

**UNIVERSIDADE DO SAGRADO CORAÇÃO**

**MATHEUS LUIZ BOSO**

**RECEITA CONTROL  
APLICATIVO ANDROID PARA CONTROLE DE  
RECEITUÁRIO MÉDICO**

BAURU  
2016

**MATHEUS LUIZ BOSO**

**RECEITA CONTROL  
APLICATIVO ANDROID PARA CONTROLE DE  
RECEITUÁRIO MÉDICO**

Trabalho de Conclusão de Curso apresentado ao Centro de Ciências Exatas e Sociais Aplicadas da Universidade do Sagrado Coração, como parte dos requisitos para obtenção do título de Bacharel em Ciência da Computação, sob orientação do Prof. Me. Márcio Henrique Castilho Cardim.

BAURU  
2016

Boso, Matheus Luiz

B744r

Receita Control: aplicativo Android para controle de  
receituário médico / Matheus Luiz Boso. -- 2016.

46f. : il.

Orientador: Prof. M.e Márcio Henrique Castilho Cardim.

Trabalho de Conclusão de Curso (Graduação em  
Ciência da computação) - Universidade do Sagrado Coração  
- Bauru - SP

1. Aplicativo para Android. 2. Controle de receita médica.  
3. Administração do tratamento. I. Cardim, Márcio Henrique  
Castilho. II. Título.

**MATHEUS LUIZ BOSO**

**RECEITA CONTROL**  
**APLICATIVO ANDROID PARA CONTROLE DE**  
**RECEITUÁRIO MÉDICO**

Trabalho de Conclusão de Curso apresentado ao Centro de Ciências Exatas e Sociais Aplicadas da Universidade do Sagrado Coração, como parte dos requisitos para obtenção do título de Bacharel em Ciência da Computação, sob orientação do Prof. Me. Márcio Henrique Castilho Cardim.

**Banca examinadora:**

---

Prof. Me. Márcio Henrique Castilho Cardim  
Universidade do Sagrado Coração

---

Prof. Me. Henrique Pachioni Martins  
Universidade do Sagrado Coração

---

Prof. Dr. Elvio Gilberto Silva  
Universidade do Sagrado Coração

Bauru, 21 de novembro de 2016.

## RESUMO

O objetivo deste trabalho é construir um aplicativo para Android que possua como principal função o controle de informações de receita médica, com intuito de facilitar a administração do tratamento, alertando o paciente quanto à obtenção de outra receita. O presente trabalho dispõe-se de um estudo das ferramentas utilizadas para seu desenvolvimento. Apresentam-se as medidas prévias, como análise de requisitos e modelagens criadas. Para o desenvolvimento foi utilizada a *IDE* Eclipse em conjunto com o *SDK* Android e o *Plugin ADT* do Eclipse. Conforme teste realizado, o aplicativo teve um bom comportamento. A pesquisa traz ainda um guia de uso, com os principais fluxos de interação do sistema, complementados com *screenshots* da ferramenta.

**Palavras-chave:** Android; Controle; Receita Médica.

## **ABSTRACT**

The objective of this work is to build an Android application that has as main function the control of medical prescription information, in order to facilitate the administration of the treatment, alerting the patient about obtaining another prescription. The present work is a study of the tools used for its development. The previous measures, such as requirements analysis and modeling are presented. For development Eclipse IDE was used in conjunction with the Android SDK and Plugin ADT from Eclipse. As tested, the application performed well. The research also includes a guide to use, with the main flows of interaction of the system, complemented with screenshots of the tool.

**Keywords:** Android; Control; Prescription.

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>7</b>
<b>2</b>	<b>OBJETIVOS</b>	<b>9</b>
2.1	Objetivo geral	9
2.2	Objetivos específicos	9
<b>3</b>	<b>REFERENCIAL TEÓRICO</b>	<b>11</b>
3.1	Interação Humano Computador (IHC)	11
3.1.1	Interação via Touchscreen	12
3.1.2	Interface	13
3.2	Evolução da tecnologia celular móvel	14
3.2.1	Arquitetura de software para dispositivos móveis	15
3.3	Android	16
3.3.1	Arquitetura Android	18
3.3.2	Maquina Virtual Dalvik	19
3.3.3	Bibliotecas Android	20
3.3.4	AVD Manager	20
3.4	Banco de dados	21
3.4.1	Banco de dados orientado ao objeto	21
3.4.2	SQLiteDatabase	23
3.5	Automedicação vs. medicação prescrita	23
3.5.1	Tipos de receituário médico	25
3.6	Trabalhos correlatos	26
3.6.1	MedSus	26
3.6.2	BulasMed	27
3.6.3	SisMaM	28
<b>4</b>	<b>METODOLOGIA</b>	<b>30</b>
4.1	Análise de requisitos	30
4.1.1	Requisitos funcionais	30
4.1.2	Requisitos não funcionais	31
4.2	Modelagem	31
4.2.1	Diagrama de casos de uso	31
4.2.1	Descrição dos casos de uso	32

4.2.1	Manter receita.....	32
4.2.1	Emitir notificação .....	32
4.2.2	Diagrama de classes .....	33
4.2.3	Diagrama de atividades .....	34
4.3	Desenvolvimento .....	35
4.3.1	Estruturas do projeto.....	36
4.3.2	Cadastro .....	39
4.3.3	Minhas receitas.....	40
4.3.4	Editar Receita .....	41
4.3.5	Notificações .....	42
<b>5</b>	<b>RESULTADOS E CONSIDERAÇÕES FINAIS .....</b>	<b>44</b>
<b>5.1</b>	<b>TRABALHOS FUTUROS .....</b>	<b>44</b>
	<b>REFERÊNCIAS .....</b>	<b>46</b>



## 1 INTRODUÇÃO

No Brasil, as doenças crônicas não transmissíveis (DCNT) aumentam em ritmo acelerado, tornando-se prioridade na área de saúde. Apesar dos avanços do Sistema Único de Saúde (SUS) e da Estratégia Saúde da Família (ESF), as DCNT representam a principal causa de mortes (Departamento de Informática do SUS). A venda de medicamentos de automedicação (sem receita) vem promovendo diversos efeitos prejudiciais à saúde da sociedade em geral, que geralmente acaba optando por automedicar-se devido à situação financeira não ser compatível com o alto preço dos medicamentos necessários, ou mesmo com o valor de uma consulta médica.

Assim, poderão surgir algumas consequências desagradáveis; entre elas, está a manifestação de outra doença, resultado principalmente da ingestão de medicamentos que não equivalem ao tratamento adequado, ao qual se tem acesso ao passar por um médico, que fornece a receita.

Existem medicamentos que não são tarjados, ou seja, não necessitam da apresentação de receituário, os medicamentos de automedicação. Como é o caso de analgésicos e outros produtos usados para tratar sintomas ou males menores, como resfriados, azias, dores de cabeça, entre outros. Mas, na compra dos demais medicamentos como antibióticos e tarja preta o paciente deve apresentar a receita médica, que fica armazenada na farmácia onde é feito todo um arquivo que é fiscalizado pela Vigilância Sanitária do município.

A data de vencimento da receita é bastante importante, pois se ela estiver vencida não permitirá o acesso aos medicamentos prescritos. Para que isso não seja sinônimo de ingestão de fármacos convencionais que não correspondem ao quadro medicamentoso da terapia, podendo acarretar outros males à saúde, o desenvolvimento e a adaptação de equipamentos tecnológicos móveis para controlar o vencimento de uma receita médica é um desafio em busca de uma melhoria na questão do controle de tratamento medicamentoso. O computador móvel é um equipamento extremamente simples, com interface gráfica muito parecida com o Windows e que pode ser levado para onde formos.

Tendo como base tais informações, foi posicionado o tema do trabalho afim de que se possa ter uma melhoria no controle da receita médica, lembrando o usuário quando o vencimento estiver próximo. É destinado a pessoas que utilizam do

manuseio da mesma, evitando assim, que não procure uma farmácia sem antes consultar um especialista.

Tendo em vista esta facilidade, o presente trabalho tem como foco um aplicativo para dispositivos móveis, o qual será desenvolvido em Java e estará acoplado à plataforma Android com uma base de dados para cadastro de informações referentes aos dados contidos na receita, como: vencimento, nome do medicamento, dose e intervalo de medicação.

Enfim, será um sistema de emissão de alertas de vencimento no visor do celular para controle do usuário. O sistema incluirá uma interface de fácil manuseio composta por botões para que se possa cadastrar os dados e efetuar eventuais consultas às receitas anteriores, para se obter alguma informação de referência.

## 2 OBJETIVOS

Neste capítulo, tratar-se-ão dos objetivos que o trabalho pretende atingir, começando pelo objetivo geral, que se apresenta com um texto resumido, apontando uma visão única do projeto; seguido dos objetivos específicos, com informações detalhadas por etapas.

### 2.1 Objetivo geral

Desenvolver um ambiente com tecnologia Android, a fim de mostrar um método para se fazer o controle de receita médica utilizada para fins de tratamento médico baseado em medicamento prescrito por órgão regulador.

### 2.2 Objetivos específicos

- a) Reunir Informações necessárias para controle de receitas médicas de pacientes que não podem ficar sem medicamentos prescritos.
- b) Pesquisar sobre dispositivos móveis, ferramentas, evolução e sua interação humano-computador.
- c) Levantar conteúdos referentes a plataforma Android, bem como banco de dados e SGDB para sistemas embarcados SQLite.
- d) Mapear uma aplicação que auxilie o paciente a saber quando precisa comprar o devido medicamento.
- e) Modelar uma base de dados para cadastro dos dados referente à receita.
- f) Criar um banco de dados para armazenar os conteúdos que formarão a base de conhecimento, contribuindo assim para facilitar a manipulação das informações a serem trabalhadas possibilitando alterar e excluir dados.
- g) Adaptar o usuário que utiliza receitas médicas à tecnologia atual de comunicação móvel Android.
- h) Desenvolver uma aplicação Android capaz de coletar informações de dados da receita – como nome do medicamento, dosagem, horário e a data de validade da receita – que atenda os objetivos mencionados acima.

- i) Testar o aplicativo no celular Samsung com Sistema Operacional Android 4.4.2 (KitKat).

### 3 REFERENCIAL TEÓRICO

Este capítulo apresenta alguns conceitos de softwares e informações relevantes ao contexto que agrega o trabalho; além disso, traz informações sobre hábitos medicamentosos da população.

#### 3.1 Interação humano computador (IHC)

Nos dias atuais, o impacto da tecnologia na vida das pessoas tem crescido em ritmo acelerado apresentando aparelhos cada vez mais sofisticados com inúmeros recursos, tanto de hardware como de software, atingindo um público aleatório de diferentes padrões de usabilidade. Além disso, crianças que ainda não foram alfabetizadas conseguem obter resultados na interação com tais dispositivos. Inegavelmente, a evolução das interfaces humano computador teve um papel decisivo neste processo. Desta forma, não há como contestar a contribuição da área para a inclusão digital. Apesar disto, muito ainda tem que ser feito, na área de interação humano computador, a favor dos excluídos da sociedade da informação (CARVALHO, 2003).

Segundo Barbosa (2010, p.14):

É importante lembrar que, embora os custos de desenvolvimento possam aumentar ligeiramente, o investimento nessa área traz benefícios sempre que o sistema for utilizado e para todos os envolvidos com seu uso seja direta ou indiretamente, ao longo de toda a vida útil do sistema.

As técnicas e ferramentas para desenvolvimento dos sistemas para dispositivos móveis, em evidência Android, exigem bastante atenção aos detalhes e podem ter custos mais elevados, porém, poderá ser bem valorizado posteriormente se atingir uma qualidade de uso necessária aos diferentes perfis de usuário existentes.

