

CENTRO UNIVERSITÁRIO SERRA DOS ÓRGÃOS – UNIFESO
CENTRO DE CIÊNCIAS E TECNOLOGIAS
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

**APLICATIVO DE AUXÍLIO NO CONTROLE DA DIABETES TIPO I
VOLTADO PARA CRIANÇAS E ADOLESCENTES**

Bruno Pinela Marques

Teresópolis/RJ
2014

CENTRO UNIVERSITÁRIO SERRA DOS ÓRGÃOS – UNIFESO
CENTRO DE CIÊNCIAS E TECNOLOGIAS
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

**APLICATIVO DE AUXÍLIO NO CONTROLE DA DIABETES TIPO I
VOLTADO PARA CRIANÇAS E ADOLESCENTES**

Bruno Pinela Marques

Trabalho de Conclusão de Curso apresentado ao

Centro Universitário Serra dos Órgãos como
requisito para obtenção de título de bacharel em

Ciência da Computação.

Orientador: José Carlos Tavares da Silva

Teresópolis/RJ
2014

Palavras são, na minha nada humilde opinião, nossa
inesgotável fonte de magia. Capazes de formar
grandes sofrimentos e também de remedia-los.

Albus Percival Wulfric Brian Dumbledore

RESUMO

Este trabalho teve como objetivo realizar estudos sobre a utilização da mobilidade no auxílio ao controle da diabetes mellitus. Com base nos estudos realizados, pode-se observar que há uma necessidade de manter os usuários portadores de diabetes, assim como seus respectivos responsáveis bem informados e oferecer-lhes uma ferramenta de uso dinâmico e que possa estar sempre ao seu lado. A partir da necessidade identificada foi desenvolvida uma aplicação para dispositivos móveis Android com o intuito de auxiliar crianças e adolescentes portadores desta doença com suas marcações de glicemia diárias. A vantagem da ferramenta é permitir que informações sejam fornecidas não só para os usuários mas também para seus médicos que com base em gráficos e históricos mais detalhados podem dar um melhor diagnóstico para os seus pacientes.

Palavras-Chave : Diabetes mellitus, Aplicativo Android, Uso da mobilidade na área da saúde

Abstract

The current work aimed to conduct studies on the use of mobility aid in the control of diabetes mellitus . Based on the studies conducted, it can be seen that there is a need to keep the users of diabetes patients and their guardians informed and offer them a dynamic tool use and you can always be at your side . From the identified need has developed an application for Android mobile devices with the intent of helping children and adolescents with this disorder with their daily blood glucose markings . The advantage of the tool is to allow information to be provided not only for users but also to their doctors based on more detailed graphics and history can give a better diagnosis for their patients .

Keywords : Diabetes mellitus, Android App , Use of mobility in healthcare

Sumário

Lista de Figuras	7
Lista de Siglas.....	8
1. Introdução	9
1.1. Objetivos e contribuições.....	10
1.2. Organização do trabalho	11
2. Fundamentação Teórica	12
2.1. Conceito básico sobre o diabetes.....	12
2.1.1. Diabetes Tipo1	12
2.2. Mobilidade	13
2.2.1. Smartphones	14
2.2.2. Uso da mobilidade na saúde	14
2.3. Android ®	15
3. Metodologia.....	25
3.1. Ambiente de Desenvolvimento	25
3.2. Ferramentas utilizadas	26
3.2.1. Android Studio	26
3.2.2. GenyMotion	27
3.2.3. SQLite.....	28
4. Desenvolvimento.....	32
4.1. Sistema Mobile.....	32
4.2. Detalhamento do desenvolvimento	33
5. Resultados.....	42
6. Considerações Finais	46
6.1. Conclusões	46
6.2. Trabalhos Futuros.....	46
7. Referências Bibliográficas.....	47

Lista de Figuras

Figura 1: Gráfico de crescimento da utilização do Android entre 2009 e 2011

Figura 2: Diagrama da arquitetura Android®

Figura 3: Componentes de uma aplicação Android Fonte

Figura 4: Ciclo de vida de uma *activity*

Figura 5: Hierarquia dos *widgets* no Android®

Figura 6: Tela Principal do aplicativo Diabetes DevFS emulado no software GenyMotion

Figura 7: Diagrama Model-View-Controller

Figura 8 : Utilização de versões do Android®

Figura 9: Tela de carregamento ao abrir o aplicativo Diabetes DevFS emulado no software GenyMotion

Figura 10: Tela principal do aplicativo Diabetes DevFS emulado no software GenyMotion

Figura 11: Tela de marcação de glicemia do aplicativo Diabetes DevFS emulado no software GenyMotion

Figura 12: Tela de Avaliação do aplicativo Diabetes DevFS emulado no software GenyMotion

Figura 13: Tela de Visualização de dados do aplicativo Diabetes DevFS emulado no software GenyMotion

Figura 14: Tela do Relatório exportado para .pdf do aplicativo Diabetes DevFS emulado no software GenyMotion

Figura 15: Tela Medido de insulina do aplicativo no app Diabetes DevFS emulado no software GenyMotion

Figura 16: Tela de configurações do app Diabetes DevFS emulado no software GenyMotion

Figura 17: Relatório de *downloads* e instalações atuais do aplicativo.

Figura 18: Instalações e desinstalações diárias

Figura 19: Sexo dos usuários

Figura 20: Gráfico da versão do Android® mais utilizada no aplicativo

Lista de Siglas

API - *Application Programming Interface*

AVD - *Android Virtual Device*

DM - Diabetes Mellitus

GAD - Antidescarboxilase do ácido glutâmico

GPS - *Global Positioning System*

IA2 - Antitirosina-fosfatase

IAA - Auto anticorpos anti-insulina

IBGE - Instituto Brasileiro de Geografia e Estatística

ICA512 - Anti-ilhota pancreática

IDE - *Integrated Development Environment*

IMC - Índice de Massa Corporal

HLA - Histocompatibilidade humano

HTTP - *Hypertext Transfer Protocol*

MVC - *Model-View-Controller*

RMS - *Record Management System*

SDK - *Software Development Kit*

SO - Sistema Operacional

SQL - *Structured Query Language*

XML - *Extensible Markup Language*

1. Introdução

Segundo o estudo realizado pelo IBGE (2011) sobre acesso à Internet e posse de telefone móvel celular para uso pessoal, estimou que (20,7%) da população brasileira que possui telefone celular para uso pessoal tem entre 10 e 19 anos de idade. Destaca-se também que entre 2005 e 2011, enquanto a população de 10 anos ou mais de idade cresceu 7,4%, o contingente daqueles que possuíam celulares teve aumento de 54,9% (DADOS IBGE, SENSO 2011).

Sendo assim, os celulares apesar de terem poder computacional diferenciado de um computador normal, a cada momento possuem uma nova forma, tamanho, aumento na capacidade de processamento, bem como novos aplicativos agregados com as mais diversas finalidades, e com o desenvolvimento desses aplicativos para celulares vem tornando-se cada vez mais frequente empresas especializadas em produção desse tipo de *software* (ROMEIRO, 2005).

Quando foram lançados, os celulares eram grandes e apenas pessoas com boas condições financeiras podiam possuir um. Com o tempo eles evoluíram, ficaram cada vez menores e mais populares. Assim surgiram os celulares inteligentes, *smartphones*, aparelhos que possuem múltiplas tarefas e funções, diferentes de um simples celular utilizado a alguns anos atrás.

Em paralelo a essa realidade, o crescimento da tecnologia móvel relacionada à *smartphones*, *tablets* dentre outros dispositivos vem ganhando cada vez mais importância. E com o objetivo de suprir a demanda deste mercado, a empresa Google® desenvolveu um sistema operacional denominado Android® baseado em licenciamento de código aberto, o que possibilita a contribuição de qualquer desenvolvedor criar, personalizar ou até mesmo corrigir erros do sistema. O sistema operacional Android® é baseado no *Kernel 2.6* do Linux e é o responsável por gerenciar a memória, os processos, os *threads* e a segurança do sistema no qual o mesmo está operando (SILVA *et al.*, 2013).

Também é possível mencionar que as aplicações executadas em um celular podem ter um comportamento de funcionamento *offline* ou *online*, sincronizando informações diretamente de um servidor confiável (LECHETA, 2013).

O Android® possui muitos diferenciais interessantes e uma arquitetura flexível focada na integração de aplicações, não existindo diferença entre uma aplicação nativa e uma construída pelo desenvolvedor (LECHETA, 2013).

Muitos avanços tecnológicos ocorreram, inclusive na área da saúde, com a introdução da informática e do surgimento de aparelhos modernos e sofisticados que trouxeram muitos benefícios e rapidez no controle e luta contra as doenças trazendo desta forma uma melhor qualidade de vida, bem estar e saúde que são termos cada vez mais presentes no nosso cotidiano e agora poderão receber auxílio desses avanços tecnológicos. (BARRA *et al.*, 2006).

1.1. Objetivos e contribuições

O principal objetivo deste trabalho é promover uma ferramenta que preste auxílio ao monitorar e controlar sua saúde, através do registro de seus níveis de glicemia e sua alimentação para a obtenção de uma melhor qualidade de vida.

A partir da obtenção destes registros, os dados colocados no aplicativo são armazenados e outros recursos são ativados como a geração de gráficos, histogramas, o cálculo da contagem de carboidratos, dentre outros, assim substituindo os processos anteriormente feitos de forma manual, processos estes cansativos, visto que são feitos diversas vezes ao dia, propiciando o esquecimento ou até mesmo a perda dessas informações, uma vez que todas as anotações precisam ser guardadas e feitas de modo organizado afim de entendê-las.

Por isso, faz-se necessária a construção de uma aplicação móvel, que utilizando as mais recentes tecnologias de desenvolvimento para a plataforma Android, auxilie as crianças bem como seus respectivos responsáveis no acompanhamento da sua doença.

1.2. Organização do trabalho

A organização do presente trabalho encontra-se da seguinte forma : no capítulo 2 são apresentadas as bases teóricas para este trabalho, bem como os conceitos básicos sobre a diabetes com enfoque maior na diabetes tipo 1, universo de estudo deste trabalho; no capítulo 3 é descrito a metodologia utilizada nesse trabalho; no capítulo 4 é mostrado todo o processo de desenvolvimento do aplicativo; o capítulo 5 é feito a amostragem dos resultados e por fim o capítulo 6 apresenta as considerações finais que conta com as conclusões além de algumas propostas de trabalhos futuros relacionados.

2. Fundamentação Teórica

Neste capítulo é apresentada a fundamentação teórica necessária para a melhor compreensão dos principais conceitos utilizados na elaboração dessa monografia. Sua estruturação foi definida com base nas fases de desenvolvimento deste trabalho.

