



UFAM - Engenharia da Computação

DIÁRIO DA SAÚDE: SISTEMA DE REGISTRO DE EVOLUÇÕES DIÁRIAS COM
COMUNICAÇÃO BLUETOOTH

Isadora Santarém de Queiroz

Orientadores:

Miguel Grimm do Nascimento

Manaus

Fevereiro de 2023



UFAM - Engenharia da Computação

DIÁRIO DA SAÚDE: SISTEMA DE REGISTRO DE EVOLUÇÕES DIÁRIAS COM
COMUNICAÇÃO BLUETOOTH

Isadora Santarém de Queiroz

Monografia de Graduação apresentada à
Coordenação de Engenharia da Computação,
UFAM, da Universidade Federal do Amazonas,
como parte dos requisitos necessários à
obtenção do título de Engenheiro da
Computação.

Orientadores:

Miguel Grimm do Nascimento

Manaus

Fevereiro de 2023

Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

Q3d Queiroz, Isadora Santarém de
Diário da Saúde: Sistema de registro de evoluções diárias com
comunicação Bluetooth / Isadora Santarém de Queiroz . 2023
41 f.: il. color; 31 cm.

Orientador: Miguel Grimm do Nascimento
TCC de Graduação (Engenharia da Computação) - Universidade
Federal do Amazonas.

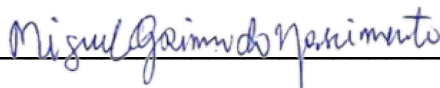
1. Android. 2. Aplicativo. 3. Saúde. 4. Registro. I. Nascimento,
Miguel Grimm do. II. Universidade Federal do Amazonas III. Título

DIÁRIO DA SAÚDE: SISTEMA DE REGISTRO DE EVOLUÇÕES DIÁRIAS COM
COMUNICAÇÃO BLUETOOTH

Isadora Santarém de Queiroz

MONOGRAFIA SUBMETIDA AO CORPO DOCENTE DO CURSO DE
ENGENHARIA DA COMPUTAÇÃO DA UNIVERSIDADE FEDERAL DO AMAZONAS
COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU
DE ENGENHEIRO.

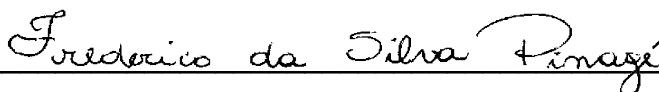
Aprovada por:



Prof. Miguel Grimm do Nascimento.



Prof. Thiago Brito Bezerra.



Prof. Frederico da Silva Pinage.

Manaus

Fevereiro de 2023

Agradecimentos

Primeiramente agradecer a Deus, que esteve a frente e guiou meus passos durante toda a trajetória até este momento.

Em segundo à minha família. Meu avô, Pedro, que esteve presente no começo desta jornada e certamente estará em espírito acompanhando até o final. Meus pais, Ianderley e Viviane, que sempre foram suporte para continuar focando nos meus estudos e profissional. Às minhas avós, Amélia e Pedrina, minha tia Irlandia, meus tios, Rubens e Juliano, que estavam presentes no meu dia-a-dia, facilitando muitas vezes. E às minhas irmãs, Isabel e Isis, principalmente, por me apoiarem e incentivarem independente dos planos pequenos ou grandes que eu elaborava.

Agradeço também aos meus professores que guiaram até o momento, e especialmente ao professor Miguel Grimm, pela orientação e paciência durante o processo deste trabalho.

Ao Fong, que tornou tudo isso possível e mais próximo, ao acreditar e incentivar minha educação fazendo-a mais acessível com meu primeiro notebook. À Carla Soares, por confiar no projeto e disponibilizar seu conhecimento. À Francilene Queiroz, que estava presente desde o começo da trajetória na UFAM e continua. Aos amigos, que sempre estiveram ao meu lado, tornando a caminhada mais leve e fácil de aproveitar enquanto progredimos.

Resumo da Monografia apresentada à UFAM como parte dos requisitos necessários para a obtenção do grau de Engenheiro

DIÁRIO DA SAÚDE: SISTEMA DE REGISTRO DE EVOLUÇÕES DIÁRIAS COM COMUNICAÇÃO BLUETOOTH

Isadora Santarém de Queiroz

Fevereiro/2023

Orientadores: Miguel Grimm do Nascimento

Curso: Engenharia da Computação

O registro de medidas médicas é uma parte importante e usual do processo de acompanhamento de um paciente, especialmente para idosos. No entanto, muitas vezes este monitoramento ainda é feito de modo manual por profissionais da área como médicos, enfermeiros e cuidadores. Deste modo, o objetivo principal deste trabalho é o desenvolvimento de um sistema para controle e armazenamento de uma medida enfatizada para o melhor gerenciamento. Realizando método quantitativo com utilização de um formulário com perguntas abertas e fechadas para obtenção de base para definições iniciais deste trabalho. Com este norte, o sistema contou com uma divisão entre hardware, com a montagem de um dispositivo eletrônico, e software, com um aplicativo para apresentação de forma gráfica e registro dos dados coletados.

Palavras-chave: Android, Aplicativo, Saúde, Idoso, Registro médico

Abstract of Monograph presented to UFAM as a partial fulfillment of the requirements for the degree of Engineer

HEALTHY DIARY: DAILY EVOLUTIONS RECORD SYSTEM WITH BLUETOOTH COMMUNICATION

Isadora Santarém de Queiroz

Fevereiro/2023

Advisors: Miguel Grimm do Nascimento

Course: Computer Engineering

Recording medical measurements is an essential and everyday part of the patient follow-up process, especially for the elderly. However, this monitoring is often done manually by professionals in the field as doctors, nurses, and caregivers. Thus, the main objective of this work is to implement a system for the control and storage of an emphasized measure for better management of records. Performing the quantitative method with a form with open and closed questions to obtain the basis for initial definitions of this work. With this north, the system had a division between hardware, with the assembly of an electronic device, and software, with an application for graphical presentation and recording of the collected data.