O crescente interesse da indústria de software pela usabilidade para produzir sistemas interativos, acabam desenvolvendo interfaces inadequadas para alguns perfis de usuário deixando de lado um fator importante, a ergonomia. Conceito pelo qual é definido pelo método de proporcionar um bem estar entre o usuário e o sistema proporcionando uma satisfação na utilização do mesmo, a fim de se evitar

uma insatisfação ou frustração no manuseio do sistema (MATIAS; HEEMAN; SANTOS, 2001).

Para uma melhor satisfação de uso devemos considerar alguns elementos importantes envolvidos na interação humano computador, que podem ser úteis no desenvolvimento do sistema proporcionando uma interface amigável de acordo com o usuário que irá acessar. Tais elementos apresentam uma breve definição nos tópicos mencionados a seguir.

### **3.1.1 Interação via Touchscreen**

Uma abordagem que tem evoluído ao longo dos anos é o conceito de interação. No começo, por volta da década de 1980, os sistemas comerciais eram mais simples, realizavam operações básicas, como: gerenciar arquivos e funções, abrir e remover informações, com acesso emitido pelo usuário através de comandos via teclado. Além do mais, a tela do monitor era grande e pesada a ponto de dificultar o seu manuseio. A tela monocromática CRT apresentava os mesmos recursos gráficos de TV da época, porém, com uma visão mais perceptível, não envolvendo interfaces gráficas e aplicações com recursos avançados de áudio e imagem.

Conforme os anos foram passando, a tecnologia passou por muitas mudanças em relação à usabilidade dos dispositivos, apresentando equipamentos menores e mais rápidos, razão pela qual agregou uma nova funcionalidade: o *Touchscreen*.

Conforme relatam Palacio e Cunha (2012, p. 673), este conceito pode ser definido como:

Denominam-se gestos tácteis, aqueles realizados pelo usuário a partir de movimentos dos dedos sobre a tela do dispositivo sensível ao toque (*touchscreen*). Entende-se que a comunicação entre aplicativo e usuário ocorre por meio destes gestos, substituindo ou complementando diversos mecanismos tradicionais de entrada, tais como mouse e teclado.

Através da Figura 1 demonstra-se uma tela *touchscreen*, com o mecanismo de toque entre usuário e dispositivo.

Figura 1 – Tela *touchscreen*



Fonte: Brasilescola (2016).

Com o passar dos anos, a tendência evolutiva da tecnologia aumentou em ritmo acelerado, apresentando telas mais compactas e sistemas com recursos gráficos com maior capacidade de processamento, gerando interfaces com inúmeras cores. Pensando nisso e para facilitar a vida dos usuários, os programadores começaram a trabalhar no design de interfaces proporcionando uma melhor praticidade para trabalhar com o sistema, razão pela qual foi associado o uso de outros dispositivos de interação, tais como: o mouse, óculos estereótipos e microfone.

### **3.1.2 Interface**

Existem vários estudos com a finalidade de descobrir as melhores formas de imagens e de textos indicados para não carregar nem confundir a visão dos usuários, inclusive dos que utilizam dispositivos móveis. Passando por Design gráfico voltado para melhores resultados que se destinam aperfeiçoar a percepção visual dos seres humanos, até os estudos de usabilidade para interfaces gráficas interativas, todos esses estudos visam minimizar a carga de trabalho e maximizar os efeitos positivos no entendimento das ações para os processos de interação dos envolvidos na utilização do sistema (GUEDES, 2009).

A inserção de interfaces gráficas em conjunto com a interação de software se formalizou com a intenção de proporcionar uma melhor maneira de organização dos

dados. O meio de armazenamento disponível na época não comportava tantas informações, tornando difícil à procura em meio a muitos dados.

O desenvolvimento apropriado das interfaces entre o ser humano e o computador é uma tarefa complexa, pois envolve os aspectos humanos das relações. Pela sua complexidade, o conceito envolvido na interação exige uma abrangência multidisciplinar, incluindo estudos ligados a aspectos e comportamentos humanos, algo que vai além da computação (CARVALHO, 2003).

### **3.2 Evolução da tecnologia de celular móvel**

No contexto envolvendo dispositivos móveis, a comunicação sem fio tem sido usada muito antes das redes celulares, com as emissões via rádio AM e FM; as comunicações navais e a própria televisão (SANTANA, 2008).

Sendo assim, se aproximando da atual realidade comunicativa, nota-se um grande avanço e popularização de dispositivos computacionais de pequeno porte, tais como, celulares e tablets. Há mais usuários de celulares e equipamentos móveis do que usuários de PCs. E a procura por celulares no mercado tende ser muito maior devido às diferenças de preço entre esse equipamento e um sofisticado microcomputador (TAURION, 2002).

Dentre as atuais funcionalidades dos celulares de hoje, podemos utilizar a internet, trocar mensagens instantâneas, jogar, tirar fotos online e compartilhá-las com amigos, ler e editar textos, criar vídeos e personalizar o smartphone a nosso gosto e vontade, os primeiros celulares foram fabricados com o objetivo de tentar estabelecer uma ligação telefônica em um aparelho que pudesse ser movido de um lugar a outro e não estivesse conectado a nenhuma base (TECHTUBE, 2014).

Segundo Deitel (2013, p. 9):

A demanda por dispositivos móveis está aumentando à medida que mais pessoas contam com smartphones e tablets para se conectar e produzir enquanto estão longe de seus computadores pessoais. De acordo com a comScore, 234 milhões de norte-americanos usaram dispositivos móveis em um período de três meses terminando em julho de 2011. Desses assinantes, 40,6% usaram aplicativos.

Levando em consideração os meios de comunicação, devido à necessidade de padronização para o sistema celular e a crescente demanda pelo serviço móvel, foi necessário dar início ao desenvolvimento de sistemas digitais para dispositivos



móveis. Os sistemas começaram a ser efetivamente utilizados no início de 1990, e isso foi impulsionado pelo avanço da tecnologia dos circuitos integrados, que permitiram a efetiva utilização da transmissão digital (ALMEIDA, 2013).

Com base na evolução dos dispositivos de telefonia celular, destaca-se a Figura 2, começando pelos aparelhos de grande dimensão até chegar nos de menor porte.

Figura 2 – Evolução de celulares móveis



Fonte: Ismael Carvalho (2012).

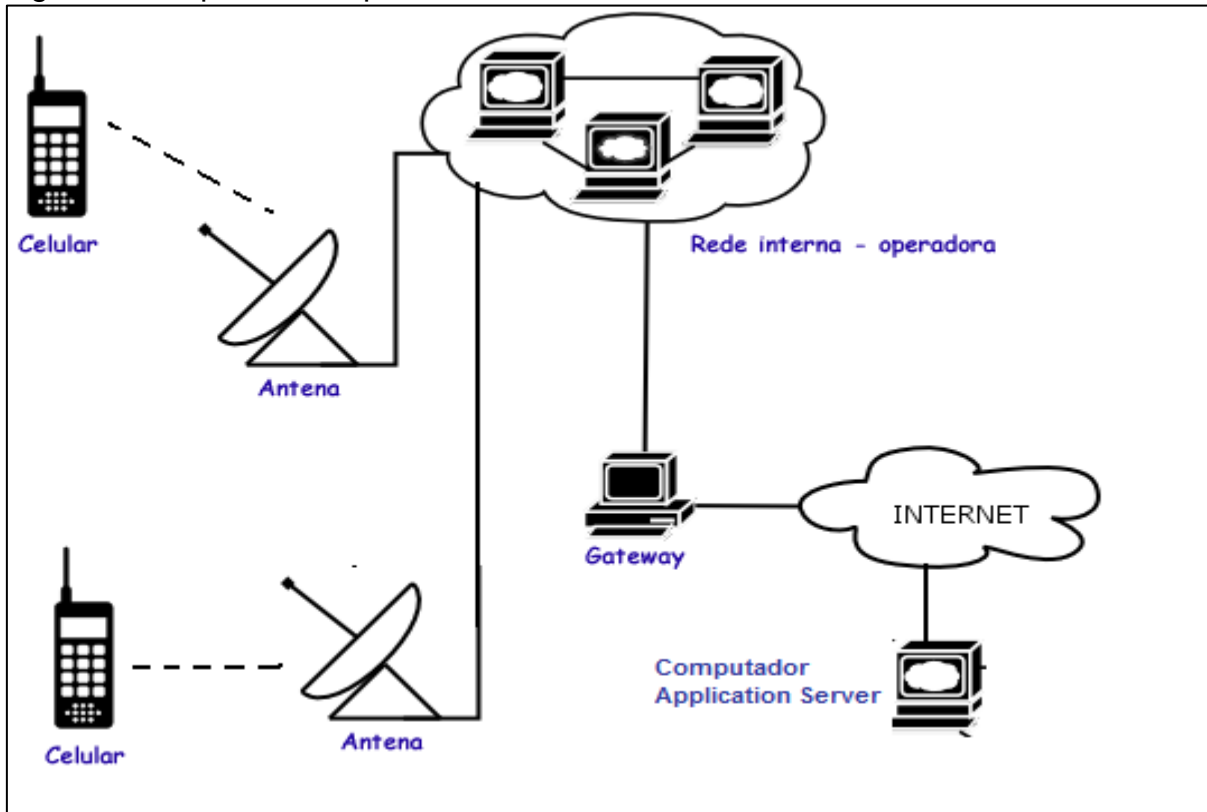
### 3.2.1 Arquitetura de software para dispositivos móveis

Basicamente, um celular tem como serventia sua utilização em aplicações para se comunicar com outros celulares e também para navegar na internet. Com o dispositivo, é possível enviar mensagens SMS/MMS, localizar-se por georreferenciação (GPS), e ainda utilizar softwares para entretenimento, como jogos, interação com redes sociais, manipulação de imagens, etc.

Para Lee, Schneider, Scheel (2012), posicionando o desenvolvimento no contexto empresarial de negócios, as aplicações móveis são construídas a fim de proporcionar soluções para melhorar o desempenho e tornar móvel seu esforço de trabalho. Dentre os candidatos envolvidos com usabilidade, estão os profissionais em trânsito local e remoto, agentes de vendas e certos profissionais de escritório.

De acordo com a Figura 3, podemos ter uma noção da arquitetura de dispositivos móveis, tendo como exemplo sua funcionalidade na comunicação com outros celulares e utilização da internet.

Figura 3 – Arquitetura dispositivos móveis



Fonte: Frameworks (2012).

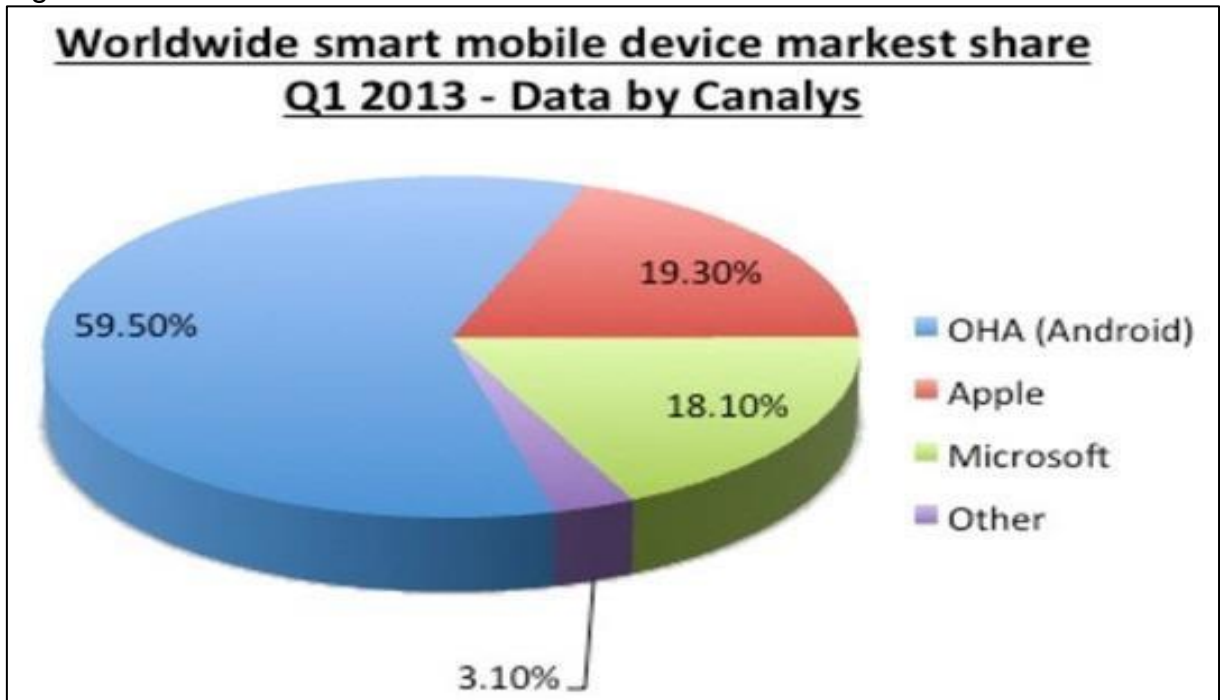
### 3.3 Android

A plataforma Android é uma tecnologia utilizada no desenvolvimento de aplicativos para dispositivos móveis. Estando no mercado a mais de seis anos, esta foi a plataforma escolhida por muitos fabricantes de celular, destacando-se a Samsung. Por ser uma plataforma utilizada por uma grande maioria de pessoas, esse foi o motivo de escolha para rodar o aplicativo.