2.1. Conceito básico sobre o diabetes

Diabetes mellitus refere-se a um grupo de distúrbios metabólicos, que compartilham o fenótipo de hiperglicemia. Existem vários tipos distintos de diabetes mellitus, causados por uma interação complexa de fatores genéticos e ambientais, sendo classificados de acordo com o processo patogênico que resulta em hiperglicemia. Em 1985, estimava-se haver 30 milhões de adultos com DM no mundo; esse número cresceu para 173 milhões em 2002 e atualmente existem 250 milhões de casos. Cerca de dois terços desses indivíduos com DM vivem em países em desenvolvimento, onde a epidemia tem maior intensidade, sendo o diabetes mellitus tipo1, universo de estudo do presente trabalho, responsável por cerca de 5% a 10% dos casos.

2.1.1. Diabetes Tipo1

O diabetes mellitus tipo1 que será focalizado nessa Monografia, é uma doença autoimune decorrente da destruição das células beta pancreáticas resultando em uma deficiência absoluta de insulina ou ainda, na ausência de evidências de processo autoimune, referida como idiopática (GROSS *et al.*, 2002).

Ela afeta indivíduos geneticamente suscetíveis, os quais apresentam alguns marcadores de auto imunidade como os auto anticorpos anti-insulina (IAA), antidescarboxilase do ácido glutâmico (GAD), antitirosina-fosfatase (IA2) e anti-ilhota pancreática (ICA512) que podem estar presentes meses ou anos antes do diagnóstico clínico, ou seja, na fase pré-clínica da doença. Esses marcadores são identificados em até 90% dos indivíduos quando se detecta hiperglicemia. Mas, a despeito da possibilidade de serem encontrados em qualquer faixa etária,

são mais comuns na infância e adolescência. Os genes do sistema de histocompatibilidade humano (HLA), sobretudo os de classe II DR3, DR4 e DQ, responsáveis por 40% do componente genético desta doença crônica. (Sociedade Brasileira de Dermatologia, 2012). As manifestações clínicas do diabetes mellitus tipo 1, no entanto, só se tornam evidentes tardiamente quando já ocorreu o comprometimento de cerca de 80% da massa das ilhotas pancreáticas (TSCHIEDEL, 2004).

Essa doença afeta predominantemente crianças e adolescentes, com o pico de início entre 11 e 12 anos de idade (SESTERHEIM *et al.*, 2007). Esse fato é corroborado pela taxa de prevalência elevada já que o diabetes mellitus tipo 1 é a segunda doença crônica mais comum nessa faixa etária, sendo superada apenas pela asma (BORUS; LAFFEL, 2010). Ainda assim, diversos estudos recentes apontam para uma tendência mundial de aumento na incidência em menores de 5 anos de idade. Em relação à incidência, o diabetes mellitus tipo 1 tem grande variabilidade, sendo a taxa bastante elevada em alguns países nórdicos como a Finlândia (40 casos/100.000 nascimentos/ano) ao passo que países asiáticos apresentam taxas significativamente ¼ menores (inferiores a 3/100.000 nascimentos/ano (SPOLLETT, 2009). E a nível mundial ele representa de 5 a 10% de todos os casos de diabetes, já no Brasil, são escassos os estudos que avaliaram a incidência do diabetes mellitus tipo 1 especificamente.

2.2. Mobilidade

O conceito de mobilidade consiste no uso de dispositivos móveis portáteis fornecendo serviços para que usuários se conectem, obtenham dados e os forneçam a outros usuários e sistemas. Os dispositivos móveis podem ser definidos como equipamentos portáteis que é possível carregar facilmente e que se comunica com outros dispositivos através de uma tecnologia de comunicação sem fio. Exemplos de dispositivos móveis são os *laptops*, *smartphones*, *tablets*, dentre outros. Mesmo esses dispositivos possuindo funcionalidades capazes de fornecer serviços em qualquer hora e lugar, ainda possuem restrições como: processamento, memória, duração da bateria limitada e *interfaces* de entrada de dados limitadas (LEE 2005).

O intuito da mobilidade é a simplicidade do uso e o acesso rápido às informações. Com isso, vem crescendo o uso de *smartphones* e *tablets*.

2.2.1. Smartphones

Nos últimos 10 anos, eram poucas pessoas que utilizavam celulares. Entretanto os avanços tecnológicos propiciaram uma maior acessibilidade e assim, tornou-se um recurso importante para as pessoas (OLIVEIRA *et al.*, 2011).

Os primeiros celulares comerciais foram lançados na década de 80, não sendo nada portáteis, com custo elevado e que possibilitavam apenas realizar ligações. No decorrer dos anos, os celulares além de diminuir de tamanho acabaram agregando novas funções. Atualmente, os celulares realizam uma grande variedade de funções, sendo assim caracterizados como *smartphones* (celulares inteligentes). Estes possuem um SO (Sistema Operacional) para gerenciamento de todos componentes do dispositivo móvel. (DIAS *et al.*, 2011).

Atualmente, ao redor do mundo aproximadamente 1 bilhão de pessoas utilizam a Internet e quase 3 bilhões de pessoas possuem um telefone celular, sendo este um dos produtos de consumo mais bem sucedidos. (OHA *et al.*, 2014)

Um dos fatores que propiciaram este aumento da utilização da Internet pode estar associado ao mercado de smartphone, que cresceu 110% no segundo trimestre de 2013 em comparação ao mesmo período de 2012 (EXAME, 2013).

2.2.2. Uso da mobilidade na saúde

É comum considerar a ciência e tecnologia como facilitadores para o progresso, que proporcionam não só o desenvolvimento do saber humano como também uma evolução para o mesmo. Desta forma, subentende-se que ambas trarão benefícios à humanidade (PINHEIRO *et al.*, 2007).

A emergente aplicação da informática na monitoração da saúde tem proporcionado o surgimento de novas tecnologias e aplicações, assim produzindo muitas e variadas soluções para as necessidades do profissional da saúde armazenar e recuperar informações de pacientes (SIGULEM; SALOMÃO, 2004).

A tecnologia deve propiciar a integridade, disponibilidade e confidencialidade das informações, isso faz com que se substitua o papel de celulose por um “papel virtual”, em que tudo pode ser armazenado de maneira segura e de fácil organização (BREGA *et al.*, 2011).

Dentre suas diversas oportunidades, a tecnologia permite a população e os profissionais da saúde, o acesso às informações médicas, melhorando a eficiência dos cuidados a saúde em tempo útil. Com o surgimento dos sistemas móveis e a adoção generalizada dos *smartphones*, as aplicações móveis são de grande utilidade e estão tendo uma rápida expansão (KATZ *et al.*, 2005).

2.3. Android ®

O Android ® é, sob diversos ângulos, uma antítese do iPhone ®. Enquanto a Apple® optou por manter um controle restrito sob sua plataforma, impondo restrições aos desenvolvedores e controlando a distribuição dos aplicativos, o Google® optou por seguir o caminho oposto, criando um sistema aberto e incentivando a criação de aplicativos para a plataforma, inclusive com prêmios em dinheiro. Assim como nos *desktops*, as plataformas de smartphones estão se consolidando e o mais importante passou a serem os aplicativos e não apenas o *hardware* ou as funções básicas do sistema.



Figura 1 – Gráfico de crescimento da utilização do Android entre 2009 e 2011

Fonte – Android Developers (2014)

Observando a Figura 1, é possível concluir que o Android® segue crescendo exponencialmente em número de utilização do seu SO em *smartphones*. E todo esse crescimento deve-se ao investimento e consolidação de uma plataforma muito boa e que oferece a seus desenvolvedores ferramentas gratuitas para desenvolvimento de aplicativos trazendo assim não apenas desenvolvedores mas também usuários que se identificam com softwares que atendem suas necessidades.

Lançado em 2008, está disponível em 190 países ao redor do mundo e em centenas de milhões de dispositivos móveis, oferecendo um conjunto completo de *softwares* para dispositivos móveis (ANDROID, 2014).

O Android® foi construído sobre o *Kernel*¹ do Linux. Além disso, utiliza uma máquina virtual personalizada, Dalvik², que foi projetada para otimizar a memória e recursos de *hardware* em um ambiente móvel. Sua arquitetura é dividida em cinco camadas: *Kernel*, Bibliotecas, *Runtime*³, *Framework* de Aplicação e Aplicativos, conforme apresentado na Figura 2. Essas camadas têm suas características próprias e juntas dão sentido ao montante das aplicações para execução.

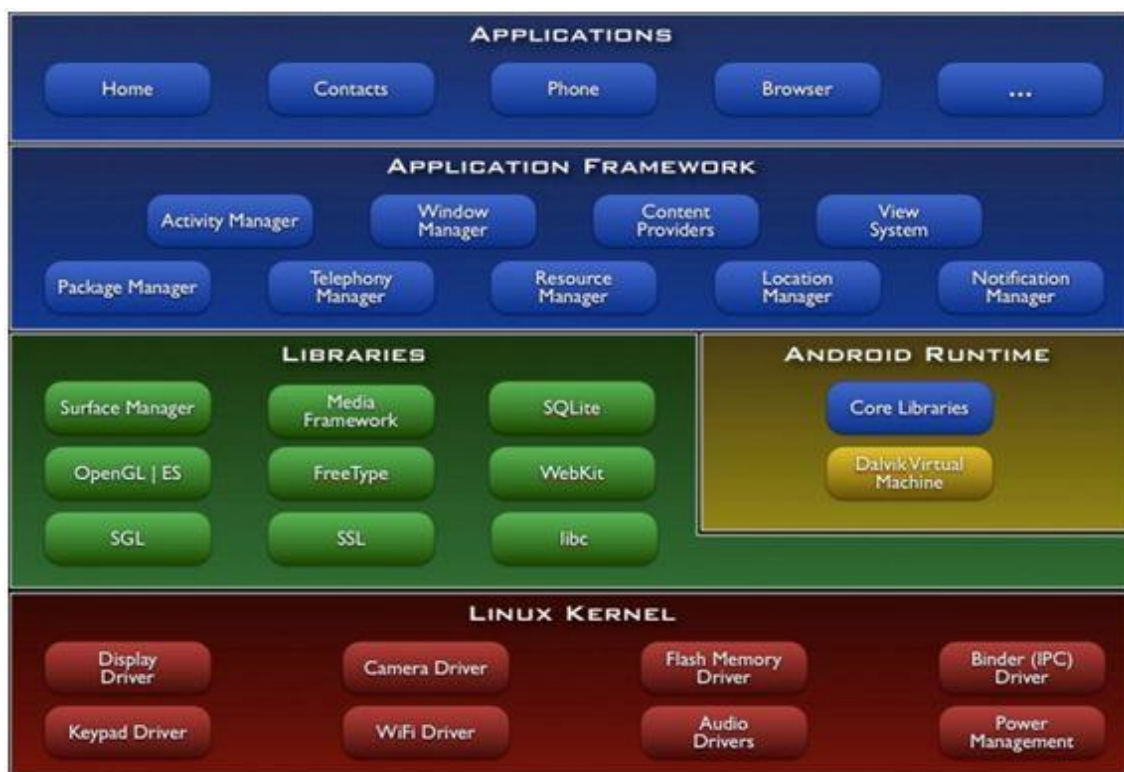


Figura 2 - Diagrama da arquitetura Android®
Fonte – LECHETA (2013)