Keywords: Android, Application, Health, Elderly, Medical record

Sumário

Capítulo 1	9
Introdução	9
1.1. Justificativa	9
1.2. Objetivos	10
1.2.1. Objetivos Gerais	10
1.2.2. Objetivos Específicos	10
1.3. Organização da Tese	11
Capítulo 2	11
Fundamentação do Trabalho	11
2.1. Arduino	11
2.1.1. Sensor de Frequência Cardíaca	12
2.1.2. Módulo bluetooth	13
2.2. Android	14
2.3. Android Studio IDE	16
2.4. Kotlin	16
2.5. Firebase	17
2.6. MVVM	18
Capítulo 3	19
Desenvolvimento	19
3.1. Pesquisa	19
3.2. Hardware	23
3.2.1. Arduino	23
3.3. Campo de Interação	26
3.4. Software	29
3.4.1. Configuração do Firebase ao Projeto	29
3.4.2. Aplicativo Diário da Saúde	29
Capítulo 4	33
Conclusão	33
4.1. Sobre o Trabalho Realizado	34
4.2. Sobre Trabalhos Futuros	35
Referências Bibliográficas	38
Anexos	41
Anexo 1	41
Questionário	41

Lista de Figuras

Figura 1 - Arduino Uno	12
Figura 2 - Sensor de frequência cardíaca	13
Figura 3 - Módulo bluetooth HC-05	13
Figura 4 - Componentes Android	14
Figura 5 - Produtos Firebase	17
Figura 6 - Padrão MVVM	18
Figura 7 - Gráfico de registro de medidas	20
Figura 8 - Gráfico de registro digital	20
Figura 9 - Gráfico de uso passado de aplicativos	21
Figura 10 - Gráfico de uso futuro de aplicativos	21
Figura 11 - Gráfico de principais medidas	22
Figura 12 - Sistema do dispositivo eletrônico	23
Figura 13 - Configurações do Arduino	24
Figura 14 - Método principal do Arduino	24
Figura 15 - Função para leitura Bluetooth	25
Figura 16 - Métodos condicionais do Led	25
Figura 17 - Função de envio de dados	26
Figura 18 - Esquema de cores para design	26
Figura 19 - Fluxo de telas parte 1	27
Figura 20 - Fluxo de telas parte 2	27
Figura 21 - Mensagens apresentadas ao usuário	28
Figura 22 - Diretório do projeto no Android Studio	28
Figura 23 - Modelo de Dados do Paciente	29
Figura 24 - Diretório do projeto no Android Studio	30
Figura 25 - Modelo de Dados do Paciente	31
Figura 26 - Modelo de registro de datas	31
Figura 27 - Modelo do registro das medidas	31
Figura 28 - Comunicação do viewModel	32
Figura 29 - Fluxo geral de telas e serviços	33

Capítulo 1

Introdução

O envelhecimento populacional torna-se cada vez mais expressivo, trazendo para debate os pontos de vulnerabilidade que atingem a pessoa anciã. De acordo com o Instituto Brasileiro de Geografia e Estatística (IBGE) do ano de 2019, o número de pessoas acima de 60 anos no país supera o número de crianças até 9 anos^[1]. Coerente à base de que o aumento do número de idosos pode ser fundamentado pelo desempenho das taxas de natalidade e ainda das taxas de mortalidade. Dados do IBGE apontam que até 2050, cerca de 30% da população será de idoso no Brasil^[2].

Assim como citado na Caderneta de Saúde da Pessoa Idosa^[3], disponibilizada pelo Ministério da Saúde, o processo de envelhecimento é “um processo de alterações morfológicas, fisiológicas, bioquímicas e psicológicas que acarretam uma lentificação ou uma diminuição do desempenho do sistema orgânico e, conseqüentemente, uma diminuição da capacidade funcional.”, enfatizando que o termo capacidades funcionais utilizado refere-se como a habilidade de efetuar atividades físicas e mentais necessárias para manutenção de suas tarefas básicas.

Nestas condições, a medicina preventiva e o estímulo a projetos voltados para a saúde da terceira idade é indispensável. Isto é uma dificuldade que atinge além do sistema privado, também o sistema público de saúde, considerando o crescimento da procura por assistência especializada, sendo esta por vezes com alto custo.

1.1. Justificativa

Tendo em vista o declínio das funções físicas e cognitivas, sabe-se que idosos podem enfrentar diversas dificuldades no cotidiano. Neste ponto, o papel de um profissional da área de saúde pode ser importante.

Considerando, que tanto médicos, enfermeiros, quanto cuidadores, necessitam de um controle dos registros de um paciente, e a utilização, ainda frequente, de métodos manuais para estes apontamentos, este trabalho, debate sobre a alternativa mobile para a retenção das informações.

Atentando para os números de casos de idosos com doenças cardiovasculares, inicialmente, este projeto armazena e exibe somente entradas de frequências cardíacas. O aplicativo ainda abrange a possibilidade da criação de um arquivo no formato de planilha para estes dados, de modo a facilitar o envio dos mesmos para outras mídias ou transmitir a informação de modo mais rápido.

1.2. Objetivos

1.2.1. Objetivos Gerais

O projeto tem como objetivo geral a implementação de um aplicativo para o monitoramento e armazenamento de medidas de frequência cardíaca usualmente coletadas em diários de evolução do paciente, para o controle do cuidador profissional ou domiciliar. Assim como o desenvolvimento de um dispositivo eletrônico para a captação destas entradas.

1.2.2. Objetivos Específicos

O trabalho, divide-se em duas partes, na qual, o aplicativo fundamenta-se em atingir os requisitos funcionais e não-funcionais a seguir:

- a) A inserção de novos pacientes no banco de dados utilizando o CPF como identificador do mesmo;
- b) A consulta ao banco buscando dados de um paciente pelo identificador ;
- c) A conexão com um dispositivo bluetooth listado ou encontrado pelo smartphone;
- d) A obtenção das medidas em tempo de envio diretamente do dispositivo;
- e) O armazenamento automático das 3 medidas básicas ao desconectar um dispositivo;

- f) A possibilidade de baixar um arquivo em excel com as medidas de todos os pacientes registrados, ou somente do filtrado.

Enquanto a parte de hardware, tem como requisitos:

- a) A conexão com dispositivos bluetooth;
- b) A decodificação de batimentos cardíacos pelo sensor;
- c) O envio dos batimentos para um smartphone;
- d) A exibição de um alerta, por meio de um Led, para medidas não usuais.

1.3. Organização da Tese

A tese está organizada da seguinte maneira, este primeiro capítulo contém a justificativa e os objetivos do trabalho em questão, tanto geral quanto específicos. A partir do capítulo dois há a apresentação da Fundamentação teórica do trabalho, relatando as tecnologias pensadas para o projeto. No capítulo seguinte é descrito o desenvolvimento da tese, apresentando a metodologia, os diagramas e conceitos utilizados na evolução do trabalho. O capítulo quatro aponta os resultados obtidos com o capítulo anterior e também indica as possibilidades de continuação do projeto, por fim, temos as referências bibliográficas utilizadas para o contexto e base do projeto.

Capítulo 2

Fundamentação do Trabalho

2.1. Arduino

Criado em 2005 por uma equipe de pesquisadores, o Arduino tinha como propósito a criação de um dispositivo que pudesse ser acessível financeiramente e fácil de programar, tornando alcançável para estudantes e projetistas amadores. Constituído basicamente por dois componentes: a placa, ou seja, o hardware e a IDE Arduino, o software utilizado para escrita do código^[4].