A plataforma Android atualmente é o principal sistema operacional para dispositivos móveis, seguido pelo iOS da Apple, conforme estudo de mercado produzido pela empresa Canalsys (CANALYS, 2013). Este estudo mostra que durante o primeiro trimestre de 2013, a plataforma Android representa cerca de 59,5% dos smartphones vendidos em todo o mundo, a Apple, com o iOS, segue a

plataforma da Google, com cerca de 19,3% do mercado e, por fim, a Microsoft com 18,1%, conforme mostra a Figura 4.

Figura 4 – Estatística de Plataformas Mobile



Fonte: Canalys (2013).

Para iniciar a programação de aplicativos acoplados a essa tecnologia, é utilizada a codificação em Java com princípios XML. Para isso, devemos preparar o ambiente de desenvolvimento inicial pelo qual é dividido em kits de software e IDE de desenvolvimento:

- a) **JAVA JDK** – (*Java Development Kit*): É um Kit de Desenvolvimento Java, ou seja, um conjunto de utilitários que permitem criar sistemas de software, traz consigo a API do Java com todas as classes necessárias e outras ferramentas utilitárias.
- b) **ANDROID SDK** – (*Software Development Kit*): Conjunto de ferramentas necessárias para desenvolvimento de aplicativos Android, incluindo a plataforma e o emulador que simula uma tela de celular.
- c) **IDE** – (*Integrated Development Environment*): É uma ferramenta de desenvolvimento que compreende vários tipos de linguagens, e que aceita a instalação de plug-ins para emular o desenvolvimento da plataforma.

As aplicações android dependem do software Android SDK, pois contém um emulador para simular uma tela de celular, além de incluir ferramentas utilitárias e uma API para atender a linguagem Java; sem ele, fica impossível completar o processo de desenvolvimento do aplicativo por não conter essas ferramentas (LECHETA, 2013).

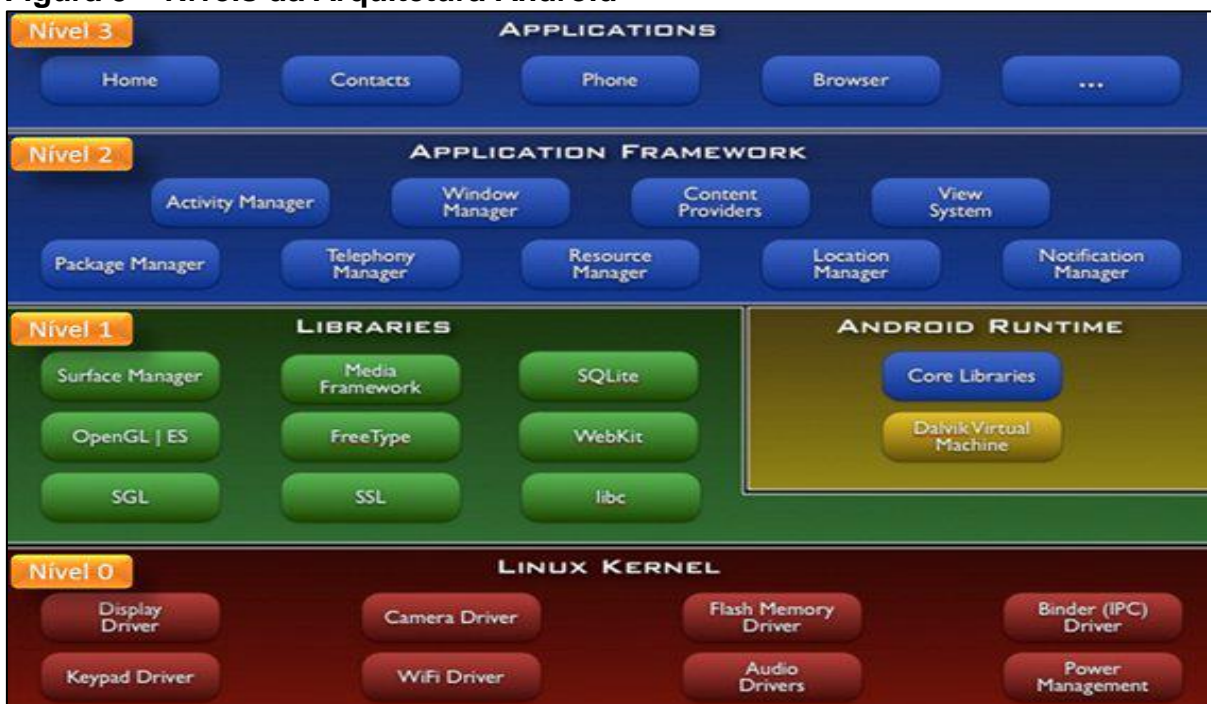
Conforme relata Deitel (2013, p.18), “o SDK (Software Development Kit) do Android fornece as ferramentas necessárias para construir aplicativos Android.” Um desenvolvedor escreverá o código na linguagem de sua escolha, porém ele também vai precisar do SDK para completar o produto e implantá-lo na plataforma de sua preferência.

### 3.3.1 Arquitetura Android

A arquitetura do sistema Android é estruturada em um modelo de camadas. Essas camadas têm suas características próprias e juntas dão sentido ao montante das aplicações para execução.

Os níveis estão divididos de zero a três; esses conceitos são exemplificados na Figura 5.

**Figura 5 – Níveis da Arquitetura Android**



Fonte: TI selvagem (2011).

No nível zero, encontramos a base para a execução das aplicações em Android. Nesta camada, localiza-se o sistema e drivers, no entanto contém apenas uma pequena parte do sistema operacional, o núcleo garante que todos os outros processos no sistema sejam sincronizados para trabalhar em conjunto corretamente (TI SELVAGEM, 2011).

No nível acima do kernel ficam as bibliotecas C/C++, na qual carregam consigo componentes utilizados pelo sistema, como: uma inclusão da biblioteca padrão do C (libc); bibliotecas para suporte a formatos de áudio, vídeo e imagens; um gerenciador que intermedia o acesso ao display e compõe as camadas de imagem 2D e 3D; o engine para navegadores Web; bibliotecas para gráficos 2D (SGL) e 3D (OpenGL ES); um renderizador de fontes bitmap e vetoriais; e o banco de dados SQLite (PEREIRA; SILVA, 2009).

No nível dois, temos a camada de framework de aplicação que se constitui de programas que gerenciam as aplicações básicas do telefone como GPS, recursos gráficos e de áudio e disponibilização de dados. Os desenvolvedores têm acesso aberto ao framework como um conjunto de ferramentas básicas com o qual poderá construir ferramentas mais complexas (PAULA, 2013).

No nível três, encontra-se a camada de aplicações e as funções básicas do dispositivo. Esta é a camada em que permite que o usuário, de forma mais clara, interagir com o sistema e o dispositivo móvel, nela encontramos aplicativos cliente de e-mail, programa de SMS, calendário, mapas, navegador, contatos entre outros (TI SELVAGEM, 2011).

### **3.3.2 Máquina Virtual Dalvik**

Apesar de a linguagem Java ser utilizada para construir aplicações para Android, não existe uma máquina virtual Java (JVM) em seu sistema. Pensando nisso, foi desenvolvida uma máquina virtual especialmente para dispositivos móveis chamada máquina virtual Dalvik.

De acordo com Lecheta (2010, p. 24):

Ao desenvolver as aplicações para o Android você vai utilizar a linguagem Java e todos os seus recursos normalmente, mas depois que o bytcode (.class) é compilado ele é convertido para o formato .dex (Dali Executable), que representa a aplicação do Android compilada. Depois disso, os arquivos .dex e outros recursos como imagens são compactados em um único

arquivo com a extensão .apk (Android Package File), que representa a aplicação final, pronta para ser distribuída e instalada).

A Dalvik é uma máquina virtual com melhor desempenho, projetada para maior integração com novas gerações de hardware para executar várias máquinas virtuais paralelamente, com processadores de baixa frequência, pouca memória ram, sem espaço para Swap, consumo mínimo de memória, bateria e processador (PEREIRA; SILVA, 2009).

### 3.3.3 Bibliotecas Android

O ambiente de desenvolvimento Android inclui um conjunto de bibliotecas C/C++, usadas por diversos componentes do sistema. Elas têm a capacidade de apresentar funcionalidades aos desenvolvedores através da estrutura de aplicativos Android. Algumas das principais bibliotecas são:

1. Sistema C library - Derivadas da biblioteca padrão do sistema C (libc), configurada e acoplada para dispositivos baseados em Linux.
2. Bibliotecas de mídia – Tendo como base OpenCORE e PacketVideo, que são as bibliotecas de reprodução e gravação de áudios populares e muitos formatos de vídeo, bem como arquivos de imagem estática.
3. Gerente de superfície - Gerencia o acesso ao subsistema de display, compõe camadas gráficas 2D e 3D de múltiplas aplicações.
4. LibWebCore - Um navegador que alimenta tanto o navegador do Android como uma vista web, ou seja, tecnologia Google Chrome que permite as aplicações Android apresentar conteúdo web.
5. SGL - Contém os arquivos subjacentes ao motor gráfico 2D.
6. Bibliotecas 3D - As bibliotecas usam Dispositivo de aceleração 3D altamente otimizado para aplicativo 3D.
7. FreeType - Apresenta bitmap e renderização de fontes em vetor.

### 3.3.4 AVD Manager

A emulação do Android ocorre através do kit SDK do Android, o qual contém o AVD Manager, uma espécie de emulador com os recursos necessários para rodar o ambiente funcional de um sistema Android, dando a sensação de realismo em

comparação ao celular e podendo ser executado em diferentes sistemas operacionais. Conforme relata Deitel e Morgano (2013, p.19), “o emulador do Android, acrescido do SDK do Android, permite executar aplicativos em um ambiente simulado dentro do Windows, Mac OS ou Linux.”

Para isso, é necessário criar um AVD (Android Virtual Device), que exige como requisito um *plug-in* para Eclipse (ADT) que permite executar aplicativos Android, com intuito de proporcionar um ambiente integrado e poderoso para as aplicações. Uma AVD consiste de uma configuração preliminar que enfoca o dispositivo móvel receptor, o que irá testar o aplicativo. A partir deste momento, são requisitados ao desenvolvedor, alguns dados aos elementos que envolvem o funcionamento do equipamento, incluindo configurações de hardware, imagem de tela, armazenamento de dados entre outras (DEITEL, 2013).