- 1 É o núcleo do S.O., componente que faz a interação do *hardware* com o *software*.
- 2 É uma máquina virtual otimizada para execução em dispositivos móveis, converte *bytecode* compilado (.class) em (.dex)
- 3 Tempo de execução

Segundo OLIVEIRA (2012), é possível descrever cada camada da seguinte forma:

- **Applications:** Esta é a camada mais alta da arquitetura, onde esta localizada uma lista de aplicações padrões, sendo capazes de realizar uma infinidade de atividades diretas com o usuário, como: calendário, navegador, despertador, *e-mail*, mapas, jogos e outros que serão desenvolvidos.
- **Framework Applications:** Nesta camada são fornecidas todas as funcionalidades necessárias para a construção de aplicativos, através das bibliotecas nativas. Ela é o elo entre a camada de aplicativos e a de bibliotecas, promovendo a facilidade de reutilização de procedimentos.
- **Libraries:** No Android® encontra-se um conjunto de bibliotecas C / C++ utilizadas por diversos componentes do sistema. Elas têm a capacidade de expor funcionalidades aos desenvolvedores através da estrutura de aplicativos Android®. Algumas das principais bibliotecas são: LibWebCore, SGL, Bibliotecas 3D, FreeType, SQLite.
- **Android Runtime:** Está no mesmo nível da camada de bibliotecas e é composto pela máquina virtual Dalvik, sendo responsável por permitir a execução de cada processo em sua própria instância virtual. A Dalvik é otimizada para utilizar pouca memória e permitir que tenha várias execuções de uma só vez.
- **Linux Kernel:** É a base para a execução das aplicações, onde acontece a abstração do *hardware* para o *software*. Foi baseado no *Kernel* do Linux versão 2.6, sendo responsável pelos serviços centrais do sistema, tais como segurança, gerenciamento de memória, gestão de processos, pilha de rede e modelo de *driver*.

O Android® é um sistema operacional baseado em um licenciamento *open source* podendo ser livremente estendido para incorporar novas tecnologias de ponta quando elas surgem. A plataforma continua a evoluir com a comunidade de desenvolvedores trabalhando em conjunto para construir aplicações móveis inovadoras (OHA, 2014).

Os desenvolvedores de *softwares* para *smartphones* seguem trabalhando com a ideia de tornar este equipamento essencial e necessário no dia a dia, sendo capaz de dispensar outros aparelhos de diversas funcionalidades. Atualmente os *smartphones* realizam tarefas de outros equipamentos, como por exemplo, os despertadores, o GPS (*Global Positioning System*), aparelhos de reprodução de música e vídeo, dentre outros. Ainda assim, existem ramos pouco explorados, como por exemplo, o uso na saúde. (OLIVEIRA *et al.*, 2011).

Um dos fatores importantes para os aplicativos refere-se aos mecanismos de armazenamento e recuperação de dados. Segundo Pereira e Silva (2009), o Android® oferece quatro mecanismos para armazenar e recuperar dados:

- Preferências compartilhadas: é um mecanismo simples para armazenar e recuperar pares de valores. Retorna o tipo e o valor do dado. Normalmente é usado para armazenar e recuperar as preferências do usuário. Estes dados irão persistir nas sessões do usuário, mesmo que o aplicativo seja finalizado.
- Arquivos: consiste no armazenamento de arquivos diretamente na memória interna do dispositivo ou em uma memória externa, o cartão SD.
- Base de Dados: o Android® oferece o suporte à criação e utilização de bases de dados SQLite®. É uma ferramenta simples e eficiente que permite armazenar os dados da aplicação em tabelas e manipular estes dados através de comandos SQL (*Structured Query Language*).

- Rede: é possível usar a rede (quando disponível) para armazenar e recuperar dados de servidores na Web, com a manipulação de endereços da Internet e solicitações HTTP (*Hypertext Transfer Protocol*).

Desenvolver aplicações para Android significa compor uma série de componentes para que o objetivo final da aplicação seja atingido. (Figura 3).



Figura 3: Componentes de uma aplicação Android
Fonte – LECHETA (2013)

Activities são as representantes das telas da aplicação. Associada a uma *activity* normalmente existe uma *view*, que define como será feita a exibição visual para o usuário. As *activities* são responsáveis por gerenciar os eventos de tela e também coordenam o fluxo da aplicação.

Os **Services** são códigos que executam em segundo plano. Normalmente são utilizados para tarefas que demandam um grande tempo de execução.

Os **Content Providers** (provedores de conteúdos) são a maneira utilizada pela plataforma para compartilhar dados entre as aplicações que executam no dispositivo. Um exemplo bem claro disto é a aplicação de gerenciamento de contatos do Android, que é nativa. Aplicações desenvolvidas por terceiros podem utilizar um

content provider a fim de ler os contatos armazenados no dispositivo de forma simples.

Os **Broadcast Receivers** são componentes que ficam "escutando" a ocorrência de determinados eventos, que podem ser nativos ou disparados por aplicações. Uma aplicação pode, por exemplo, utilizar um *broadcast receiver* para ser avisada quando o dispositivo estiver recebendo uma ligação e, com base nessa informação, realizar algum tipo de processamento.

Junto a estes componentes existe o manifesto **AndroidManifest.xml**. Ele é obrigatório e único para cada aplicação. É nele que são feitas as configurações gerais da aplicação e dos componentes que fazem parte dela. E, juntando tudo isto, existe a figura do **Android Core**, que na verdade não é um componente específico, mas sim a plataforma Android propriamente dita. É ele quem proporciona a interação entre os componentes e as aplicações e torna possível a execução do código.

Aplicações Android® tem uma característica de execução, compreender o ciclo de vida (Figura 4) de uma *activity*⁴ é muito importante, isto é, os possíveis estados em que elas se encontram, podendo ser: executando, temporariamente interrompida em segundo plano ou completamente destruída. O sistema operacional cuida desse ciclo de vida, mas ao desenvolver aplicações é importante levar cada estado possível em consideração para desenvolver uma aplicação mais robusta. O Android® fornece toda a estrutura necessária para isso, basta que se entenda o ciclo de vida de uma *activity* e implementá-la corretamente (LECHETA, 2013).

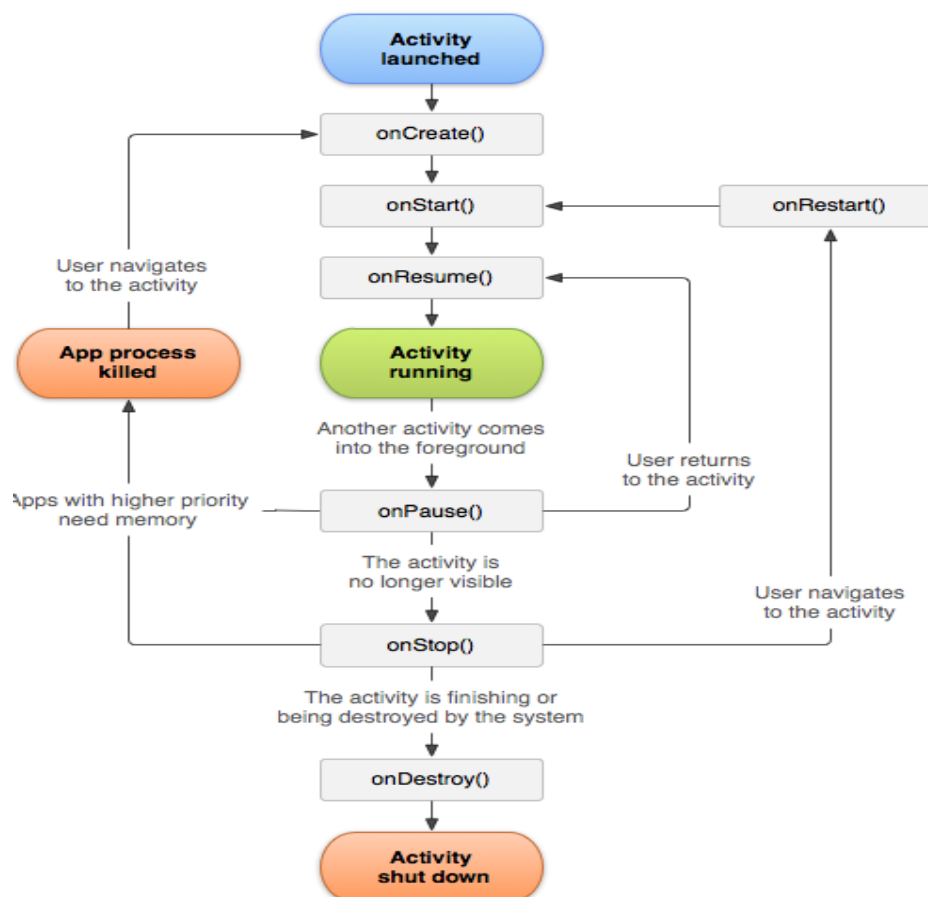


Figura 4: Ciclo de vida de uma *activity*
 Fonte – LECHETA (2013)

⁴ No Android® cada tela é representada por uma *activity*, que tem a finalidade de interagir com o usuário e tratar os eventos da tela, isso, basta que se entenda o ciclo de vida de uma *activity* e implementá-la corretamente (LECHETA, 2013).

A Figura 4 demonstra o ciclo de vida completo de uma *activity*, exibindo os estados possíveis e a chamada de cada método. Dos métodos mostrados acima, o único obrigatório para codificação é o `onCreate()`. Este método é executado quando a Activity é chamada, e sua principal função costuma ser apresentar a tela associada à Activity

Após a execução do método `onCreate()`, outros dois métodos são executados: `onStart()` e `onResume()`. Basicamente, estes são diferenciados pelo momento que são executados: o primeiro Método chamado quando a *activity* está ficando visível ao usuário e o segundo quando uma *activity* está parada temporariamente e esta sendo iniciada novamente, o método `onStart()` é chamado de forma automática.