Com a capacidade de interpretação de entradas e controle de saídas a plataforma assemelha-se ao processamento de um pequeno computador, com a

necessidade de programação. Esta programação é possível por meio do ambiente integrado de desenvolvimento próprio do Arduino, um software para escrita de código em uma linguagem semelhante a C/C++ compreensível para a placa^[5].

A plataforma possui uma variedade de modelos de placas com especificações próprias de hardware. Dentre as mais populares podemos citar Mega, Uno, Leonardo, Nano entre outras. A estrutura de todas atua de forma simples e prática, com as entradas e saídas sendo determinadas a nível de código, ainda sem a precisão de componentes externos.

A placa escolhida e utilizada para o projeto foi a UNO R3, apresentada na figura 1, pela praticidade e ampla distribuição da mesma para iniciantes na plataforma.



Figura 1 - Arduino Uno
Fonte: Arduino Datasheet

Conforme o manual de referência^[6], esta placa oferece um microcontrolador ATmega328 com clock de 16MHZ, 32KB de memória flash, 2KB de SRAM, 14 pinos digitais para escolha de entrada e saída, sendo 6 destes possíveis saídas *Pulse Width Modulation* (PWM), mais 6 pinos analógicos, botão de reset, entrada e saída serial, jack de alimentação 5V, dentre outras especificações. Ainda é possível mencionar que a conexão com o computador usa um cabo USB A/B, havendo chance de ser alimentado com uma fonte externa chaveada de 7 a 12 VDC.

Outro ponto característico da plataforma é a gama de módulos e shields para o Arduino, tal como acessórios que incrementam a variedade de aplicações do dispositivo, como conexão bluetooth, sensores, entre outros.

2.1.1. Sensor de Frequência Cardíaca

Dada a importância do conhecimento de dados de batimentos cardíacos, foi escolhido o sensor de pulso e frequência cardíaca para entrada principal dos registros neste trabalho. O sensor é um dispositivo óptico amplificado com capacidade de medir atividade elétrica do coração, sendo possível a inspeção da frequência cardíaca em diferentes partes do corpo humano, conforme definido em seu guia de iniciação^[7].



Figura 2 - Sensor de frequência cardíaca
Fonte: Pulse Sensor Guide

Estes dados obtidos pelo sensor da figura 2, onde podemos observar os três pinos, são encaminhados para o microcontrolador do Arduino por meio de um único pino de sinal. Com um consumo de energia de 4mA considerado baixo, é indicado para projetos embarcados e wearables.

Dentre as principais especificações do sensor podemos citar que a tensão de operação do mesmo varia entre 3V a 5V DC, tem um sensor óptico integrado, facilidade de conexão com o microcontrolador, e tamanho reduzido. Além disso, o sensor escolhido para o projeto conta com uma biblioteca open source própria para a interface de software do Arduino, com disponibilidade pelo site Github para o melhor uso do dispositivo pelos desenvolvedores.

2.1.2. Módulo bluetooth

Uma vez que a comunicação bluetooth é popular e já utilizada em vários dispositivos, como smartphones, tablets entre outros, é uma boa escolha para implementação de uma rede sem fio com Arduino^[9]. Esta rede é definida como *Personal Area Network* (PAN), e consiste em um protocolo para comunicação de rádio para uso próprio. O bluetooth utiliza a faixa de frequência de 2.4 a 2.485 GHz.

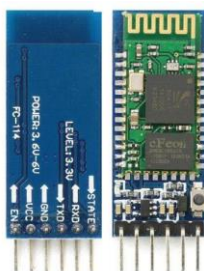


Figura 3 - Módulo bluetooth HC-05
Fonte: eletrogate

O módulo bluetooth RS232 HC-05 da figura 3 foi escolhido para este trabalho por disponibilizar de modo acessível e prático a comunicação necessária entre a placa e a aplicação Android. Este módulo suporta tanto o modo mestre, no qual há o envio das informações, quanto o escravo, onde há recepção de informações, até mesmo em ambos os modos simultaneamente, com uma configuração simplificada.

Conforme apresentado no manual^[11], atua com o protocolo bluetooth na versão 2.0 + EDR, simplificando a substituição de conexões seriais sem fio, finalidade para

qual é usado o perfil *Serial Port Profile* (SPP) do bluetooth. Como pertence a classe dois, que possui potência de 2,5mW, tem um alcance aproximado de 10 metros.

A alimentação deste módulo é de 5V, no entanto a tensão de comunicação é de 3.3V. Uma vez que o RX do Arduino recebe apenas sinais lógicos do HC-05 e o Arduino entende 3.3V como nível alto, não há a precisão de um divisor de tensão para obtenção deste valor em TX, qual conecta com RX do Arduino, somente no RX, que liga ao TX do Arduino.

2.2. Android

Para este projeto, o sistema operacional escolhido foi o Android, devido a popularidade e maior compatibilidade entre dispositivos com outros elementos escolhidos para compor o trabalho.

Criado em 2003 pela companhia Android Inc^[12], mas lançado para consumidores em 2008 pela Open Handset Alliance, uma aliança entre diversas companhias, dentre elas a Google, o sistema operacional Android abrange hoje um pouco mais de 71% do mercado móvel, notavelmente à frente do segundo colocado o sistema da Apple, iOS, que atinge cerca de 28%^[13]. É uma plataforma de código aberto, baseado no núcleo Linux, desse modo um código fonte disponível para a visualização, download, modificação, aprimoramento, redistribuição, diversas possibilidades de personalização^[14].

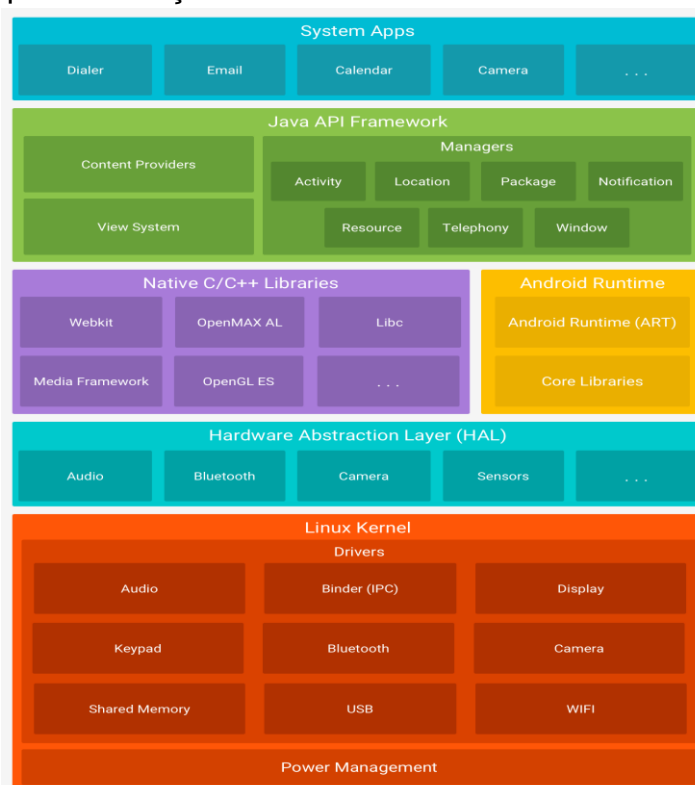


Figura 4 - Componentes Android
Fonte: Android Developers

Segundo site oficial do Android^[15], as atualizações do sistema tendem a ser nomeadas com nomes de doces, como Cupcake, KitKat entre outros, tendo sendo

