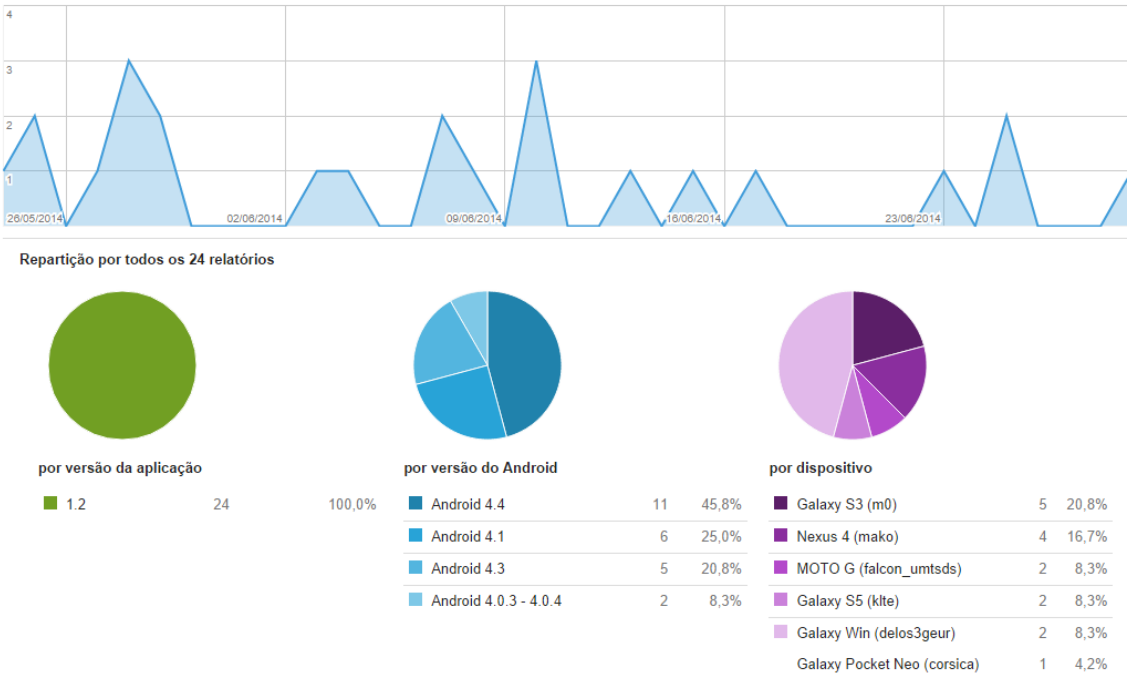


Figura 26 - Registo de falha java.lang.NullPointerException entre 24/05/2014 e 29/06/2014

Última ocorrência
13/06 21:16

Ocorridos esta semana
0

N.º de ocorrências
5

Versão da aplicação
1.2 5

Versão do Android
Android 4.3 5

Dispositivo
Galaxy S3 (m0) 5

```
java.lang.NullPointerException: storage == null
    at java.util.Arrays$ArrayList.<init>(Arrays.java:38)
    at java.util.Arrays.asList(Arrays.java:154)
    at android.widget.ArrayAdapter.<init>(ArrayAdapter.java:141)
    at online.MedicamentosOnline.setList(MedicamentosOnline.java:141)
    at online.MedicamentosOnline$load.onPostExecute(MedicamentosOnline.java:112)
    at online.MedicamentosOnline$load.onPostExecute(MedicamentosOnline.java:1)
    at android.os.AsyncTask.finish(AsyncTask.java:631)
    at android.os.AsyncTask.access$600(AsyncTask.java:177)
    at android.os.AsyncTask$InternalHandler.handleMessage(AsyncTask.java:644)
    at android.os.Handler.dispatchMessage(Handler.java:99)
    at android.os.Looper.loop(Looper.java:176)
    at android.app.ActivityThread.main(ActivityThread.java:5419)
    at java.lang.reflect.Method.invokeNative(Native Method)
    at java.lang.reflect.Method.invoke(Method.java:525)
    at com.android.internal.os.ZygoteInit$MethodAndArgsCaller.run(ZygoteInit.java:1046)
    at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:862)
    at dalvik.system.NativeStart.main(Native Method)
```