A diferença entre o estado de pausa e parado é que o primeiro acontece quando a tela está parcialmente visível, na verdade, quando uma Activity está executando e uma nova tela parcial aparece, como a tela do despertador do dispositivo ou uma tela de Dialog. A Activity que estava na tela entra em estado de pausa, enquanto a segunda tela parcial é apresentada.

Já o estado de parado acontece quando a Activity sai da tela do dispositivo, dando lugar a outra aplicação ou Activity (por exemplo, a aplicação de ligação, calculadora, calendário, etc.). Neste caso, a Activity é parada, já que seu conteúdo na tela foi substituído pelo conteúdo de outra aplicação.

Para o estado de pausa, os métodos `onPause()` e `onResume()` são executados.

Já para o estado de parado de uma Activity, o método `onPause()` e `onStop()` são executados, e após estes, uma nova aplicação ou Activity assume o topo da pilha de Activities. Ao retornar para a Activity que está parada, os métodos `onRestart()`, `onStart()` e `onResume()` são executados, para só então a Activity assumir o topo da pilha.

Por fim, o método `onDestroy()` é chamado quando a Activity é encerrada, e este método é antecedido pelos métodos `onPause()` e `onStop()`. Assim, cabe ao usuário decidir quais métodos serão melhor aproveitados em situações como salvar o estado atual da aplicação, alocar ou liberar recursos como câmera digital, salvar informações em arquivos de log e muito mais.

Outro fator importante a ser considerado no desenvolvimento para dispositivos móveis é a usabilidade, que pode ajudar a minimizar as dificuldades de interação com o usuários. Técnicas de avaliação de usabilidade específicas para aplicações móveis podem beneficiar o processo de desenvolvimento em tais dispositivos. Várias técnicas de avaliação de usabilidade têm sido propostas, como os *User Interface Guidelines* que são guias para auxiliar o desenvolvedor a aproveitar ao máximo os recursos de interface da plataforma, mas estas técnicas ainda não estão consolidadas (BONIFÁCIO *et al.*, 2010).

Segundo o Android (2014) as aplicações para este sistema operacional devem seguir três objetivos, são eles:

- Aplicativos Android® devem ser elegantes e ter *interfaces* agradáveis, as transições devem ser de fácil acesso, rápidas e claras, o *layout*, a tipografia e os ícones devem ser nítidos e significativos. A aplicação deve combinar beleza, simplicidade e criar uma experiência mágica, fácil e poderosa ao usuário.
- Ao utilizarem o aplicativo pela primeira vez, as pessoas precisam compreender as características mais importantes. Tarefas simples nunca devem exigir procedimentos complexos e tarefas complexas devem ser adaptadas para facilitar.
- Aplicativos Android® capacitam as pessoas a experimentar coisas novas e usá-las de novas maneiras, permitindo que os usuários combinem aplicações em novos fluxos de trabalho por meio de multitarefas, com outros aplicativos, notificações e compartilhamento.

O Android® oferece um sofisticado e poderoso modelo, baseado em componentes para construir sua *interface*, baseado no esquema de classes: *widgets* e *layouts* (LECHETA, 2010). Também é possível criar subclasses para personalizar algum componente ou até criar novas classes para personalizar totalmente os elementos gráficos.

Alguns exemplos de *widgets* e *layouts*: Button, TextView, EditText, ListView, CheckBox, RadioButton, Gallery, Spinner, ProgressBar, AnalogClock, ImageView, ImageButton, LinearLayout, FrameLayout, RelativeLayout, GridLayout, dentre outros, como mostrado na Figura 5.

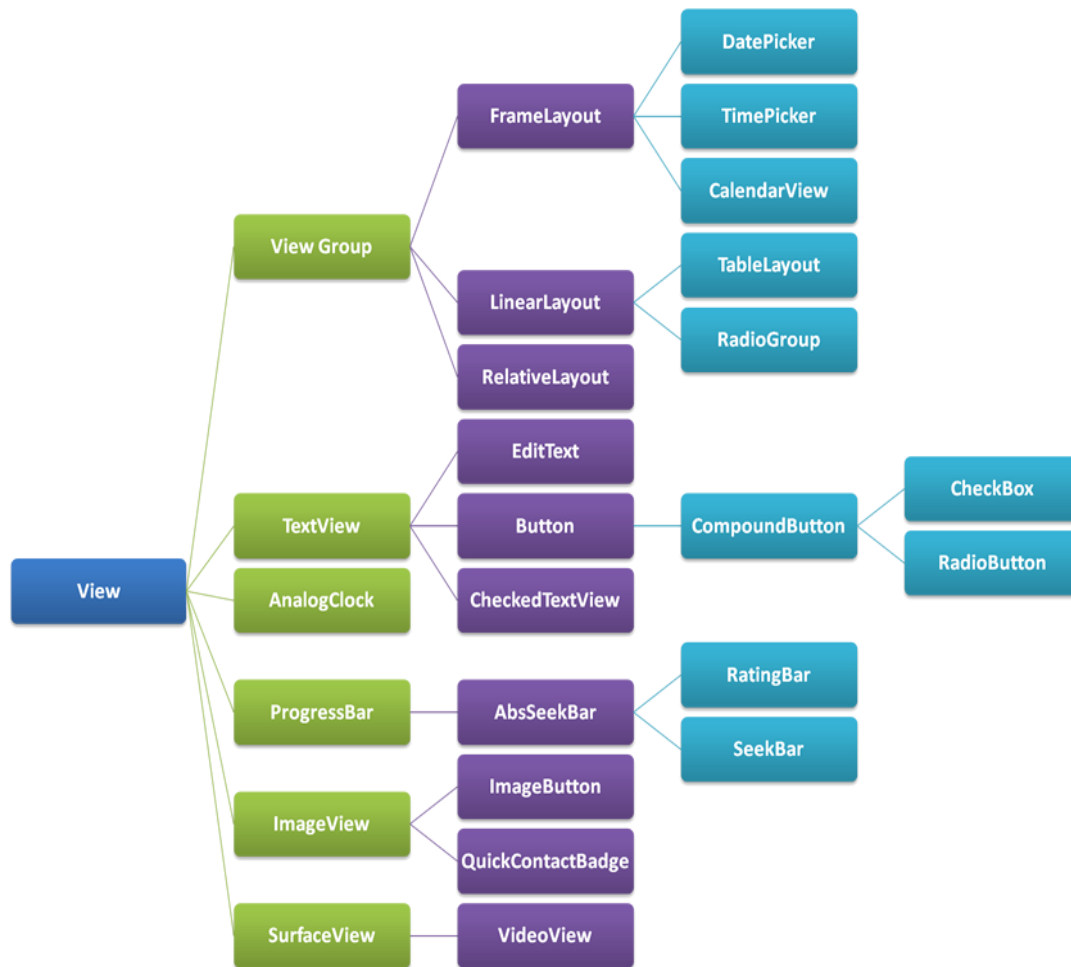


Figura 5: Hierarquia dos *widgets* no Android®

FONTE : Isux (2014)

3. Metodologia

3.1. Ambiente de Desenvolvimento

A escolha do Sistema Operacional Android® foi feita primeiramente pelo conhecimento prévio que já possuía sobre a plataforma, por possuir conhecimento na linguagem JAVA e pelo ambiente de desenvolvimento sem custo.

As ferramentas utilizadas no desenvolvimento do aplicativo Controle de Alimentação e medições de glicemia foram: o Android SDK (*Software Development Kit*), a IDE (*Integrated Development Environment*) Android Studio ,a linguagem de programação Java (Nativa para Android), o banco de dados SQLite e o Geny motion, uma ferramenta que oferece uma interação realística com dispositivos móveis assim como o AVD(Android Virtual Device), porém com mais eficiência e velocidade se comparados.

O Android SDK é o *kit* de desenvolvimento provido pela Google® que contém as ferramentas e bibliotecas necessárias para construir e testar aplicações para Android®. Junto com o SDK é disponibilizado o simulador de dispositivos Android®,o AVD (*Android Virtual Device*) que por todas as limitações e seu baixo desempenho não foi utilizado, tomando como melhor opção a utilização do software GenyMotion(Figura 5), que por sua vez possibilitou realizar todos os testes necessários, com a vantagem de poder simular os diversos tamanhos de tela, resoluções e versões do sistema, tudo isso gratuitamente. O Android Studio foi escolhido por ser uma ferramenta que dispõe de diversos recursos e oferecer maior praticidade para o desenvolvimento de aplicações para Android®.



Figura 6: Tela Principal do aplicativo Diabetes DevFS emulado no software GenyMotion
Fonte – Elaborada pelo autor

3.2. Ferramentas utilizadas

3.2.1. Android Studio

Android Studio é um ambiente de desenvolvimento integrado (IDE) para a plataforma Android. Foi anunciado em 16 de Maio de 2013, na conferência da Google I/O pelo Diretor Executivo, Ellie Powers e então entrou em fase de pré-visualização, em seguida, entrou em estágio beta a partir da versão 0.8, que foi lançado em Junho de 2014. Com base em software IDEA JetBrains 'IntelliJ, o Android Studio é projetado especificamente para o desenvolvimento do Android. Ele está disponível para as plataformas Windows, Mac OS X e Linux, além disso é distribuído de forma totalmente gratuita para os desenvolvedores.

Atualmente, alguns dos recursos mais importantes fornecidos pelo ambiente de desenvolvimento são:

- Codificação ao vivo - em tempo real App Rendering
- Gradle à base de apoio de construção.
- Ferramentas Lint para identificar e medir desempenho, usabilidade, compatibilidade de versão e outros problemas.
- Assistentes com base em modelo para criar designs Android comuns e componentes.
- Um editor de layout rico que permite aos usuários arrastar e soltar componentes de interface do usuário, opção de layouts de visualização em múltiplas configurações de tela.
- O suporte integrado para Cloud Platform Google, permitindo a integração com o Google Cloud Messaging e App Engine.

3.2.2. GenyMotion

Devido ao emulador AVD(Android Virtual Device) do Android Studio possuir um desempenho não muito agradável , Geny Motion foi a escolha perfeita para um emulador rápido, consistente e ótimo para rodar e testar este aplicativo.

O emulador consiste em uma solução simples para o desenvolvedor Android que busca rapidez e praticidade para executar suas aplicações.

Genymotion se baseia no trabalho anterior do projeto AndroVM de código aberto para oferecer um ambiente Android realista e totalmente funcional com base em uma gama de diferentes telefones e tablets.