lançadas 24 versões desde o lançamento, onde a versão mais recente é a 13. Dentre os componentes do Android, podemos destacar as camadas da figura 4 acima.

Como citado anteriormente, a base da plataforma Android é o kernel do Linux. Sendo este executor de tarefas de baixo nível, mais adjacentes do hardware do dispositivo, tal qual encadeamento e gerenciamento de memória. Devido a isso, é possível o aproveitamento de recursos de segurança principais e deste modo, permite aos fabricantes de dispositivos desenvolver drivers de hardware para um kernel regular.

Segundo também, a documentação oficial seguida por desenvolvedores^[17], em seguida, temos a camada de abstração de hardware (HAL) que se compõe de módulos de bibliotecas, que implementam uma interface para um tipo de componente de hardware. Exemplificando, no caso de um usuário realizar uma chamada ao hardware para recepção de arquivo por bluetooth, o sistema Android carrega o módulo da biblioteca para este hardware executando uma interface para este modelo específico de componente. Estas interfaces padrões exibem as capacidades de hardware do dispositivo para a estrutura da Java API de maior nível.

O Android Runtime (ART) é um ambiente de tempo de execução controlado para dispositivos com versão 5.0 ou mais, onde cada aplicativo carrega o próprio processo com uma instância própria do ART. Planejado para operar diversas máquinas virtuais em dispositivos de baixa memória executando arquivos DEX, sendo este um formato de bytecode específico para Android, otimizado para disponibilizar um consumo mínimo de memória. Dentre os recursos é possível citar coletor de lixo otimizado e compilação 'ahead-of-time' (AOT) e 'just-in-time' (JIT). Também serve para facilitação da depuração, criação de relatórios de erros e diagnósticos.

Tanto ART quanto HAL tem vários componentes e serviços desenvolvidos de forma nativa em C e C++. O conjunto de bibliotecas nativas, elaboradas para variados objetivos, é acessível pelas APIs de alto nível escritas em linguagem Java. Por exemplo, existe a possibilidade de acessar OpenGL ES, uma API gráfica destinada a dispositivos encoados, pela Java OpenGL API da estrutura do Android para acrescentar a condição de desenhar e manipular gráficos 2D e 3D na aplicação.

A API Java do sistema abrange os componentes necessários para criação de aplicativos usando serviços e acessando recursos do Android, de modo a facilitar a reutilização destes. Implementa diversas funcionalidades tais como: um sistema de visualização para programação da IU de um app, um gerenciador de recursos que disponibiliza acesso a recursos sem código, um gerenciador de notificações para exibição de alertas personalizados por app, um gerenciador de atividades coordenando o ciclo de vida dos aplicativos e ainda provedores de conteúdo para interação entre aplicações.

Como último nível, o sistema operacional apresenta um conjunto de aplicativos principais para envio de mensagens, calendários, e-mail entre outros. Estas aplicações já presentes na plataforma não têm destaque entre os aplicativos instalados posteriormente, de modo que um aplicativo terceirizado pode, por exemplo, tornar-se até mesmo o teclado padrão do usuário. Desta forma, aplicativos do sistema

atuam como aplicativos para usuários e fornecem capacidades principais, que os desenvolvedores podem obter pelos próprios aplicativos.

2.3. Android Studio IDE

Disponibilizado gratuitamente pela Google, o Android Studio é um *Integrated Development Environment* (IDE), ou ambiente de desenvolvimento integrado, baseado no IntelliJ IDEA. Há outras alternativas para criação de aplicativos mobile, como o Eclipse ou o Xamarin. No entanto, o Android Studio, principalmente se comparado ao Eclipse, possui maior variedade de personalização e de recursos, além de menor recorrência de bugs. Dentre os recursos oferecidos para este aumento de produtividade temos: o sistema de compilação flexível baseado em Gradle, frameworks e ferramentas de teste com muitas possibilidades, compatibilidade com C++ e NDK, e vários outros, segundo Júnior^[18].

A plataforma apresenta diferentes tipos de módulo, exibindo por padrão, os arquivos do módulo de apps Android, para facilitar o acesso ao usuário. Este módulo abrange as pastas: *manifests*, com o arquivo *AndroidManifest*, Java, qual contém os arquivos de código e testes, e recursos, abrangendo arquivos como layouts XML, string, imagens e todos outros, conforme apresentado no site oficial^[19].

Juntamente com recursos típicos do Android concedidos pelo plug-in do Android para Gradle, o Android Studio usa o Gradle como sistema de compilação de base. Conforme definido no site oficial para desenvolvedores Android Developers “Esse sistema de compilação é executado como uma ferramenta integrada no menu do Android Studio e de forma independente na linha de comando.”

Como o Gradle possui versatilidade, não é necessária a alteração de arquivos de origem principais do app. Com arquivos de texto simples, que utilizam a sintaxe do Groovy, é possível a configuração de elementos disponibilizados pelo plug-in citado anteriormente. De modo que se um projeto já existente for importado na plataforma, ela diretamente gera os arquivos de compilação necessários.

2.4. Kotlin

Desenvolvido em 2011 pela JetBrains e patrocinado pela Google, Kotlin é uma linguagem de programação de uso geral e estaticamente tipada, com código aberto e em conformidade com a programação orientada a objetos. Pelo patrocínio e características citadas anteriormente, é compatível com o desenvolvimento do Google para Android^[21].

Define-se uma linguagem estaticamente tipada como uma linguagem que o tipo de cada variável e expressão é identificado em tempo de compilação. Também é válido citar que há redução de código ao optar por Kotlin e tem interoperabilidade com Java, uma vez que é executado em *Java Virtual Machine* (JVM).