### **3.4 Banco de dados**

Conforme Korth (1994), um banco de dados é uma coleção de dados inter-relacionados, representando informações sobre um domínio específico, organizadas para que possam ser facilmente acessadas, gerenciadas e atualizadas; tais informações podem estar armazenadas em computador ou em um armário, por exemplo.

Pacievitch (2016) explica que, em informática, o termo banco de dados às vezes se refere ao SGBD, que é o Sistema Gerenciador de Banco de Dados, mas a palavra também é empregada para definir uma base de dados, ou seja, o grupo de dados agrupados por um SGBD.

#### **3.4.1 Banco de dados orientado ao objeto**

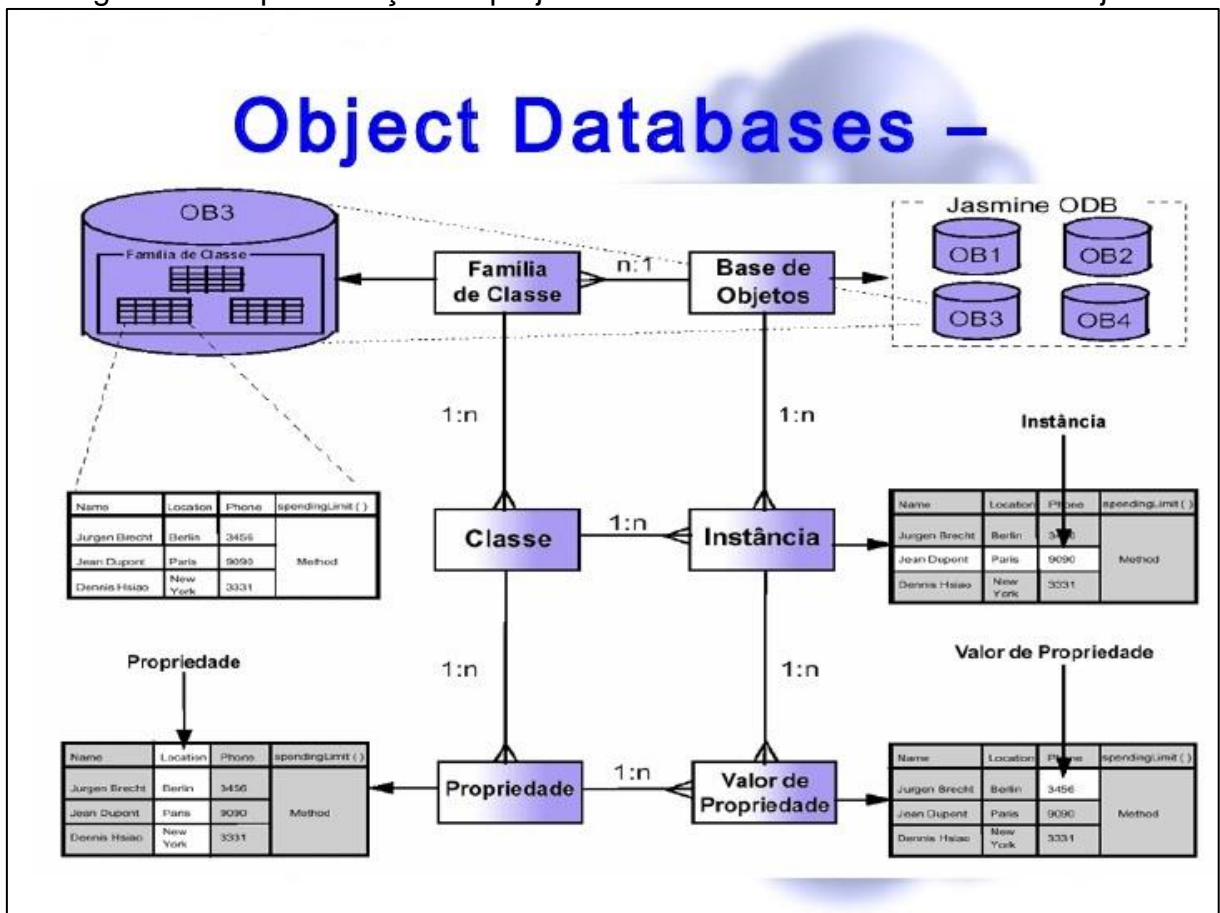
Há princípio, o conceito de banco de dados orientado a objetos se especifica na combinação de ideias de banco de dados tradicionais com linguagem de programação orientada a objetos. A inovação se deu na década de 80, a partir do surgimento de projetos com grande concentração de base de dados, pelo qual seria necessário acrescentá-las no projeto, deixando o trabalho mais demorado e exaustivo. Dentro deste paradigma, existem os sistemas de gerenciamento de banco

de dados relacionais, que consiste em ferramentas gráficas para gerenciar tabelas (DATE, 2003).

Os sistemas de gerenciamentos de banco de dados multiplicaram-se nas últimas décadas, devido ao fato do surgimento de sistemas complexos centrados em manipulação de estruturas pela qual contava com grande demanda de tabelas.

Os atuais bancos de dados são voltados ao conceito de Linguagem Orientada a Objetos (LOO), possuindo além das funções básicas de armazenamento e recuperação de dados, ferramentas para manipulação de muitas bases de dados. Através da construção orientada a objetos, os desenvolvedores podem esconder os detalhes de implementação de seus registros; compartilhar referências a objetos e expandir seu sistema através de módulos existentes. Banco de dados orientados ao objeto se agrega ao modelo de dados relacional, quando na construção de uma nova tabela a mesma precisar manipular registros (CARVALHO, 2011). Pode-se ver na Figura 6 sua utilização.

Figura 6 – Representação de projeto de banco de dados orientado ao objeto



Fonte: UnB (2009).



### 3.4.2 SQLiteDatabase

De acordo com Kreibich (2010, p. 1), SQLite se define como um pacote de software de domínio público que fornece um sistema gerenciador de banco de dados relacional funcional e flexível, indicado para sistemas embarcados o pacote vem incluso em todos os sistemas projetados para Android não sendo necessário instalação. SQLite foi projetado para consumir o mínimo de recursos necessários e eliminar a complexidade normalmente encontrada em gerenciadores de banco de dados tradicionais.

Kreibich (2010, p. 2-6) aponta suas principais características, como:

- a) **Sem servidor:** Não possui uma arquitetura cliente/servidor (não tem um servidor separado), o motor do banco de dados inteiro é integrado ao aplicativo e o acesso é feito através de leitura e escrita diretamente nos arquivos de banco de dados armazenados no disco.
- b) **Arquivo único:** O banco de dados inteiro é empacotado em um único arquivo independente de plataforma, podendo ser acessado em qualquer dispositivo, ignorando o formato nativo de armazenamento em disco.

### 3.5 Automedicação vs. medicação prescrita

Este capítulo apresenta algumas informações sobre os hábitos medicamentosos da população hoje.

Observando a evolução na cultura médico farmacêutica, será possível identificar que cada vez mais os indivíduos estão deixando o hábito de ingerir medicação prescrita sob orientação médica, e adotam um fator preocupante, a automedicação.

Com base nesse contexto, é necessário que o medicamento seja prescrito adequadamente, na forma farmacêutica, doses e período de duração do tratamento; que esteja disponível de modo oportuno, a um preço acessível, e que responda sempre aos critérios de qualidade exigidos, que se dispense em condições adequadas, com a necessária orientação e responsabilidade, e, finalmente, que se cumpra o tratamento já prescrito, da melhor maneira possível (AQUINO, 2007).

É evidente que o risco dessa prática está correlacionado com o grau de instrução e informação dos usuários sobre medicamentos, bem como com a

acessibilidade dos mesmos ao sistema de saúde. Provavelmente, a qualidade da oferta de medicamentos e a eficiência do trabalho das várias instâncias que controlam este mercado também exercem papel de grande relevância nos riscos implícitos na automedicação. A necessidade da população em obter uma receita médica devido a fatores socioeconômicos ou profissionais é uma das principais causas dessa mudança no quadro medicamentoso (ARRAIS, 1997).

Assim, opta-se por opções mais práticas e baratas de medicação, como os medicamentos de uso simples e comum (analgésicos, antitérmicos, etc.), facilmente encontrados em qualquer farmácia e que dispensa o tempo gasto na ida ao médico para obtenção de uma receita, porém aliviando os sintomas momentaneamente e não produzindo o efeito esperado.

Conforme demonstra a Figura 7 a qual se apresenta como um comparativo de frequência de uso da automedicação.

Figura 7 – Frequência da automedicação

Classes terapêuticas de maior frequência na prática de automedicação avaliada.			
Classes terapêuticas	Eventos (n)	Frequência relativa(%)	Frequência cumulativa(%)
Analgésicos, antipiréticos e antiinflamatórios	38	44,7	44,7
Medicamento para o trato gastrointestinal	9	10,6	55,3
Suplementos minerais e vitamínicos	6	7,1	62,4
Cardiovasculares	6	7,1	69,5
Antialérgicos	5	5,9	75,4
Outros (alopáticos)	9	10,6	86,0
Medicamentos caseiros ou fitoterápicos	12	14,0	100,0
<b>Total</b>	<b>85</b>	<b>100,0</b>	<b>100,0</b>

Fonte: Ciência & Saúde Coletiva (2006).

Levando em consideração o motivo do uso relacionando com a classe terapêutica, conclui-se que: a ocasião mais comum é a dor de cabeça, seguido de problemas do trato respiratório, que leva à procura por medicamentos específicos como os analgésicos, antipiréticos e anti-inflamatórios. Em sucessão, a tabela se completa pelos grupos dos medicamentos para o trato gastrointestinal, suplementos minerais e vitamínicos, medicamentos para o sistema cardiovascular e antialérgicos

que, em conjunto (BORTOLON; MEDEIROS; NAVES; KARNIKOWSKI; NÓBREGA, 2006).

Se por um lado a prática da automedicação tem impacto positivo sobre a adesão, por outro, poderão surgir efeitos negativos para aqueles que não optam por medicamentos prescritos.

De maneira mais ampla, Remondi, Cabre e Souza (2014, p. 129) ressaltam que:

Entre as consequências da não adesão, é possível pontuar a diminuição do controle efetivo das doenças, o aumento no risco de hospitalizações e o aumento da mortalidade. A avaliação em extensão do impacto da não adesão representa um desafio, pois não há consenso sobre a forma de mensuração e há pouca disponibilidade de informações que permitam analisar fator associado a eventuais complicações, hospitalizações, mortes e custos diretos e/ou indiretos.

Existem hoje profissionais capacitados para auxiliar as pessoas no controle da medicação, além de sugerir outros métodos para prevenir posteriores problemas relacionados à saúde, porém, os custos podem ser elevados e os tratamentos rigorosos.

O mercado farmacêutico está concentrado nos países economicamente mais ricos e voltado às classes sociais mais abastadas. Sabe-se que os maiores investimentos estão no desenvolvimento de tratamentos para problemas que atingem essas populações, em detrimento dos medicamentos demandados pelas necessidades mais comuns nos países mais pobres.

Diante dos fatos, podemos revelar que devido à classe social a não adesão a uma consulta médica para assim aderir um tratamento torna-se muito frequente, o que mantém o hábito de tomar medicamentos mais baratos e simplesmente conversar com o farmacêutico (a).

### **3.5.1 Tipos de receituário médico**

No Brasil é considerado crime comercializar medicamentos sem receitas, deixando quem for pego praticando tais atividades sujeito às medidas impostas pelas leis federais 5.991/73 e 9.787/99 e pela Resolução 357/01 do Conselho Federal de Farmácia (MADRUGA; SOUZA, 2012).

A receita deve ser escrita à tinta, em letra de forma, clara e por extenso.