Para isso o desenvolvedor irá apenas precisar da VirtualBox instalada em sua máquina , e também cadastrar uma conta no site da Genymotion (Isso tudo é gratuito) para então iniciar o download de máquinas virtuais.

Realismo é a característica mais chamativa desta aplicação, com Genymotion configurado para entregar o mais preciso de uma experiência possível. Isso significa apoiar uma ampla gama de protocolos de rede, incluindo GPS, 3G e Wi-Fi, aceleração OpenGL e multi-display, níveis da bateria e muito

mais, e ainda conta com sensores adicionais, atualmente em fase de desenvolvimento.

Genymotion é voltada principalmente para desenvolvedores, testadores e até mesmo para jogadores e quem quer experimentar Android sem ter que comprar um smartphone ou tablet pela primeira vez.

3.2.3. SQLite

O uso dos smartphones cresceu muito nos últimos anos. Isso acarretou em um grande esforço das fabricantes em melhorias no hardware e das gigantes de software um reposicionamento quanto aos sistemas operacionais para estes tipos de dispositivos. E isso gerou uma bola de neve que trouxe grandes e constantes inovações no setor, refletindo cada vez mais em usuários e desenvolvedores mais satisfeitos com as novas tecnologias, que os permitem ficar mais tempo conectados.

Isso também teve outra consequência direta: os usuários passaram a armazenar seus dados não só nos computadores Desktop, mas também em seus smartphones. Logo, estes dispositivos passaram a receber uma quantidade maior e crescente de dados. Principalmente dados de áudio e mídia.

Logicamente, as plataformas móveis perceberam que teriam que fornecer métodos de persistência de dados mais sofisticados que o tradicional armazenamento baseado em arquivos, como acontecia com o RMS (Record Management System), este utilizado pelos programadores Java ME/MIDP para a persistência dos dados. Hoje em dia, todas as principais plataformas oferecem mais de uma maneira de salvar e recuperar informações produzidas por nossos aplicativos.

Com a plataforma Android não poderia ser diferente. Este fornece diferentes métodos de persistência. Alguns deles são através de armazenamento interno (no smartphone) e externo (cartão de memória). Ambos podem ser muito úteis para fazer armazenamento de imagens, por exemplo. No caso de armazenamento em cartão, deve-se atentar para não utilizá-los no armazenamento de informações essenciais para o correto funcionamento do

aplicativo, pois a mídia pode ser removida a qualquer momento, e o aplicativo deixará de ser executado.

Além disso, a plataforma também fornece as classes chamadas *SharedPreferences*, sendo esta uma forma de persistir pares de chave/valor para todos os tipos primitivos Java e mais o tipo *String*. Algo semelhante à utilização de *Properties* na programação Java tradicional. Seu uso e funcionamento são muito simples, podendo ser muito útil para persistências simples, como a pontuação de um jogo, por exemplo.

Porém, o *SharedPreferences* não é indicado para grande quantidade de dados ou conjunto de informações com grande complexidade. Nesse momento entram em ação os bancos de dados relacionais, utilizados há alguns anos em outros ambientes, como web e desktop, por exemplo. No Android o banco de dados mais utilizado é o *SQLite*.

Uma característica interessante da plataforma é que o *SQLite* já está disponível na plataforma Android, não havendo necessidade de instalá-lo. Além disso, o Android oferece suporte completo ao banco, através de uma API com um rico conjunto de classes e métodos que abstraem as complexidades dos códigos SQL. Assim, não precisamos montar a cláusula SQL inteira para atualizar uma linha na tabela, ou ainda, para fazer uma pesquisa na mesma. O Android nos fornece um método, onde passando alguns parâmetros obtemos um apontador para os dados retornados, podendo navegar pelo resultado como se estivéssemos escolhendo uma folha em um arquivo. Assim então o Banco *SQLite* se torna o mais propício para o desenvolvimento desta aplicação.

3.3. Arquitetura MVC

Um fator importante a ser observado, trata-se do modelo em camadas baseado em MVC (*Model-View-Controller*) que o desenvolvimento Android® utiliza. O MVC é um padrão de arquitetura de *software* fundamental na separação dos dados (*Model*), do *layout* (*View*) e dos controladores (*Controller*). Desta forma, alterações feitas no *layout* não afetam a manipulação de dados nem os

controladores, estes poderão ser reorganizados sem alterar o *layout* (MÄHLMANN; WAGNER, 2012).

Conforme explicado por Gonçalves (2008), o MVC (Figura 7) possui os seguintes componentes:

- **Model:** O *Model* (Modelo) é a camada que representa os seus dados, provendo meios de acesso à leitura e escrita. Essa camada não possui conhecimento específico dos controladores (*controller*) e das apresentações (*views*).
- **View:** É na camada *View* (Apresentação) que acontece a interação com os usuários, apresentando as entradas e saídas das informações obtidos pelo *Model*.
- **Controller:** O *Controller* (Controlador) é responsável por controlar todo o fluxo de informação que passa pela aplicação, servindo como uma camada intermediária entre a camada *view* e a *model*. Dirige as apresentações a serem exibidas e com as devidas mudanças de dados vindas da camada modelo.

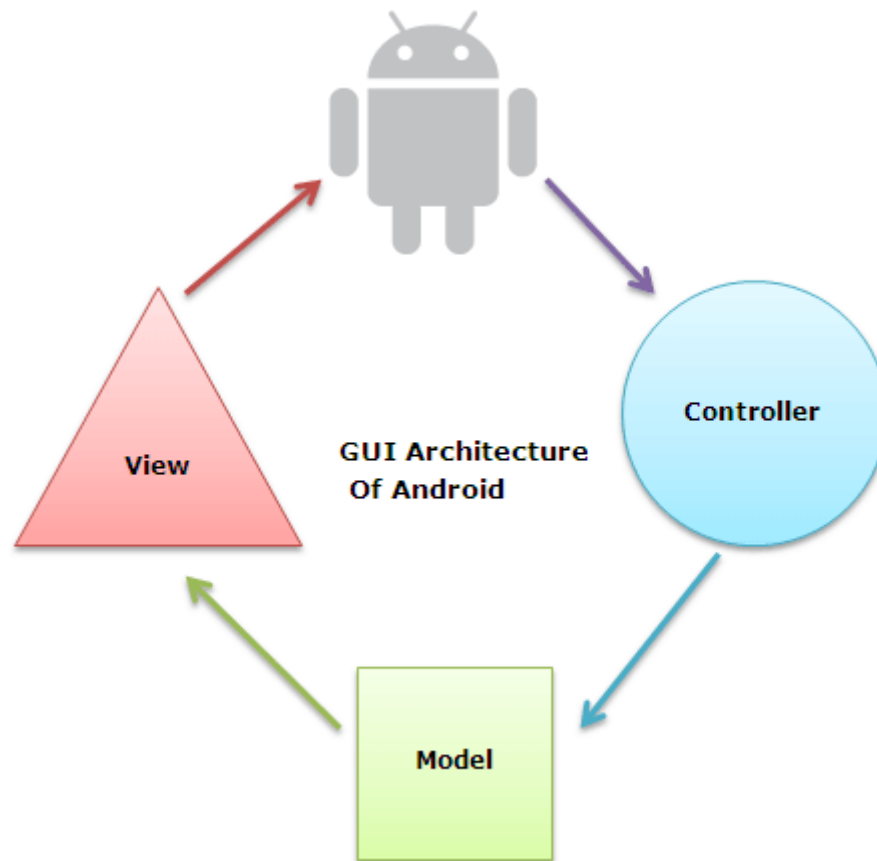


Figura 7: Diagrama *Model-View-Controller*

Fonte : [StackOverflow.com/mvc-model-view-controller-structure](https://stackoverflow.com/mvc-model-view-controller-structure)

Vários problemas podem surgir quando os *softwares* tem associação no código de acesso aos dados, controles e apresentação dos dados. Tais aplicações com os códigos desorganizados podem ser de difícil entendimento e manutenção, pois todos os componentes estão juntos, causando fortes efeitos em cascata sempre que uma alteração precisa ser feita em qualquer lugar. Com esse acoplamento se torna praticamente impossível à reutilização do código. (ORACLE, 2002).

A separação lógica da aplicação nestas partes assegura que a camada de modelo fique restrita a representar os componentes do problema. Igualmente, a camada de apresentação que está relacionada na exibição dos dados. Já o controlador, tem a responsabilidade de atuar como um intermediador (GONÇALVES, 2008).

4. Desenvolvimento

As etapas do desenvolvimento foram: design das telas, criação da *interface*, programação do aplicativo em JAVA, realização do CRUD no banco de dados SQLite, testes no emulador e em diversos dispositivos para verificar erros, publicação do aplicativo gratuitamente no Google Play®, tornando acessível a qualquer usuário da plataforma Android® com a versão 4.0.x (ICS) ou mais recente, correções de *bugs* e melhorias, criação do módulo de acompanhamento médico e por fim a criação do painel Web para visualização.

4.1. Sistema Mobile

Analisando os dados disponibilizados pelo Google® (Figura 8), foi feita a escolha da versão 4.4 KitKat API (*Application Programming Interface*) 21 como versão mínima de utilização para o desenvolvimento do aplicativo Diabetes DevFs, devido a maioria dos seus usuários que utilizarem esta versão ou inferiores, além de contar com os recursos que só podem ser utilizados com essa API durante o desenvolvimento.