Segundo a documentação oficial da linguagem^[22], dentre as vantagens do Kotlin é possível apontar que é multiplataforma, ou seja, abrange todos os IDEs de

Java, também é mais seguro que o próprio Java, já que foi desenvolvido para remover referências de ponteiro nulo e facilitar o tratamento de valores nulos. A facilidade para aprender, pela semelhança com Java, também é um ponto a ser considerado, além do uso dos frameworks e bibliotecas Java em projetos Kotlin, sem a precisão da modificação de todo o projeto.

Como mencionado, Kotlin pode ser utilizado na criação de aplicativos Android, mas também pode compilar para JavaScript e ter a viabilização para o frontend, assim como foi pensado para trabalhar bem para o desenvolvimento Web e o desenvolvimento do lado do servidor. Por essas razões, muitas empresas como Uber, Evernote e Pinterest já incorporaram a tecnologia.

2.5. Firebase

Lançada inicialmente em 2011 pela startup de James Tamplin e Andrew Lee, a plataforma só foi adquirida pela Google em 2014, quando ganhou força. Segundo Kumar S^[23], o Firebase é uma plataforma digital caracterizada como um programa de banco de dados NoSQL, com armazenamento de dados. Abrangendo diversos serviços, ele é comumente aplicado para desenvolvimento de aplicativos mobile ou web, sendo assim multiplataforma, e considerado mais prático, suportando diversas linguagens, como Java, Swift, Node.js, entre outras. Não somente na área de desenvolvimento, é possível utilizar no gerenciamento e monetização de produtos, com a utilização do Admob.

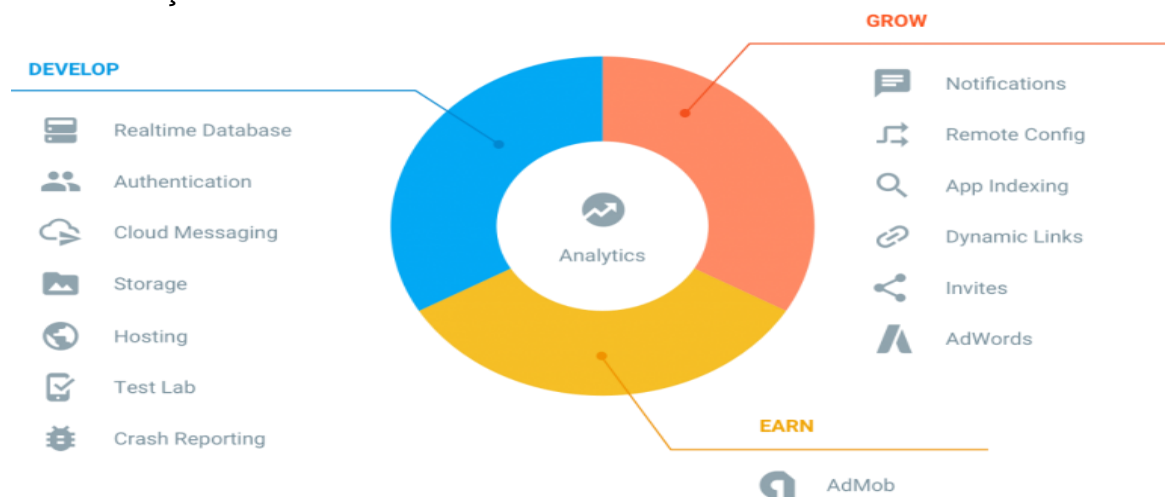


Figura 5 - Produtos Firebase
Fonte: Google Developers

Conforme a página oficial de desenvolvimento da Google para desenvolvedores^[24], destacando as vantagens principais do produto exibidos na figura 5, temos o fato de ser uma API já definida, com dados em tempo real e armazenados pelo *Google Cloud Storage*. Concede mais segurança, com autenticação via e-mail, Google, Github dentre outros, tornando aplicativos altamente escaláveis. Já como

desvantagens, é possível mencionar alguns problemas com relação *UI/UX*, costumeiros bugs de atualizações e restrições de infraestrutura.

Dentre os variados serviços que pode oferecer é possível evidenciar os principais utilizados para este trabalho: o *Realtime Database* que conceitua-se, segundo a documentação oficial do Firebase^[25], como um banco de dados que tem probabilidade de sincronização com dados em tempo real, com a possibilidade de configuração das regras de segurança para definição de perfis de acesso. Também ressaltar o *Authentication*, que é o serviço que proporciona a autenticação por meio de contas do Google, Github, Facebook entre outras. Por fim, o *Storage* é um serviço de armazenamento de dados do usuário por meio de uma conexão estável, com a possibilidade de compartilhamento dos mesmos.

2.6. MVVM

O padrão de software do projeto *Model-View-ViewModel* (MVVM) criado por John Gossman em 2005, com a desassociação da lógica de apresentação de dados da seção principal da lógica de negócios do aplicativo^[26]. As camadas principais deste padrão são a Camada de apresentação, o Modelo e a ViewModel, como apresentado na figura 6.

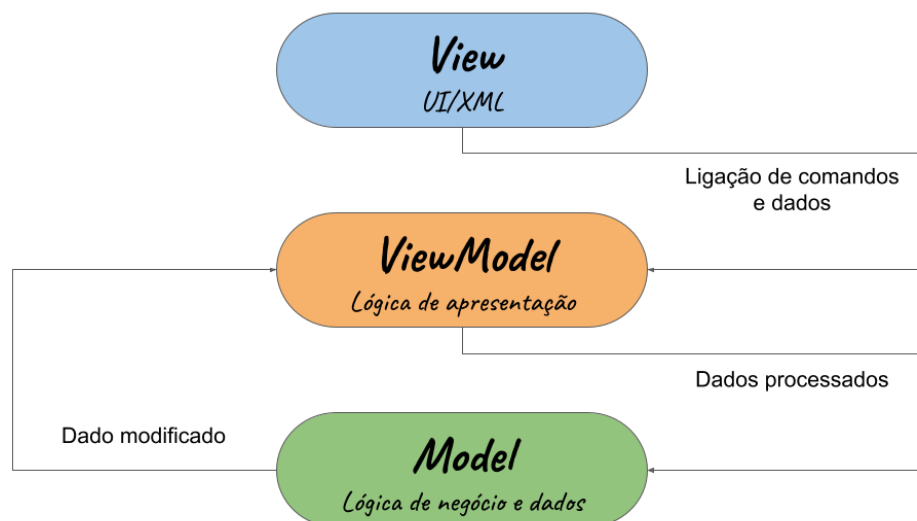


Figura 6 - Padrão MVVM

Segundo Barbosa (2022)^[27], a *View* tem como finalidade a comunicação com o *ViewModel* sobre ações do usuário, ela observa o *ViewModel*, mas não abrange nenhuma lógica de aplicação. Quanto ao *ViewModel*, esta camada apresenta os fluxos de dados, servindo como ligação entre o modelo e a camada de apresentação. Na camada de *Model* é realizada a abstração das fontes de dados, trabalhando em conjunto com a *ViewModel* para obtenção e armazenamento de dados.