- a) **Receita simples** – utilizada para a prescrição de medicamentos de tarja vermelha, com o anúncio “venda sob prescrição médica”. Validade: 10 dias.
- b) **Receita de controle especial** – utilizada para a prescrição de medicamentos de tarja vermelha e tarja preta. Validade: 30 dias.

### 3.6 Trabalhos correlatos

Para este tópico, serão apresentados alguns softwares para desktop e para celulares Android que auxiliam no controle de dados relacionados a medicamentos.

Hoje, praticamente toda a população possui acesso a computadores e Internet e pensando em atender a necessidade de muitos, já existe na Internet opções de softwares (gratuitos ou não) para serem utilizados no auxílio de obter informações sobre medicamentos como posologia, forma de administração ou grupo fármaco-terapêutico (LEWANDOWSKI, 2011).

#### 3.6.1 MedSus

Como um primeiro exemplo, temos o aplicativo MedSUS para Android, representado na figura 8.

Figura 8 – Telas do aplicativo



Fonte: Hammerschmidt; Spinilo (2015).

Este aplicativo funciona como um conjunto de ferramentas no intuito de auxiliar o usuário a obter conhecimento de um certo medicamento no contexto de tecnologia móvel. Com base nos dados de um acervo de informações padrões, o mesmo apresenta como uma forma de consulta contendo bulas digitais podendo exibir seu princípio ativo e preço, além de nome comercial, subgrupo fármaco-terapêutico e apresentação de dosagem. O programa também possui uma opção para calcular o valor final se for efetuar uma compra de medicamentos.

### 3.6.2 BulasMed

Outro aplicativo interessante dentro do contexto de medicamentos é o BulasMed, o aplicativo permite a busca por medicamentos, laboratórios, substâncias ou ações terapêuticas. Entre os recursos do utilitário, outras funcionalidades incluem: navegação rápida pelos tópicos da bula através de um índice, marcação de bulas como favoritas e acesso às últimas bulas visualizadas e às bulas favoritas na tela inicial do aplicativo (BULASMED PROJECT, 2016).

Figura 9 – Tela de busca



Fonte: BulasMed project (2016).

Este programa apresenta-se como uma forma para se visualizar as bulas dos medicamentos, já que bulas digitais são mais fáceis que os métodos convencionais, o armazenamento dos dados possui uma base contendo todo o conteúdo do site <http://www.bulas.med.br/> sem a necessidade de conexão com a internet.

### **3.6.3 SisMaM**

Outra solução interessante para se trabalhar com medicamentos é um protótipo de software desenvolvido para ser utilizado em dispositivos habilitados a ler código de barras. Em SisMAM, Soto; Pablló; Campos (2006) propuseram uma solução para auxiliar o processo de administração de medicamentos em nível hospitalar.

O protótipo do sistema móvel tem um diferencial em comparação a outros sistemas, já que foi desenvolvido para ser executado em um dispositivo móvel com utilitários acoplados para leitura e processamento de códigos de barra, de modo que o mecanismo de entrada de dados da aplicação é fortemente baseado na leitura desses códigos.

Por ser um aplicativo de administração em nível hospitalar, a identificação no sistema é efetuada por um funcionário do estabelecimento, no caso, uma enfermeira. A identificação digital do paciente é efetuada através do código disponível em uma pulseira de pulso em conjunto ao utilitário do dispositivo ambos utilizados como ferramentas para inserção de dados. A Figura 10 mostra a tela com a lista de pacientes.

Figura 10 – Tela de pacientes e medicamentos



Fonte: arquivo do autor.

A lista de medicamentos exibe a representação gráfica de cada tipo de medicação. Cada elemento da lista tem três características: um ícone identificador, uma descrição textual da medicação e uma caixa colorida que indica o estado do processo de administração, complementam. Através dessa lista, é possível incluir medicamentos e administrar o comportamento de consumo dos pacientes, complementam. Tendo como exemplo representado na Figura 9, a qual se refere à interface contendo a lista de medicamentos.

Desta forma, além das funções apresentadas anteriormente, o sistema também contempla outras funcionalidades como visualização de dados do paciente detalhes dos medicamentos prescritos para a separação e preparação do mesmo, entre outros.

## **4 METODOLOGIA**

As pesquisas exploratórias têm como objetivo proporcionar maior familiaridade com o problema, com intenção de torná-lo mais nítido no contexto social. Pode-se dizer que estas pesquisas têm como objetivo principal o aprimoramento de ideias ou a descoberta de intuições. Seu planejamento é, portanto, bastante flexível de modo que possibilite ao projeto a consideração dos mais variados aspectos relativos ao fato estudado (GIL, 2010).

O aplicativo em evidência pode ser representado como um novo paradigma computacional que permite que usuários desse ambiente tenham acesso a serviços independentemente de sua localização, podendo inclusive, estar em movimento. Mais tecnicamente, é um conceito que envolve processamento, mobilidade e comunicação sem fio.

### **4.1 Análise de requisitos**

Segundo (PRESSMAN, 2011) a análise de requisitos possibilita o engenheiro do software ou analista especificar as interfaces do sistema, bem como as funções e restrições que este deve possuir. A especificação de requisitos proporciona ao desenvolvedor critérios para avaliar se o software foi desenvolvido corretamente para atender as necessidades para a qual foi desenvolvido.

A seguir serão descritos os requisitos funcionais, não funcionais para a solução proposta neste trabalho.

#### **4.1.1 Requisitos funcionais**

1. O aplicativo deverá permitir ao usuário cadastrar informações de receita;
2. O aplicativo deverá permitir ao usuário consultar receita por nome ordenado de medicamento;
3. O aplicativo deverá permitir ao usuário listar registros para alterar dados;
4. O aplicativo deverá permitir ao usuário listar registros para excluir dados;
5. O aplicativo deverá emitir ao usuário alertas como forma de lembrete para determinado vencimento de receita próximo;



### 4.1.2 Requisitos não funcionais

1. O aplicativo deverá funcionar em dispositivos com o sistema operacional Android;
2. O aplicativo deverá armazenar as informações no banco de dados SQLite;
3. A interface do aplicativo deverá permitir ao usuário manipular seus componentes através de toques (*touch screen*);
4. A interface do aplicativo deverá ser simples e intuitiva.

## 4.2 Modelagem

A modelagem de dados consiste em um estudo no sentido de prever como serão construídas as estruturas de dados que darão suporte aos processos do software. Além disso, possibilita estabelecer uma definição de como esses dados estarão organizados e quais os relacionamentos que se pretende estabelecer entre eles. No decorrer da modelagem, o programador deve se atentar nos fatos dentro do contexto problema e realidade, para assim propor uma solução que automatize as necessidades do usuário (DEBASTIANI, 2015).

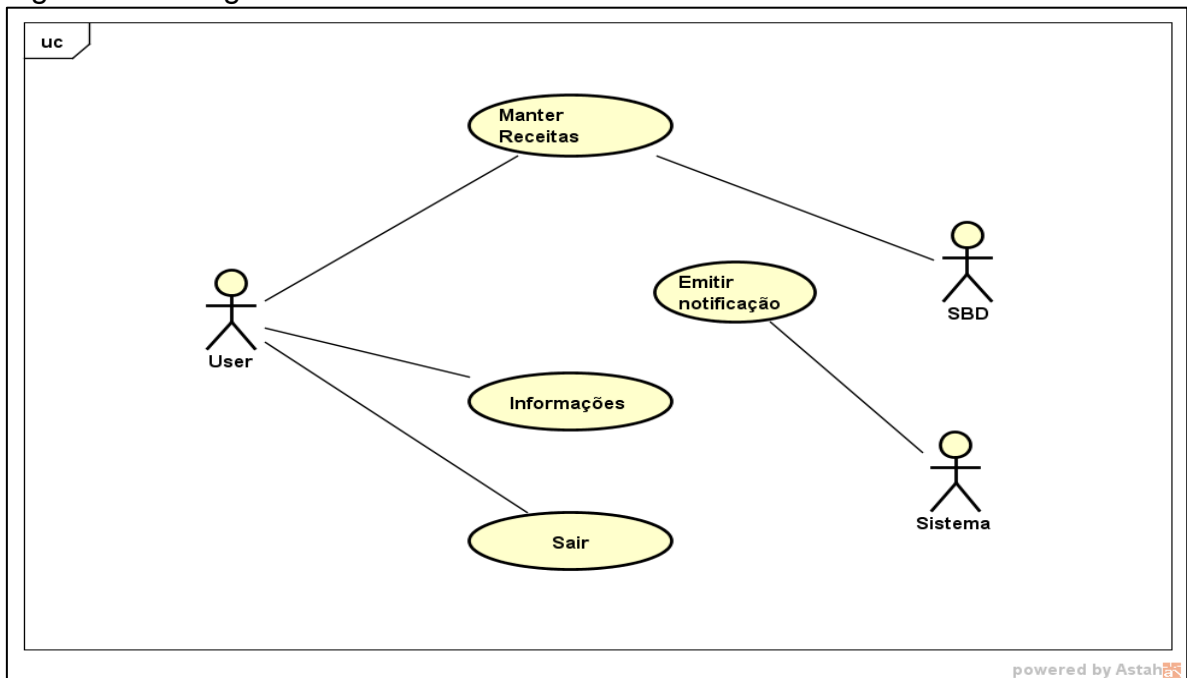
Com o surgimento da UML (Unified Modeling Language), os desenvolvedores de softwares orientado a objetos se entusiasmaram com os conceitos, já que são muito úteis para agilizar e organizar o processo de construção do software (ETELG, 2016).

Para tanto optou-se por utilizar o software Astah na modelagem. Astah, conhecido anteriormente por JUDE é uma ferramenta que oferece ao desenvolvedor a possibilidade de elaborar diagramas e documentar o software a ser criado.

### 4.2.1 Diagrama de Caso de Uso

Através do esboço do diagrama de caso de uso é possível verificar o comportamento do usuário com as funcionalidades do aplicativo. O modelo é utilizado para sintetizar os cenários que fazem parte dessa função e quais atores (usuários externos) que irão fazer parte da interação.

Figura 11 – Diagrama de Caso de Uso



Fonte: Elaborada pelo autor.

Tendo como representação a Figura 11, pode-se destacar os cenários e atores que compõem o diagrama.

## 4.2.2 Descrição dos casos de uso

### 4.2.2.1 Manter Receitas

**Meta:** O usuário cadastra bem como consulta, edita e exclui as informações contidas na receita.

**Ator:** usuário e SBD.

#### Cenário Principal

1. O usuário seleciona no menu principal a opção Cadastrar Receita.
2. O usuário insere dados opcionais e obrigatório (vencimento).
3. O usuário salva as informações.
4. O usuário seleciona no menu principal a opção minhas receitas.
5. O SBD lista todos os registros cadastrados.