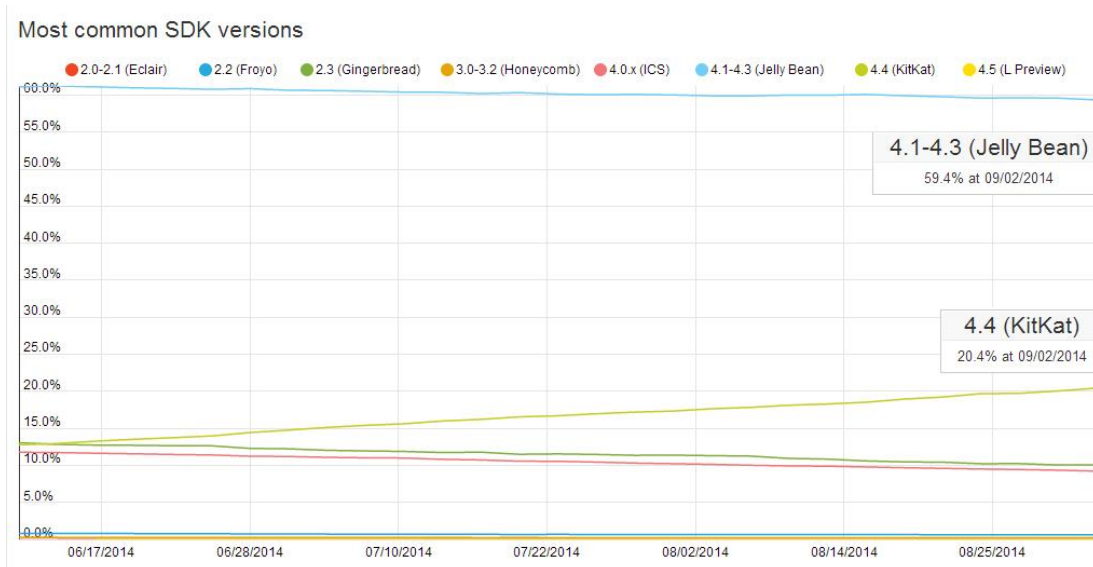


Figura 8 : Utilização de versões do Android®
 FONTE : AppBrains(2014)

No Android®, é possível criar *interfaces* gráficas por dois modos diferentes, a partir da codificação Java dentro do próprio código do aplicativo ou pelo arquivo XML (*Extensible Markup Language*) que é carregado no início da aplicação. Para manter o aplicativo em modelo de camadas, a *interface* foi desenvolvida usando arquivos XML, desta forma existe a separação da apresentação com o código da aplicação. A primeira impressão visual é muito importante, é a oportunidade de manter o foco do usuário na sua aplicação e passar as especialidades que seu aplicativo vai oferecer.

4.2. Detalhamento do desenvolvimento

Diabetes DevFS tem o intuito de controlar seus registros de medição de glicemia assim como a alimentação diária do usuário, salvando-os e assim ajudando na visualização e facilitando o compartilhamento.

Quando o aplicativo é executado, a tela principal é apresentada, onde acontece toda a interação do usuário com o aplicativo. Podendo navegar entre os menus e escolher a função que ele desejar. O menu principal conta com 3 abas, sendo elas, ‘Marcações’, ‘Visualização’ e ‘Configurações’.

A aba marcações tem por objetivo a entrada dos dados em geral, já a aba de visualização oferece a visualização dos dados gravados no aplicativo, bem como a conversão dos mesmos para o formato .pdf . E por ultimo a aba de configurações dá ao usuário a opção de escolha do que melhor lhe convier de acordo com sua necessidade, onde o mesmo escolherá de forma personalizado como interagir com o aplicativo.

Esses itens foram escolhidos por serem as informações mais importantes sobre a condição de saúde do usuário, estes itens irão auxiliar os próprios usuários e os profissionais da área de saúde sobre quais as ações necessárias para o auxílio no controle do usuário proporcionando-o uma vida mais saudável. Todas essas funções serão abordadas mais a frente de forma mais detalhada.

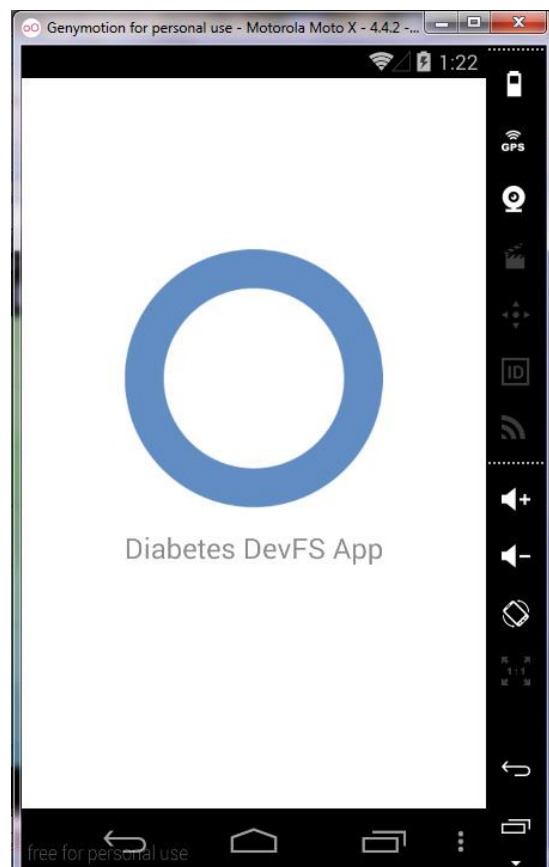


Figura 9: Tela de carregamento ao abrir o aplicativo Diabetes DevFS emulado no software GenyMotion

Fonte – Elaborada pelo autor



Figura 10: Tela principal do aplicativo Diabetes DevFS emulado no software GenyMotion
Fonte – Elaborada pelo autor

A inclusão de um novo registro pode ser realizada pelo botão Nova Marcação no menu da tela principal. Ao clicar, uma nova janela é apresentada conforme a Figura 11. Esta tela contém os campos glicose que seria o nível de açúcar no sangue medido pelo usuário naquele determinado momento, insulina que representa quantas unidades ou mls (Configurado pelo usuário na tela de configurações) ele injetou no momento da medição, o campo de teor de carboidratos que auxilia o usuário realizar a contagem de carboidratos ingerida no dia¹ e por último a opção de escolher se é uma marcação diurna ou noturna. Ao salvar, os dados são armazenados no banco de dados interno do aplicativo.

¹Controle de Carboidratos : A Contagem de Carboidratos é uma terapia nutricional, onde contabiliza-se os gramas de carboidratos consumidos nas refeições e lanches, com o objetivo de manter a glicemia dentro de limites convenientes. A razão pela qual se contam as gramas de carboidratos é porque os carboidratos tendem a ter maior efeito na sua glicemia.



Figura 11: Tela de marcação de glicemia do aplicativo Diabetes DevFS emulado no software GenyMotion

Fonte – Elaborada pelo autor

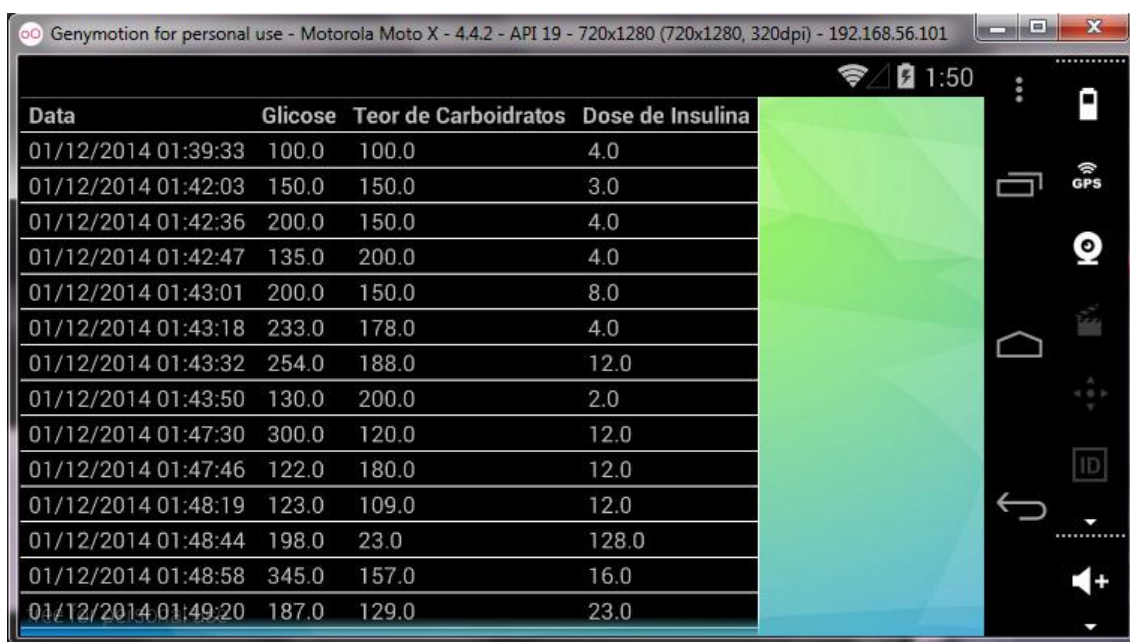
Na figura 12 é apresentada a tela de avaliação, onde será possível avaliar uma medição anteriormente feita avaliando-a de uma a cinco estrelas. Esse tipo de avaliação atrai o olhar da criança, uma vez que a mesma deseja sempre obter um nível de cinco estrelas. Envolvida então por uma disputa saudável a mesma se encaminhará com um premissa de permanecer sempre saudável, obtendo assim uma melhor avaliação em suas marcações.

Nesta função também é possível adicionar observações extras sobre aquela marcação de glicemia.



Figura 12: Tela de Avaliação do aplicativo Diabetes DevFS emulado no software GenyMotion
Fonte – Elaborada pelo autor

Já na aba de visualização temos na primeira opção a função de visualização dos dados, onde é mostrado ao usuário todas as medições feitas desde o início da utilização do software como vemos na figura 13.



Data	Glicose	Teor de Carboidratos	Dose de Insulina
01/12/2014 01:39:33	100.0	100.0	4.0
01/12/2014 01:42:03	150.0	150.0	3.0
01/12/2014 01:42:36	200.0	150.0	4.0
01/12/2014 01:42:47	135.0	200.0	4.0
01/12/2014 01:43:01	200.0	150.0	8.0
01/12/2014 01:43:18	233.0	178.0	4.0
01/12/2014 01:43:32	254.0	188.0	12.0
01/12/2014 01:43:50	130.0	200.0	2.0
01/12/2014 01:47:30	300.0	120.0	12.0
01/12/2014 01:47:46	122.0	180.0	12.0
01/12/2014 01:48:19	123.0	109.0	12.0
01/12/2014 01:48:44	198.0	23.0	128.0
01/12/2014 01:48:58	345.0	157.0	16.0
01/12/2014 01:49:20	187.0	129.0	23.0

Figura 13: Tela de Visualização de dados do aplicativo Diabetes DevFS emulado no software GenyMotion

Fonte – Elaborada pelo autor

A função de exportar dados realiza a exportação direta do histórico de marcações de glicemia para o formato .pdf, e o documento fica salvo na pasta criada pelo aplicativo juntamente aos documentos do seus smartphone. (Figura 14)

Data :	Glicose :	Teor de carboidrato	Dose de insulina :	Classificação	Comentário
0 :	0 :	g :	g :	0 :	5 :
01/12/2014 01:39:33	100.0	100.0	4.0	N/A	N/A
01/12/2014 01:42:03	150.0	150.0	3.0	N/A	N/A
01/12/2014 01:42:36	200.0	150.0	4.0	N/A	N/A
01/12/2014 01:42:47	135.0	200.0	4.0	N/A	N/A
01/12/2014 01:43:01	200.0	150.0	8.0	N/A	N/A
01/12/2014 01:43:19	233.0	178.0	4.0	N/A	N/A
01/12/2014 01:43:32	254.0	188.0	12.0	N/A	N/A
01/12/2014 01:43:39	130.0	200.0	2.0	N/A	N/A
01/12/2014 01:43:50	300.0	120.0	12.0	N/A	N/A
01/12/2014 01:47:46	122.0	180.0	12.0	N/A	N/A
01/12/2014 01:48:19	123.0	109.0	12.0	N/A	N/A
01/12/2014 01:49:44	198.0	23.0	128.0	N/A	N/A
01/12/2014 01:48:38	345.0	157.0	16.0	N/A	N/A
01/12/2014 01:49:30	187.0	129.0	23.0	N/A	N/A
01/12/2014 01:49:37	183.0	123.0	12.0	N/A	N/A

Figura 14: Tela do Relatório exportado para .pdf do aplicativo Diabetes DevFS emulado no software GenyMotion

Na aba de configurações temos a função de medidor de insulina que tem como objetivo dar a opção ao usuário o controle dos níveis da sua caneta de injeções, onde o mesmo pode regular a capacidade máxima de sua caneta e podendo observar seus níveis, para que não aconteça do usuário ficar sem sua caneta de insulina, o que é muito perigoso para a sua saúde.