O padrão não apresenta acoplamento rígido entre a visualização e o modelo, além de não haver interface entre *View* e *Model*, outras vantagens seriam a

praticidade para testes unitários e um código orientado a eventos. Dentre as desvantagens é destacado a possibilidade de código de tamanho excessivo, além da necessidade de criar observáveis para componentes da interface.

O MVVM é similar com o padrão *Model-View-Presenter* (MVP), com o *Presenter* tendo a mesma função que a atribuída ao *ViewModel*, mas com melhorias citadas sendo empregadas, supera as desvantagens presentes no MVP. Além disso, recursos como *commands*, *databinding* e eventos roteados, também trazem ao MVVM a organização e estrutura esperada pelos desenvolvedores.

Capítulo 3

Desenvolvimento

As etapas para criação do projeto consistiram em pesquisa com profissionais envolvidos na área de interesse, definição de padrões do projeto, criação das telas do aplicativo, desenvolvimento do código para comunicação com a placa Arduino, configuração da plataforma Android Studio, desenvolvimento do código para o dispositivo Android, testes com emuladores e dispositivos reais. Será abordado cada ponto relatado para detalhamento maior entre os tópicos seguintes.

3.1. Pesquisa

A primeira etapa foi a elaboração de um questionário, apresentado no anexo 1, para compreensão da problemática utilizando o método quantitativo, com aplicação de um questionário com perguntas abertas e fechadas, durante o período de 22 de janeiro de 2022 a 16 de fevereiro do mesmo ano. A escolha do método deu-se pela avaliação das características, como citadas no trabalho de Terence e Filho (2006)^[28] enfatizando o cumprimento de uma meta pré-estabelecida para enumeração ou mensuração de um evento, com o uso de dados compondo uma população específica, ou amostra, com a qual há generalização dos resultados, dentre outros fatores.

O formulário online aplicado utilizando a plataforma Google Forms continha 13 (treze) perguntas para profissionais da área de saúde aplicado dentre funcionários de uma clínica com ênfase em *homecare* e atendimentos psicológicos. Ao todo houveram 10 (dez) profissionais de saúde que responderam ao questionário. Os profissionais que responderam eram: 1 (um) enfermeira, 1 (um) técnico em análises clínicas, 6 (seis) técnicos em enfermagem, 1 (um) técnico de patologias e 1 (um) acadêmico de enfermagem. O questionário tinha ênfase em questões relacionadas a registro e acompanhamento de medidas geralmente utilizadas.

Como é realizado o registro de medidas

- Manualmente, escrevo tudo do modo mais organizado que consigo.
- Digitalmente, mas costumo utilizar mais o celular para anotações.
- Digitalmente, uso planilhas pelo notebook para organizar.

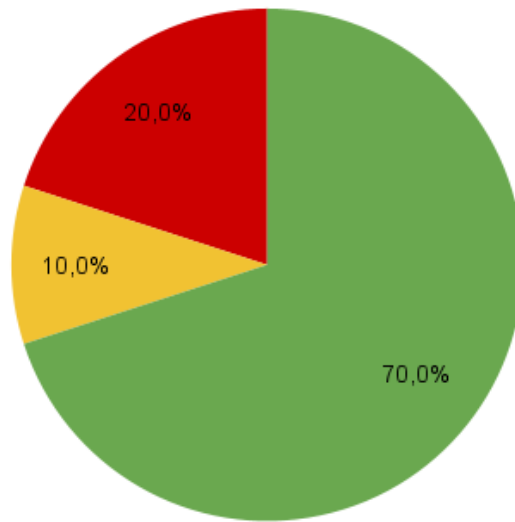


Figura 7 - Gráfico de registro de medidas

Necessidade de registro digital?

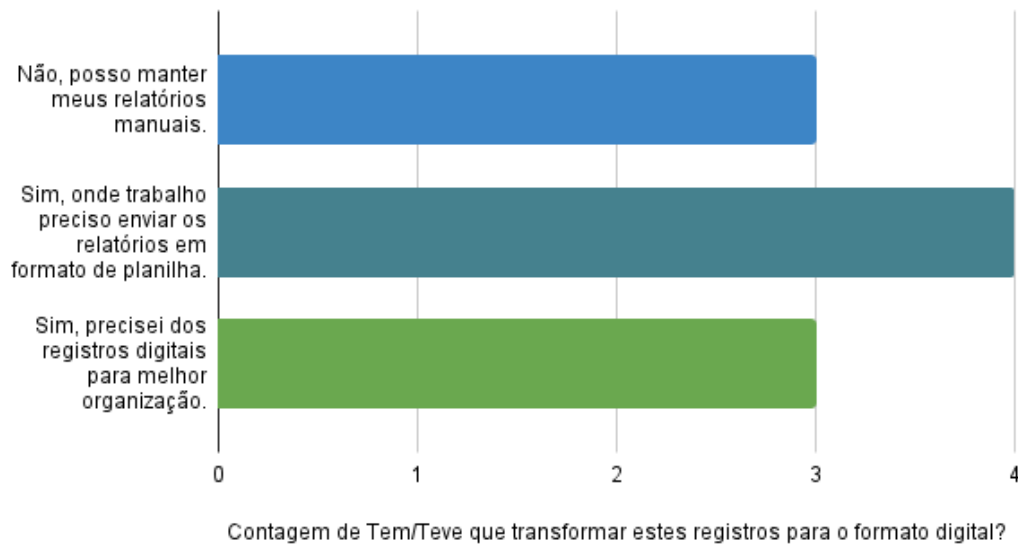


Figura 8 - Gráfico de registro digital

Os gráficos apresentados nas figuras 7 e 8 trazem dois pontos importantes para análise. No primeiro gráfico, cerca de 70% dos profissionais ainda opta pelo registro manual de medidas, enquanto 20% utiliza planilhas para organização e 10% escolhe o celular para estes registros. Enquanto isso, somente 30% podem e optam por manter seus relatórios de modo manual, os outros 70% precisam da digitalização. Portanto, ponderando-se assim, que é realizando um retrabalho durante o processo, que poderia ser evitado.