6. O usuário seleciona item da lista.
7. O usuário altera ou exclui registro.

#### **4.2.2.2 Emitir Notificação**

**Meta:** O sistema verifica se existe uma data de vencimento para alertar o usuário.

**Ator:** sistema.

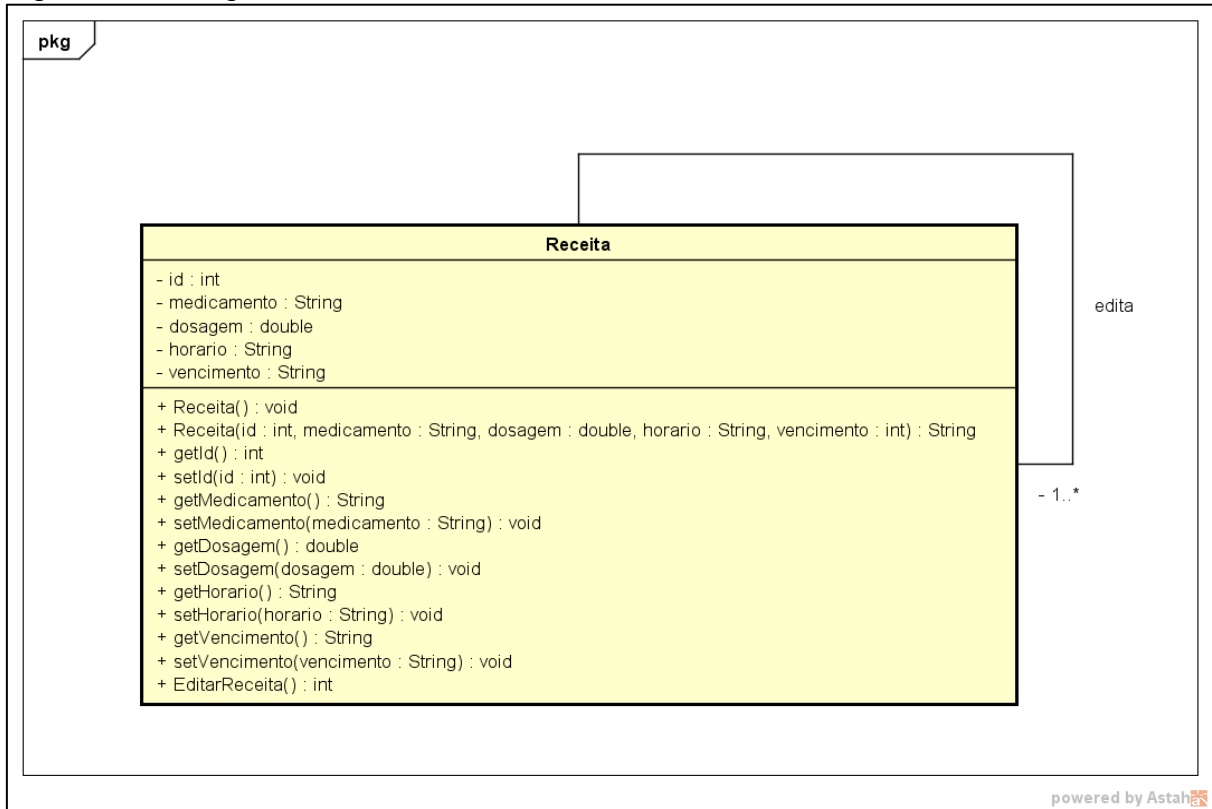
#### **Cenário Emitir Alerta**

1. O sistema emite uma notificação caso exista uma data que esteja em processo de expiração.

### **4.3 Diagrama de classes**

Complementando a modelagem de requisitos, é possível definir as classes analisando os cenários de uso para o sistema a ser construído. Na Figura 12, encontra-se o diagrama de classes com seus atributos e métodos, representando os objetos que o programa irá manipular.

Figura 12 – Diagrama de classes

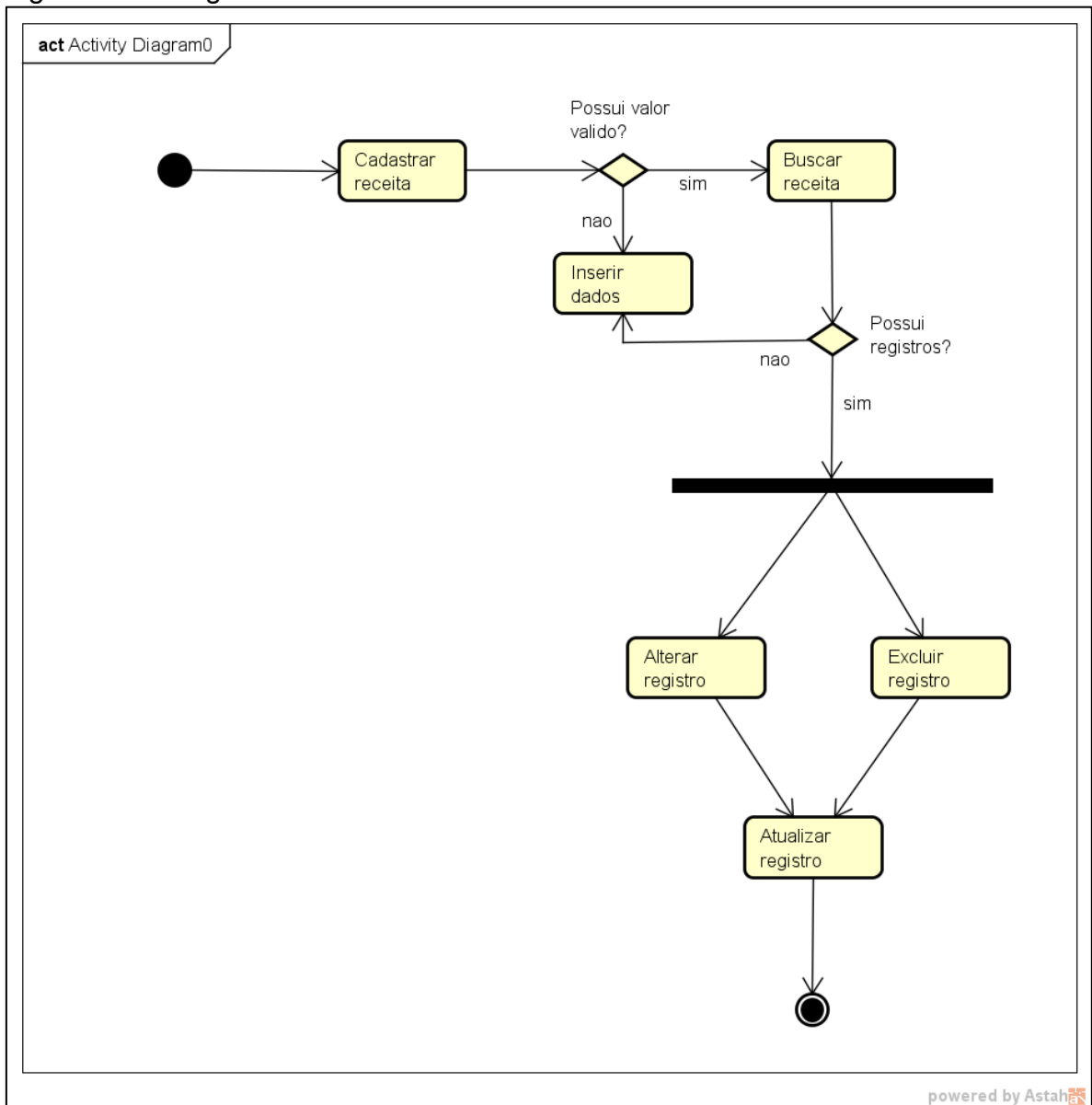


Fonte: Elaborada pelo autor.

### 4.2.3 Diagrama de Atividades

Um diagrama de atividades acrescenta o diagrama de caso de uso incorporando recursos para analisar o comportamento das funções (cenários) e seus resultados em termos das mudanças de estados do objeto. O diagrama está representado na Figura 13.

Figura 13 – Diagrama de Atividades



Fonte: Elaborada pelo autor.

### 4.3 Desenvolvimento

O desenvolvimento deste trabalho se realizou em duas etapas, onde foi realizada a análise e modelagem do mesmo. Nesse primeiro passo, foram identificadas as necessidades de um software para celulares que realizasse o controle do que foi anotado em uma receita bem como o nome do medicamento e outras informações incorporadas contendo dosagem, intervalo de uso e data de vencimento da receita, tendo como foco os medicamentos prescritos.

Optou-se pela escolha da IDE Eclipse pois, além de conter uma interface de fácil uso, possui recursos internos que facilitam a escrita de código em linguagens de programação orientada a objetos. De forma geral, para o desenvolvimento do aplicativo se aplicou o seguinte conjunto de ferramentas:

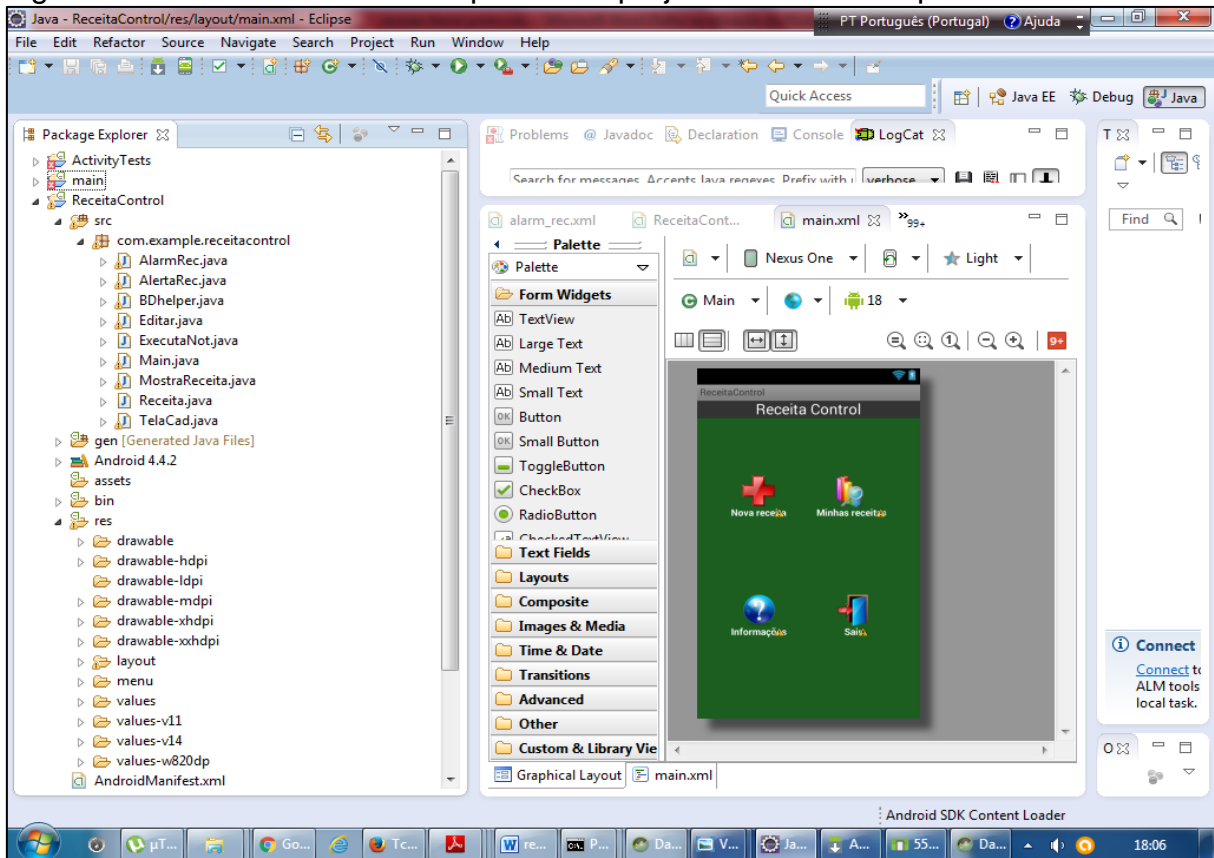
- a) Intel Core I3, 3 GB RAM, Windows 7;
- b) Java Software Development Kit;
- c) Ambiente de Desenvolvimento Integrado Eclipse Juno;
- d) Android Standard Development Kit;
- e) Android Virtual Device;
- f) Smartphone Samsung Galaxy2Duo, com sistema operacional Android 4.4.2;

Os requisitos especificam que o sistema deve ser capaz de rodar em dispositivos móveis com sistema operacional Android. A codificação do aplicativo deve ser baseada nos modelos desenvolvidos nas fases de análise de requisitos, análise e design, mais precisamente nas especificações prévias, diagrama de casos de uso, de classes e de atividades.