Uma observação importante é que durante a realização das marcações de glicemia, caso os níveis das canetas (tanto diurna quanto noturna) estiverem baixos, o aplicativo notificará o usuário sobre isso para que o mesmo tome as devidas providências.

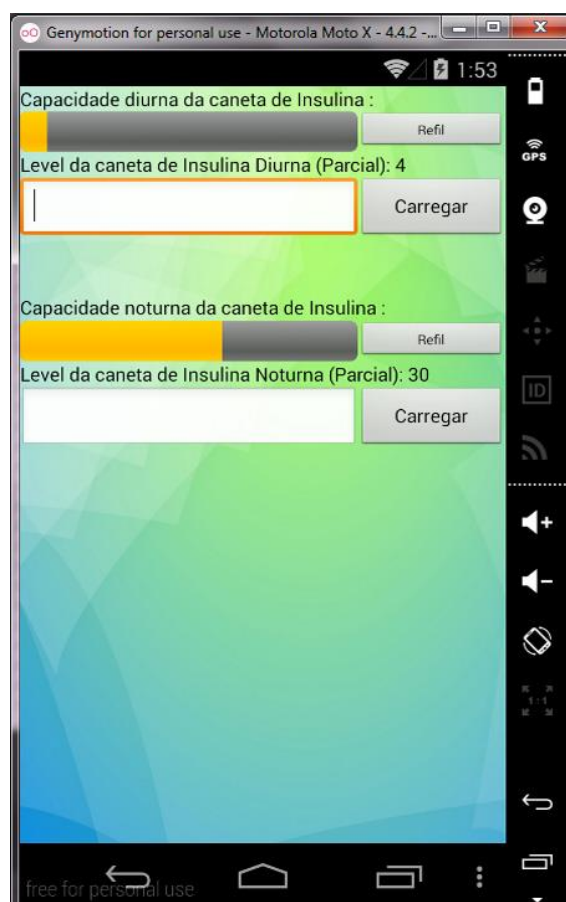


Figura 15: Tela Medido de insulina do aplicativo no app Diabetes DevFS emulado no software GenyMotion

Fonte – Elaborada pelo autor

E por último o recurso de configurações, onde o usuário terá a opção de escolher suas preferências em relação as suas marcações do seu nível de glicemia, como o nível máximo de suas canetas de insulina, para que possa ser reproduzido de forma real sua interação com o aplicativo, além de poder definir em qual ponto será emitido um alerta caso os níveis de insulina da sua caneta estiverem baixos.

Além de poder configurar se suas medições serão feitas em unidades ou mls.

A partir da escolha desses valores é possível utilizar um aplicativo configurado com suas preferências e uma reprodução bem real do seu cotidiano durante suas marcações de glicemia.

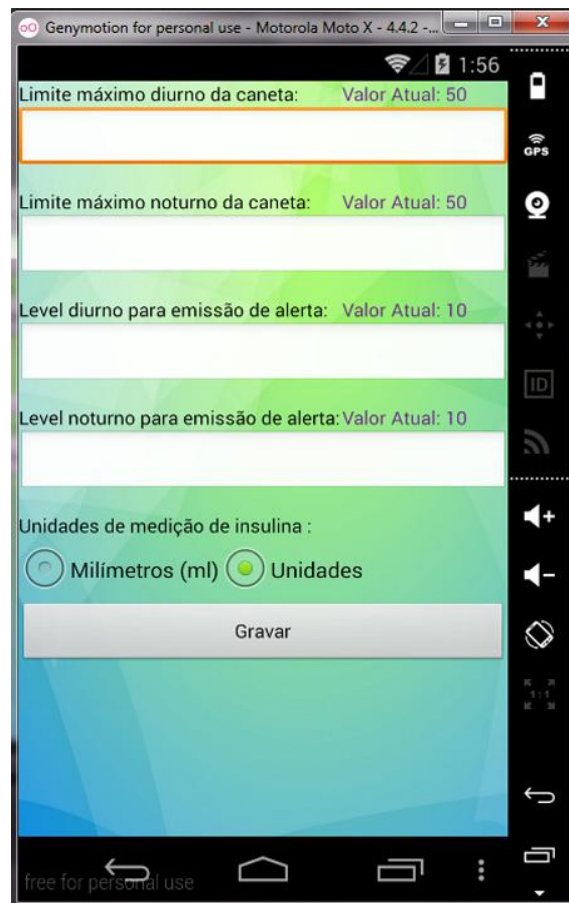


Figura 16: Tela de configurações do app Diabetes DevFS emulado no software GenyMotion
Fonte – Elaborada pelo autor

5. Resultados

O aplicativo Diabetes DevFS ficou no Google Play® por um mês (Figura 17), nesse período alcançou 186 *downloads*, das quais 96 instalações são atuais.

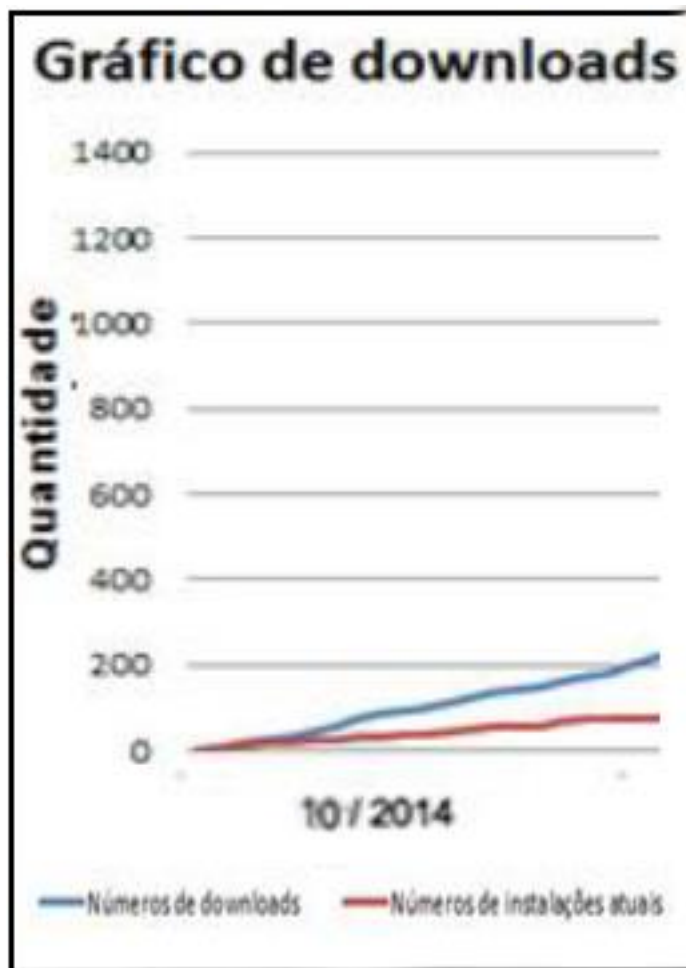


Figura 17: Relatório de *downloads* e instalações atuais do aplicativo.
Fonte – Elaborada pelo autor

Analisando a Figura 18, é possível observar as instalações e desinstalações diárias que ocorreram no aplicativo. A média diária de instalações é de 13,7 e a de desinstalações é de 10,2.

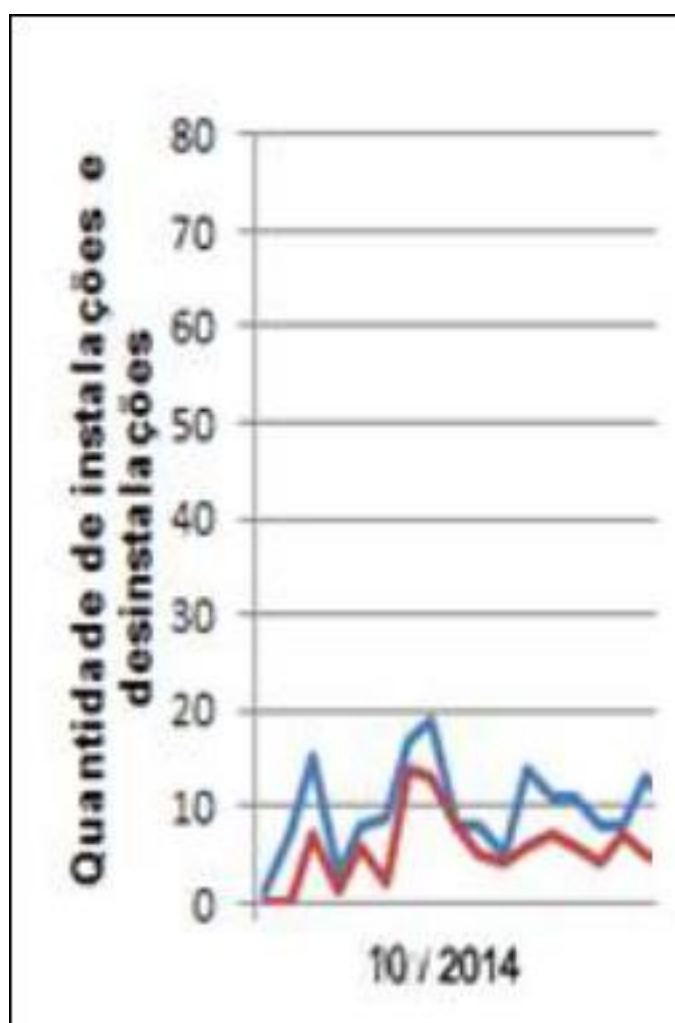


Figura 18 : Instalações e desinstalações diárias
Fonte – Elaborada pelo autor

Dentre os usuários do sistema, pode-se observar que existe uma equivalência na utilização do sistema pelos diferentes sexos, sendo 51% dos usuários do sexo feminino (Figura 19).