Utilização passada de aplicativos para registro

- Não, ainda não utilizo o celular para fazer os registros
- Sim, sempre procuro alguns para facilitar

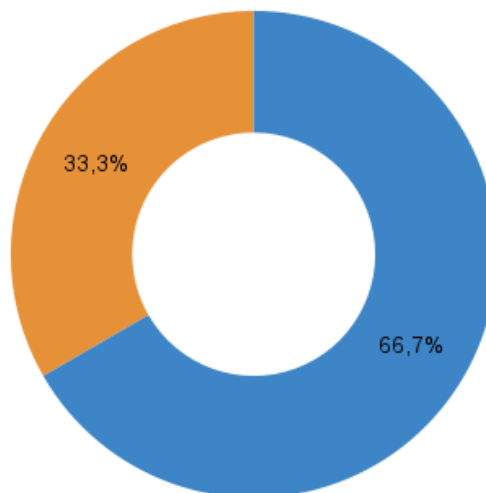


Figura 9 - Gráfico de uso passado de aplicativos

Uso futuro de aplicativo para registro médico?

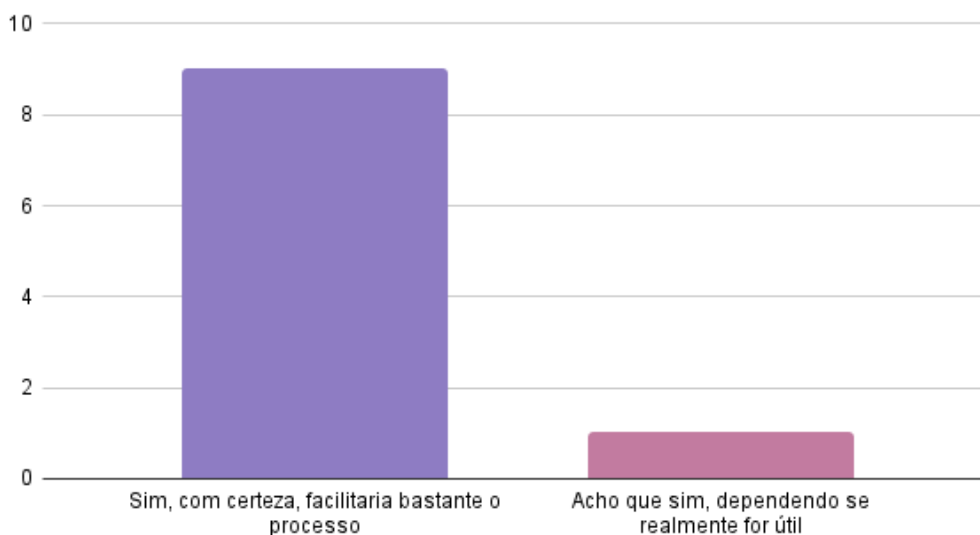


Figura 10 - Gráfico de uso futuro de aplicativos

Os gráficos acima, nas figuras 9 e 10, apresentam a utilização de aplicativos por profissionais. No primeiro gráfico, três atuantes da área declararam que procuram utilizar aplicativos para estes registros com certa frequência, enquanto sete relatam nunca terem utilizado aplicações do tipo. Já no segundo, indaga-se a possibilidade de uso futuro entre o meio, destacando-se que entre as opções de resposta havia possibilidade de resposta “Não”, “Acho que não, já possuo meu meio de organizar”, mas estas não foram escolhidas por nenhum representante, deste modo dividiu-se 90% para a certeza do uso e 10% para a possibilidade de uso, caso houvesse utilidade. Deste modo, sendo reiterado a existência de um público-alvo para um software que facilite as questões cotidianas dos profissionais.

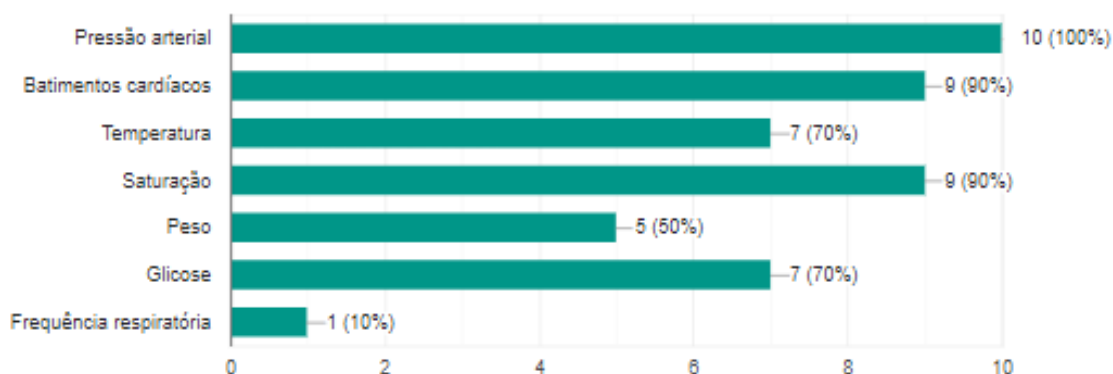


Figura 11 - Gráfico de principais medidas

Na figura 11, também foram listadas algumas medidas consideradas essenciais, com múltipla escolha para a amostra. Entre as medidas destacam-se a pressão arterial, batimentos cardíacos e saturação do oxigênio. Destas, foi analisado qual seria o modo ideal para iniciar este registro de medidas e optou-se por batimentos cardíacos considerando as pesquisas feitas com ênfase em idosos.

A partir destes dados, foi analisado qual parâmetro poderia ser mais adequado para a primeira versão do trabalho. Durante a pesquisa foi constatado que as doenças cardiovasculares acometeram mais de 230 mil brasileiros, segundo a CNN Brasil ^[29] em matéria lançada em 2021. Ainda segundo a mesma fonte, em todo território nacional, cerca de 14 milhões de pessoas apresentam alguma doença cardiovascular, e no mínimo aproximadamente 30% da população falece decorrente disto.

Também segundo o Boletim Epidemiológico da Secretaria de Saúde (2022)^[30], desde os anos 2000 doenças do aparelho circulatório, como as cardiovasculares, estão em primeiro lugar nos rankings de mortalidade no Brasil, assim como segundo a Organização Mundial de Saúde (OMS)^[31], doenças cardiovasculares também ocupam primeiro lugar no mundo em 2019. E de acordo com a Sociedade de Cardiologia^[32], é estimado que aconteça um aumento de cerca de 250% de doenças cardiovasculares, até 2040.

Deste modo, avaliando os dados de óbitos assim como de enfermos citados acima, optou-se pela escolha do armazenamento de batimentos cardíacos como acompanhamento principal para esta primeira fase do trabalho.

3.2. Hardware

Para o dispositivo físico foram usados os equipamentos já mencionados no capítulo de fundamentação teórica, como o sensor, placa Arduino, módulo Bluetooth e afins. A montagem final do dispositivo físico concluiu-se do modo abaixo na figura 12.