#### **4.3.1 Estrutura do projeto**

Neste momento, será descrita a estrutura do projeto Receita Control. Observem que dentro da pasta *ReceitaControl* existe uma pasta chamada *src* e dentro dela é que ficam os códigos fonte Java das aplicações. Observem que o arquivo *Main.java* se encontra dentro do pacote *com.example.receitacontrol* (Esse pacote também é uma pasta). Esse arquivo é uma das interfaces pertencentes a nossa aplicação Android.

Figura 14 – Tela da estrutura de pastas do projeto utilizando eclipse



Fonte: Elaborada pelo autor.

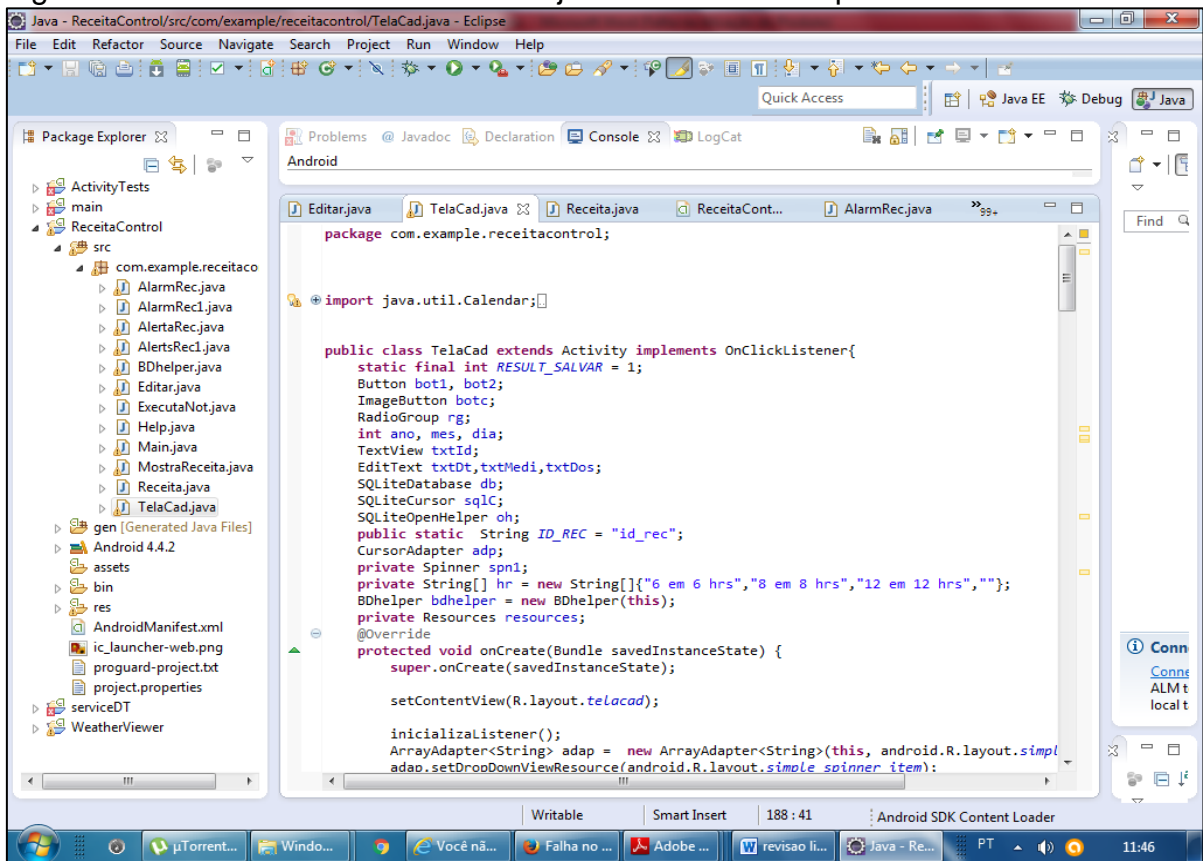
A descrição completa de todas as pastas do projeto é a seguinte:

- a) **Src**: Contém todos os códigos .Java.
- b) **Biblioteca Android 4.4.2** - Inclui o arquivo android.jar, que por sua vez contém todas as bibliotecas necessárias para rodar o app.
- c) **Gen** – Agrega o arquivo R.java, responsável por mapear todos os recursos encontrados no projeto, é gerado automaticamente não podendo ser modificado.
- d) **Res** – Contém os mecanismos responsáveis por mapear todos os recursos gráficos encontrados no projeto, na subpasta layout se encontra a interface gráfica, em drawable se encontram as imagens utilizadas.
- e) **AndroidManifest.xml** – Arquivo responsável por registrar todas as *activities* (atividades), pois contém as permissões e classificações.

As classes .java é onde se encontra o código fonte das atividades, nesse caso, podemos considerar uma *Activity* como sendo responsável pela construção

da interface da aplicação. Na figura 15 se encontra a *activity* de cadastro exibindo a classe `TelaCad.java`, necessária para desenhar a tela de cadastro:

Figura 15 – Tela da atividade `TelaCad.java` retirada do eclipse



```

package com.example.receitacontrol;

import java.util.Calendar;

public class TelaCad extends Activity implements OnClickListener{
    static final int RESULT_SALVAR = 1;
    Button bot1, bot2;
    ImageButton botc;
    RadioGroup rg;
    int ano, mes, dia;
    TextView txtId;
    EditText txtDt,txtMedi,txtDos;
    SQLiteDatabase db;
    SQLiteCursor sqlC;
    SQLiteOpenHelper oh;
    public static String ID_REC = "id_rec";
    CursorAdapter adp;
    private Spinner spn1;
    private String[] hr = new String[]{"6 em 6 hrs","8 em 8 hrs","12 em 12 hrs",""};
    BDhelper bdhelper = new BDhelper(this);
    private Resources resources;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.telacad);

        inicializaListener();
        ArrayAdapter<String> adap = new ArrayAdapter<String>(this, android.R.layout.simple_spinner_item);
        adap.setDropDownViewResource(android.R.layout.simple_spinner_item);
    }
}

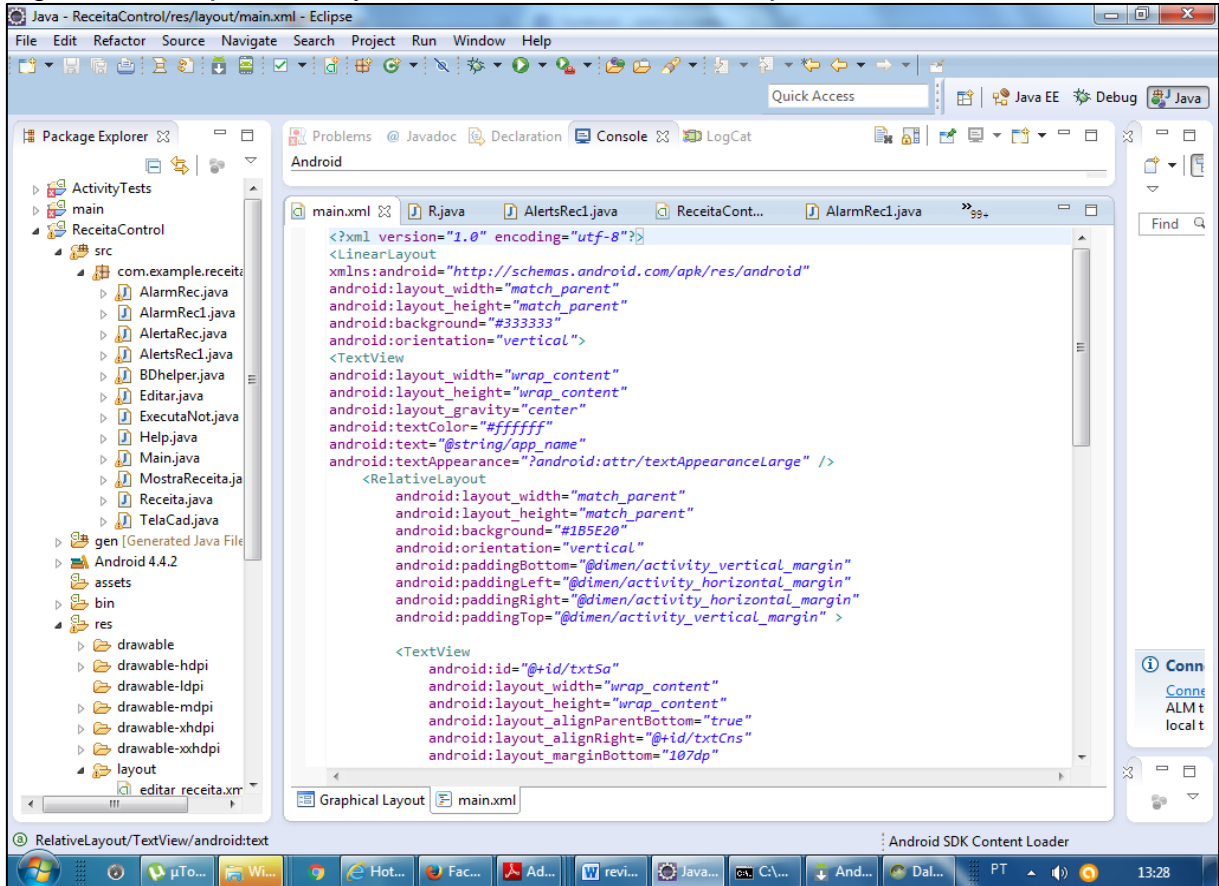
```

Fonte: Elaborada pelo autor.



Os arquivos .xml da subpasta layout da pasta res define como vai ser a interface gráfica contendo todos os componentes em formato xml.

Figura 16– Arquivo de layout main.xml retirado do eclipse



Fonte: Elaborada pelo autor.

### 4.3.2 Cadastro

Ao tocar no ícone para adicionar nova receita, o aplicativo exibirá a tela de cadastro, a qual exibirá os campos em branco para se poder acrescentar dados. As informações desta tela são identificadas por medicamento, dosagem, intervalo de uso e vencimento.

Nessa tela, poderão ser executadas as seguintes tarefas:

- Cadastrar: Para incluir um novo registro deverá ser pressionado o botão Salvar, se todos os campos estiverem preenchidos o sistema operacional exibirá uma mensagem de confirmação de cadastro, caso contrário se o campo data estiver em branco não será possível concluir o cadastro por tratar-se de campo obrigatório para poder se adequar ao objetivo principal

do app, que é emitir uma notificação na barra de status. A não inserção de dados nesse campo implica em um alerta do próprio componente.

- b) Voltar: Para retornar a tela principal deverá ser pressionado o botão Voltar, onde a mesma será fechada. A Figura 17 demonstra a respectiva tela de cadastro.

Figura 17 – Tela de cadastro

A imagem mostra a tela de cadastro de um medicamento em um aplicativo móvel. No topo, há uma barra de status com o tempo 16:31 e ícones de rede e bateria. O formulário contém os seguintes elementos:

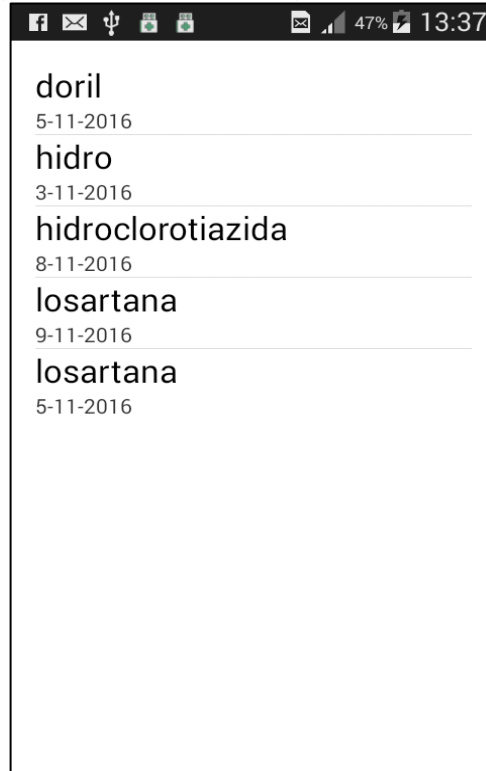
- Medicamento: Campo de texto vazio.
- Dosagem: Campo de texto vazio com duas opções de unidade: 'cmp' (selecionada com um botão verde) e 'ml' (desselecionada com um botão cinza).
- Horário: Menu suspenso com a opção '12 em 12 hrs' selecionada.
- Vencimento: Campo de texto vazio com um ícone de alerta vermelho no canto superior direito. Uma caixa de diálogo vermelha com o texto 'Campo obrigatório' está sobreposta ao campo.
- Botões: 'Salvar' (desabilitado) e 'Voltar' (desabilitado).