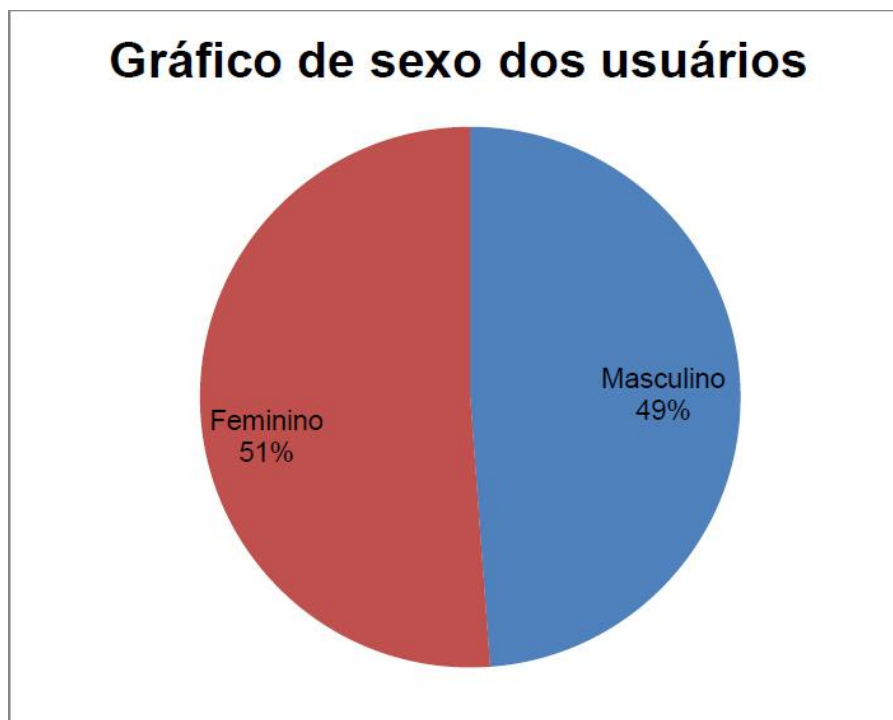


Figura 19: Sexo dos usuários
Fonte – Elaborada pelo autor

Segundo as informações obtidas pelo Console do desenvolvedor, a versão do Android® mais utilizada para fazer o *download* do aplicativo é a 4.1 com 34,15%, seguida da versão 2.33, como pode ser visto na Figura 19. Inclusive existe uma ampliação da menor faixa, onde a mesma representa 7,38% dos dados referentes as outras versões do Android®.

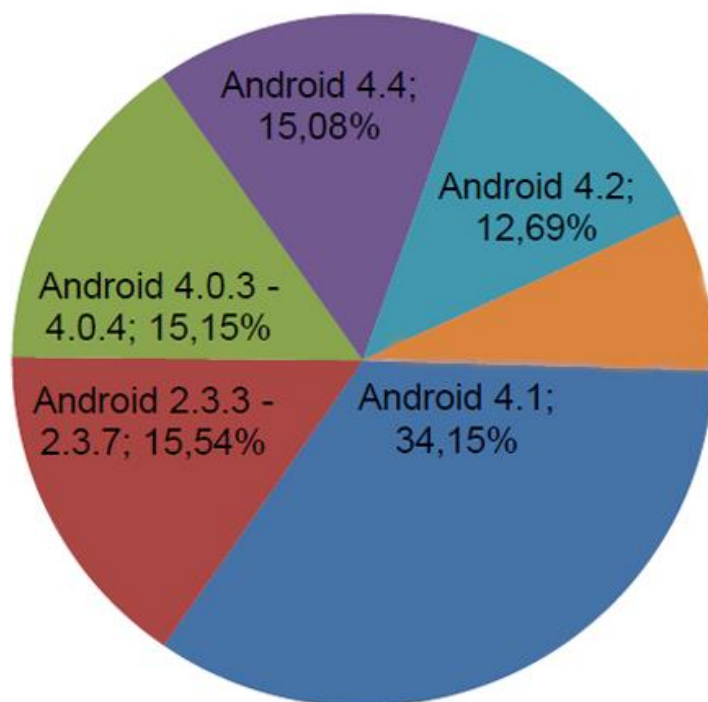


Figura 20: Gráfico da versão do Android® mais utilizada no aplicativo
Fonte – Elaborada pelo autor

6. Considerações Finais

6.1. Conclusões

O desenvolvimento do Diabetes DevFS possibilitou o estudo do diabetes e a integração com tecnologias que proporcionam mobilidade e praticidade ao usuário. Neste trabalho foi realizado o levantamento de informações com relação às necessidades das crianças diabéticas, tendo em vista oferecer com a mobilidade e a praticidade de uma aplicação móvel facilitar e oferecer melhor qualidade de vida.

Definido o domínio, estudou-se a melhor proposta para atender aos requisitos do controle da alimentação e das medições de glicemia diárias. Como achados do projeto, pode-se citar o ganho de eficiência onde a nutrição ganhou em avanços, permitindo aos usuários melhor organizar seu tempo e dedicação a atividades de relaxamento e lazer, sem se preocupar tanto com o monitoramento de atividades anteriormente feitas manualmente.

A vigilância ficava prejudicada, dado o dia a dia estressante e agitado dos pais quanto as crianças. Após os testes realizados percebeu-se que o protótipo Diabetes DevFS, otimiza o controle por parte do usuário e oferece maior praticidade e facilidades em todas as atividades exercidas pelos diabéticos e seus responsáveis. Em relação aos resultados no Google play, deu pra perceber que em apenas um mês o aplicativo foi baixado bastantes vezes, o que é um bom sinal. Porém não deu para avaliar com consistência tais resultados.

6.2. Trabalhos Futuros

A manutenção da saúde para uma pessoa diabética está diretamente ligada a sua alimentação.

Logo uma sugestão para trabalhos futuros é utilizar os recursos já existentes de cálculo do seu IMC (Índice de Massa Corporal). Utilizando-o para nortear seu usuário em relação a sua alimentação. Quanto maior estiver esse índice, mais prejudicada está sua saúde. O aplicativo então o orientará com medidas para que ele volte à zona da normalidade. Ele fará uma análise do perfil de consumo do usuário para a escolha de propostas que se encaixem melhor em seus hábitos alimentares, e assim, atingir o sucesso dos objetivos.

7. Referências Bibliográficas

ANDROID. Android Developers. Disponível em: <<http://developer.android.com/guide/index.html>>. Acesso em: 01 out 2014.

ANDROID. Brand Guidelines. Disponível em: <<http://developer.android.com/distribute/googleplay/promote/brand.html>>. Acesso em: 04 out 2014.

ANDROID. Creative Vision. Disponível em: <<http://developer.android.com/design/get-started/creative-vision.html>>. Acesso em: 04 out 2014.

ANDROID. Dashboards. Disponível em: <<http://developer.android.com/about/dashboards/index.html>>. Acesso em: 04 out 2014.

CANALYS. 2014. Smart phones, Worldwide, installed base by OS vendor, 2014. Disponível em: <<http://www.canalys.com/chart/index.html#display-98>>. Acesso em: 02 out 2014.

DEITEL, Paul; DEITEL, Harvey. JAVA - COMO PROGRAMAR. São Paulo: Pearson, 2010.

DIAS, M. S.; BITTAR, T. J.; MENDONÇA, V. R. L. Um estudo dos Sistemas Operacionais Android e iOS para o desenvolvimento de aplicativos. Disponível em: <http://www.enacomp.com.br/2011/anais/trabalhos-aprovados/pdf/enacomp2011_submission_54.pdf> Acesso em: 02 set 2014.

LECHETA, R. R. Introdução ao Android. In: _____. Google Android. 3. ed. São Paulo: Novatec, 2013.

LECHETA, Ricardo R.. Google Android: Aprenda a criar aplicações para dispositivos móveis com o Android SDK. São Paulo: Novatec, 2009.

LEE, V.; SCHENIDER, H. e SCHELLI, R. (2005) “Aplicações Móveis - Arquitetura, projetos e desenvolvimento”. São Paulo: Pearson Education do Brasil.

MASCARENHAS, M.; MARTINS, M.; BULCÃO, L.; BRITO, J.; VIEIRA, V.; DURAN, A. Um estudo de caso com análise comparativa entre plataformas para aplicações móveis aberta e proprietária: Android e iOS. 2013. Disponível em: <<http://www.cin.ufpe.br/~ubibus/artigos/112186.pdf>>. Acesso em: 02 out 2014.

MENDES, Douglas Rocha. Programação Java com Ênfase em Orientação a Objetos. São Paulo: Novatec, 2009.

MINISTÉRIO NACIONAL DA SAÚDE . Rotulagem nutricional obrigatória: manual de orientação aos consumidores. Brasília: Ministério da Saúde, Agência Nacional de Vigilância Sanitária / Universidade de Brasília, 2005. 17p.

ORACLE. Model-View-Controller. Disponível em: <<http://www.oracle.com/technetwork/java/mvc-140477.html>> Acesso em: 02 out 2014.

PEREIRA, L. C. O.; SILVA, M. L. Android para Desenvolvedores. Rio de Janeiro: BRASPORT, 2009.

RTC. Android Goes Beyond Google. Disponível em: <<http://www.rtcmagazine.com/articles/view/102586>>. Acesso em: 22 mar 2014.

SIGULEM, D.; SALOMÃO, P. Utilização do Computador de Mão Integrado à Telefonia Celular no Atendimento Médico: Desenvolvimento de Sistema e Avaliação. Disponível em: <<http://www.sbis.org.br/cbis9/arquivos/59.doc>>. Acesso em: 14 ago. 2014.

SOCIEDADE BRASILEIRA DE DIABETES. Manual de nutrição para profissionais de saúde. São Paulo: Sociedade Brasileira de Diabetes; 2009. Disponível em: <<http://www.diabetes.org.br/livros-e-manuais/550>> Acesso em: 14 set. 2014.

SOCIEDADE BRASILEIRA DE DIABETES. Manual de nutrição para o público. São Paulo: Sociedade Brasileira de Diabetes; 2009. Disponível em: <<http://www.diabetes.org.br/livros-e-manuais/549>> Acesso em : 15 set 2014