Figura 27 - Relatório de erro de falha *java.lang.NullPointerException*

4.3. Conclusões e perspetivas

A aplicação desenvolvida permite aos pacientes sobre tratamento manterem a correta toma da sua medicação. Apesar de não ter sido testada em larga escala, a "Hora do medicamento" demostrou ser uma aplicação intuitiva, de fácil utilização para o paciente, vindo de encontro aos principais objetivos. Esta foi desenvolvida para o sistema operativo *Google Androi*d, o qual mostrou beneficiar de uma *framework* com recursos bem modelados, de fácil acesso, tornando mais simples a implementação das tarefas mais comuns. A linguagem de programação base, *Java* é uma mais-valia na medida em que a programação orientada a objetos facilita, tanto em tempo de desenvolvimento, como de implementação, agregando a extensa biblioteca disponibilizada, que inclui funções genéricas características da linguagem.

4.4. Trabalho Futuro

Pretende-se dar continuidade ao desenvolvimento da aplicação, tornando-a cada vez mais confiável e completa no mundo das aplicações *mobile*. Numa primeira fase de testes, o objetivo é observar a evolução da popularidade da aplicação no *Google Play*, retendo o *feedback* dado pelos utilizadores, a fim de reparar erros que possam existir e surjam na sua utilização, assim como a implementação de sugestões positivas. Em continuidade, a expansão da base de dados de medicamentos alojada em servidor é fulcral, fornecendo aos utilizadores uma maior diversidade de medicamentos, e apostar na inserção de medicamentos mais utilizados no cotidiano.

Num futuro mais próximo pretende-se disponibilizar a função de *snoozee*, possibilitando desativar um alarme por alguns minutos quando este dispara, permitindo que este toque novamente. Esta funcionalidade permite que, caso o utilizador não possa realizar a toma naquele exato momento, seja relembrado minutos mais tarde. A carência desta funcionalidade foi comentada por um utilizador que defende ser um contra da aplicação.

Pretende-se também alterar o tutorial disponível, pois como este está implementado numa galeria de imagens, disponibilizando uma sequência de representações dos menus que existem, pode confundir os utilizadores, levando estes a pensar que estão no exato momento no respetivo menu, sem se aperceberem que estão apenas a visualizar uma imagem representativa desse menu.

5. Referências Bibliográficas

[1] - Richard Miech, Ph.D., M.P.H., Amy Bohnert, Ph.D., Kennon Heard, M.D., Jason Boardman, Ph.D., "*Increasing Use of Nonmedical Analgesics Among Younger Cohorts in the United States: A Birth Cohort Effect*", Journal of Adolescent Health Vol. 52, Issue 1, Pages 35-41, 2013.

[2] – Antti Oulasvirta, Tye Rattenbury, Lingyi Ma, Eeva Raita, "*Habits make smartphone use more pervasive*", Springer-Verlag London Limited, 2011.

[3] – Orrin I. Franko, Timothy F. Tirrell, "*Smartphone App Use Among Medical Providers in ACGME Training Programs*", Springer Science + Business Media, LLC, 2011.

[4] –"*Accounting for Medication Particularities Designing for Everyday Medication Management*", Lea Gulstav Dalgaard, Erik Gronvall, Nervo Verdezoto, Proceedings of the 7th International Conference on Pervasive Computing Technologies for Healthcare, 2013.

[5] – "*Assistive Smartphone for People with Special Needs : the Personal Social Assistant*", S. Verstockt, D. Decoo, D. Van Nieuwenhuyse, F. De Pauw and R. Van de Walle, Human System Interactions, 2009.

[6] – "*Studying Mobile Phone Use in Context: Cultural, Political, and Economic Dimensions of Mobile Phone Use*", Carolyn Wei, Beth E. Kolko, Professional Communication Conference, 2005.

[7] - Dazeinfo.com is a Technology analysis blog, http://www.dazeinfo.com/2014/05/07/microsoft-corporation-msft-windows-phone-

os-doubled-shipment-q1-2014-controls-3-global-mobile-phone-market/, acesso a 24 de novembro de 2014.