Fonte: Elaborada pelo autor.

### 4.3.3 Minhas receitas

A tela minhas receitas retorna uma lista de registros com todos os medicamentos cadastrados, além do vencimento da receita para o mesmo. Os dados serão classificados por ordem alfabética de medicamento. Após a seleção de algum item da lista, é exibida uma tela com todas as informações referentes àquela data, sendo útil para uma posterior consulta a informações do formulário.

Figura 18 – Lista de registros



doril	5-11-2016
hidro	3-11-2016
hidroclorotiazida	8-11-2016
losartana	9-11-2016
losartana	5-11-2016

Fonte: Elaborada pelo autor.

#### 4.3.4 Editar Receita

A tela de edição retorna as informações do banco de dados para consulta. Os dados da lista serão classificados por ordem alfabética de medicamento. Nessa tela também é possível excluir registro; as funcionalidades dos botões podem ser definidas como:

- Alterar: Ao pressionar Alterar, é realizado um *update* no banco ao qual se houver alteração de dados o conteúdo é atualizado;
- Excluir: Ao pressionar Excluir, é realizado um *delete* para aquele registro no banco de dados, apagando suas informações.

Figura 19 – Tela de edição

The screenshot shows a mobile application interface for editing medication data. At the top, there is a status bar with icons for social media, a USB connection, and battery level (35%) at 12:54. The main form contains the following elements:

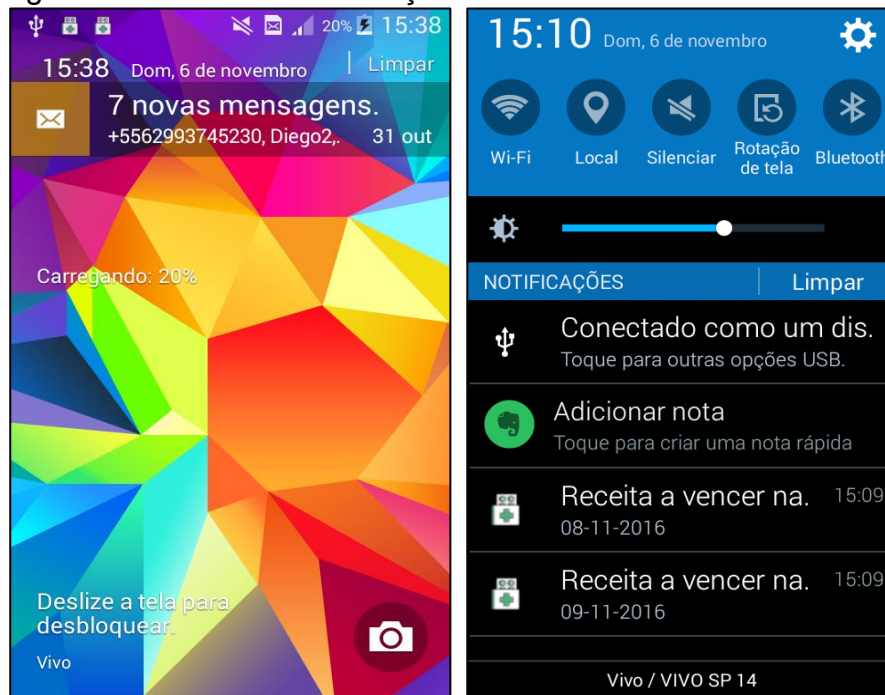
- Medicamento:** A text input field containing "losartana", highlighted with an orange border.
- Dosagem:** A text input field containing "1.5".
- Units:** Two radio buttons labeled "cmp" (selected) and "ml".
- Horario:** A dropdown menu showing "8 em 8 hrs".
- Vencimento:** A date input field containing "9-11-2016" and a calendar icon showing the number "31".
- Buttons:** Two buttons at the bottom: "Alterar" and "Excluir".

Fonte: Elaborada pelo autor.

#### 4.3.5 Notificações

As notificações ficam fixas na barra de status no momento em que o usuário ligar o celular, levando ao aplicativo caso toque na mensagem. Se for excluída, será destruída e não irá aparecer novamente. Para entrar no processo de notificação, o sistema deverá completar o ciclo de inicialização e a data atual deve estar subtraída de três dias de seu vencimento, estipulando um tempo para que o usuário possa providenciar outra receita. A partir da identificação da data, a notificação ficara na barra de status com alerta para o vencimento subtraído de três e dois dias efetuando comparação da data anterior seguida da atual, como mostra a Figura 20:

Figura 20 – Telas de notificações



Fonte: Elaborada pelo autor.

## **5 RESULTADOS E CONSIDERAÇÕES FINAIS**

Esta pesquisa teve como objetivo desenvolver um aplicativo com interface amigável que, com base em uma data, alerte o usuário, para que ele mantenha um controle sobre a receita médica.

Atualmente, muitos aplicativos vêm sendo desenvolvidos para atender a demanda do setor medicamentoso brasileiro, visando facilitar a vida de pessoas que por algum motivo fazem uso de medicamentos prescritos, e também atender a demanda da comunicação móvel Android. Esses aplicativos ajudam o paciente a obter maior controle sobre alguns elementos, como, por exemplo: horário estipulado para se consumir um determinado medicamento; dosagem necessária; telefone do médico ou clínica responsável pela receita.

No decorrer da construção do app, foi desenvolvida a aplicação do banco de dados SQLite e a utilização dos mecanismos Android necessários para emitir notificação em segundo plano (barra de status). Foi implementada a codificação de acordo com os conceitos pesquisados e estudados na revisão bibliográfica como: engenharia de software; banco de dados; programação Java.

Conforme o teste realizado no aparelho Samsung operando com a plataforma Android 4.2.2, concluiu - se que o aplicativo obteve uma boa performance tanto no processamento das informações como na usabilidade de interface.

O aplicativo desenvolvido neste trabalho mostra que é possível a utilização de um software para auxiliar os pacientes a administrar suas receitas médicas bem como alertá-lo quando em seu vencimento utilizando um celular com sistema Android

### **5.1 TRABALHOS FUTUROS**

Por fim, este projeto permanece em aberto para implantações de funcionalidades que no momento não são possíveis ser inclusas, deixando-o mais completo. Por exemplo:

- a) Implementar uma tela, com recursos para incluir mais medicamentos a uma determinada receita.

- b) Implementar uma funcionalidade para permitir ao paciente escolher entre exibir e não exibir notificações.
- c) Implementar uma funcionalidade de integração com o Google Maps para a possibilidade de localizar consultórios médicos e farmácias próximos.

## REFERÊNCIAS

- ALMEIDA, M. **Seção: telefonia celular**. Disponível em: <<http://www.teleco.com.br/tutoriais/tutorialintlte/default.asp>>. Acesso em: 2/5/2016.
- AQUINO, D. S. **Por que o uso racional de medicamentos deve ser uma prioridade?** Departamento de farmácia Faculdades Integradas da Vitoria de Santo Antão, Recife, 2007.
- BULASMED. **Referências completas de medicamentos**. Disponível em: <<http://www.bulas.med.br/>>. Acesso em: 18/09/2016.
- CANALYS. **Smart mobile device shipments exceed 300 million in Q1 2013**. Disponível em: <[http://www.canalys.com/static/press\\_release/2013/canalys-press-release-090513-smart-mobile-device-shipments-exceed-300-million-q1-2013\\_0.pdf](http://www.canalys.com/static/press_release/2013/canalys-press-release-090513-smart-mobile-device-shipments-exceed-300-million-q1-2013_0.pdf)>. Acesso em: 29/6/2016.
- CARVALHO, I. **Mundo terá mais celular do que pessoas**. 2012. Disponível em: <<http://www.ismaelcarvalho.com.br/2012/11/22/mundo-tera-mais-celular-do-que-pessoas-em-2013/>>. Acesso em: 22/4/2016.
- CARVALHO, J. O. F.; **O Papel da interação humano computador na inclusão digital**. Programa pós graduação em ciências da informação. PUC. 2003. Disponível em: <<http://periodicos.puc-campinas.edu.br/seer/index.php/transinfo/article/view/1461/1435>>. Acesso em: 12/5/2016.
- CRUZ, D. L. e NACIF, M. **Elaboração de aplicativo para avaliação nutricional para celulares com sistema Android**. Disponível em: <<http://www.e-publicacoes.uerj.br/index.php/demetra/article/view/15555/14517#.VxfoUHoUMWE>>. Acesso em: 22/4/2016.
- DATE, C. J. **Introdução a sistemas de banco de dados**. 9ª ed., Rio de Janeiro. 2003.
- DEITEL, P. **Android para programadores uma abordagem baseada em aplicativos**. Porto Alegre: Bookman, 2013.
- ETELG. **UML – Linguagem de Modelagem Unificada**. Disponível em: <[http://www.etelg.com.br/paginaete/downloads/informatica/apostila\\_uml.pdf](http://www.etelg.com.br/paginaete/downloads/informatica/apostila_uml.pdf)>. Acesso em: 14/07/2016.
- GIL, A. C. **Como Elaborar Projetos de Pesquisa**. 5ª ed., São Paulo: Atlas, 2010.
- HAMMERSCHMIDT, C.; SPINILLO, C. G. **Informações sobre medicamentos em dispositivos móveis: Análise da articulação tipográfica no aplicativo MedSUS**. Disponível em: <<http://www.proceedings.blucher.com.br/article-details/informaes->



sobre-medicamentos-em-dispositivos-mveis-anlise-da-articulao-tipografica-no-aplicativo-medsus-20257>. Acesso em: 5/6/2016.

KORTH, H. F.; SILBERSCHATZ, A. **Sistemas de Bancos de Dados**. São Paulo: Makron Books, 2ª ed., 1994.

KREIBICH, Jay A. **Using SQLite**. Sebastopol - EUA: O'Reilly Media, 2010. 528 p.

LECHETA, R. R. **Google Android: Aprenda a criar aplicações para dispositivos móveis com o android SDK**. 6ª ed., São Paulo: Novatec, 2013.

MADRUGA, C.M.C. e SOUZA, E.S.M. **Manual de orientações básicas para prescrição médica**. Universidade Federal da Paraíba. UFPB. 2012.

PACIEVITCH, YURI. **Banco de Dados**, 2016. Disponível em: <<http://www.infoescola.com/informatica/banco-de-dados/>>. Acesso em: 21/6/2016.

PRESSMAN, Roger S. **Engenharia de Software**. 7ª ed., Porto Alegre: AMGH, 2011.

SCHILD, H.; SKRIEN, D. **Programação com Java uma introdução abrangente**. Porto Alegre: AMGH, 2013.

TAURION, Cezar. **Internet Móvel Tecnologias, Aplicações e Modelos**. Rio de Janeiro. Editora: Campus, 2002.

TEIXEIRA, Mariane Mendes. **Touch screen**. Disponível em: <<http://brasilecola.uol.com.br/fisica/touch-screen.htm>>. Acesso em: 4/11/2016.

TI SELVAGEM. **Desenvolvendo para Android: Arquitetura Android**. <Disponível em: <<http://www.tiselvagem.com.br/geral/desenvolvendo-para-android-arquitetura-android/>>. Acesso em: 20/5/2016.