[8] - http://play.google.com/intl/en_us/about/overview/index.html/, acesso a 24 de novembro de 2014.

[9] – Eclipse, community for individuals and organizations, http://www.eclipse.org/org/, acesso a 24 de novembro de 2014.

[10] – http:// developer.android.com/reference/android/app/Activity.html/, acesso a 24 de novembro de 2014.

[11] – Android, Official site Provs the Android SDK and documentation for app developers, http://developer.android.com/about/index.html/, acesso a 24 de novembro de 2014.

[12] – IBM, developerWorks, https://www.ibm.com/developerworks/br /library/os-android-devel/, acesso a 24 de novembro de 2014.

[13] - Download the official Android SDK, http://developer.android.com/ sdk/index.html?hl=sk/, acesso a 24 de novembro de 2014.

[14] - Android Development Tools (ADT), custom plugin for the Eclipse IDE, https://developer.android.com/sdk/installing/installing-adt.html

[15] – The MySQL Web site, http://dev.mysql.com/doc/refman/4.1/en/what-is-mysql.html/, acesso a 24 de novembro de 2014.

[16] – The Unified Modeling Language, http://www.uml-diagrams.org/

[17] – android.database.sqlite, http://developer.android.com/reference/ android/database/sqlite/package-summary.html/, acesso a 24 de novembro de 2014.

[18] - POST method, Network Working Group, http://www.ietf.org/rfc/ rfc2616.txt/, página 54, acesso a 24 de novembro de 2014.

[19] - Introducing JSON, http://www.json.org/javadoc/org/json/JSONArray. html/, acesso a 24 de novembro de 2014.

[10] – phpMyAdmin, http://www.phpmyadmin.net/home_page/index.php/, acesso a 24 de novembro de 2014.

[21] – Serializable, Android Developers, http://developer.android.com/ reference/ java/io/Serializable.html, acesso a 24 de novembro de 2014.

[22] – ListActivity, Android Developers, http://developer.android.com/ reference/android/app/ListActivity.html, acesso a 24 de novembro de 2014.

[23] – Base64, Android Developers, http://developer.android.com/reference/ android/util/Base64.html, acesso a 24 de novembro de 2014.

[24] – SensorManager, Android Developers, http://developer.android.com/reference/ android/hardware/SensorEventListener.html/, acesso a 24 de novembro de 2014.

[25] – BroadcastReceiver, Android Developers, http://developer.android.com/ reference/android/content/BroadcastReceiver.html/, acesso a 24 de novembro de 2014.

[26] - SQLiteOpenHelper, Android Developers, http://developer.android.com /reference/android/database/sqlite/SQLiteOpenHelper.html/, acesso a 24 de novembro de 2014.

[27] – Services, Android Developers, http://developer.android.com/guide/ components/services.html/, acesso a 24 de novembro de 2014.

[28] – TelephonyManager, Android Developers, http://developer.android.com/reference/android/telephony/TelephonyManager.html/, acesso a 24 de novembro de 2014.

[29] – org.apache.http, Android Developers, http://developer.android.com/ reference/org/apache/http/package-summary.html/, acesso a 24 de novembro de 2014.

[30] – JSONArray, Android Developers, http://developer.android.com/ reference/org/json/JSONArray.html/, acesso a 24 de novembro de 2014.

[31] – JSONObject, Android Developers, http://developer.android.com/ reference/org/json/JSONObject.html/, acesso a 24 de novembro de 2014.

[32] – BaseAdapter, Android Developers, http://developer.android.com/ reference/android/widget/BaseAdapter.html/, acesso a 24 de novembro de 2014.

[33] – GestureDetector,OnGestureListener, Android Developers, http://developer.android.com/reference/android/view/GestureDetector.OnGestureListener.html/, acesso a 24 de novembro de 2014.

[34] – LinearLayout, Android Developers, http://developer.android.com/reference/android/widget/LinearLayout.html/, acesso a 24 de novembro de 2014.

[35] – UI Overview, Android Developers, http://developer.android.com/guide/topics/ui/overview.html/, acesso a 24 de novembro de 2014.