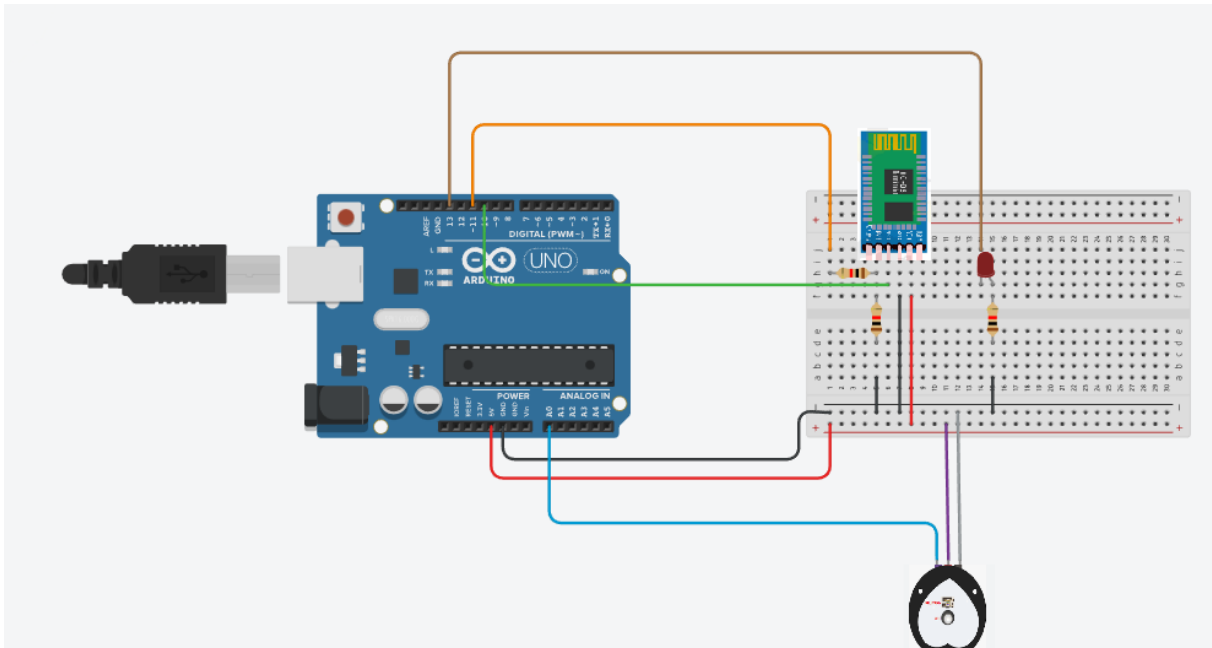


Figura 12 - Sistema do dispositivo eletrônico

Os resistores presentes no circuito são de $1K\Omega$ para evitar picos de tensão e favorecer uma proteção a mais ao sistema. O Led vermelho vem como um informativo visual para o paciente que o sensor pode ter captado uma frequência abaixo ou acima do esperado. O sensor conforme já citado tem a ligação com o VCC, o GND e a emissão de sinal que se comunica com o Arduino. Quanto ao módulo Bluetooth, temos também a conexão VCC com a fonte 3.3V emitida pelo Arduino, o Terra, além da comunicação Tx e Rx que se comunica diretamente com o Rx, Tx do Arduino respectivamente.

3.2.1. Arduino

É necessário destacar que o sensor de frequência escolhido já possui um código para utilização, disponibilizado no Github *WorldFamousElectronics* ^[33] do fabricante, sendo essenciais ajustes para melhor compatibilidade com o uso que o desenvolvedor almeja.

A atuação do Arduino no trabalho, dá-se principalmente na IDE do mesmo. Primeiramente, na figura 13, mencionando as ligações externas, a placa conecta-se com o Bluetooth, com o sensor e com o Led.


```

#define MAX_SENIOR 100
#define MIN_SENIOR 50
#define MAX_UNDER 100
#define MIN_UNDER 60

SoftwareSerial HC05(10,11);

const int pinLed = 13;
const int pulsePin = A0;
int replyByte;
int counterTimer = 0;
int age;

volatile int BPM;
volatile int Signal;
volatile int IBI = 600;
volatile boolean Pulse = false;
volatile boolean HB = false;

static int outputType = PROCESSING_VISUALIZER;

void setup() {
  Serial.begin(115200);
  interruptSetup();
  HC05.begin(9600);
  pinMode(pinLed, OUTPUT);
  digitalWrite(pinLed, LOW);
}

```

Figura 13 - Configurações do Arduino

```

void loop() {
  while (HB == true){ // Encontrou batida
    serialOutputWhenBeatHappens(); // Envia batida para Serial
    HB = false; // Reseta a Flag para proxima
    readAgeByBluetooth();
    if (age>59){
      setLedByValueSenior(BPM);
    } else {
      setLedByValueUnder(BPM);
    }
    delay(3000); // Realiza a leitura entre 3 segundos
  }
}

```

Figura 14 - Método principal do Arduino

Na parte do código acima na figura 14, temos a validação de batidas cardíacas em loop, fazendo a impressão da mesma no monitor serial apenas para verificação. O método *readAgeByBluetooth* atua na leitura enviada do app para realizar o condicional na função principal.

```

void readAgeByBluetooth(){
  while(HC05.available()){
    character = HC05.read();
    message.concat(character);
    if(character == '#'){
      message.remove(lastIndex);
      Serial.println("idade");
      Serial.println(message);
      age = message.toInt();
      message = "";
    }
  }
}

```

Figura 15 - Função para leitura Bluetooth

O método de leitura de idade, como mencionado na figura 15 acima, capta a leitura recebida pelo módulo HC05. Os caracteres são lidos e concatenados a uma mensagem, até a chegada do caractere '#' indicando o final da mensagem enviada pelo aplicativo. Assim, retira-se o último caractere, o identificador, e converte-se a mensagem para um parâmetro inteiro para a idade, e limpa a mensagem para aguardar novo recebimento.

```

void setLedByValueSenior(int value){

  if(value > MAX_SENIOR){
    digitalWrite(pinLed,HIGH);
  } else if (value < MAX_SENIOR && value >MIN_SENIOR){
    digitalWrite(pinLed,LOW);
  } else if(value > 0 && value < MIN_SENIOR){
    digitalWrite(pinLed,HIGH);
  } else {
    digitalWrite(pinLed,LOW);
  }
  sendDataToBluetooth();
}

void setLedByValueUnder(int value){

  if(value > MAX_UNDER){
    digitalWrite(pinLed,HIGH);
  } else if (value < MAX_UNDER && value > MIN_UNDER){
    digitalWrite(pinLed,LOW);
  } else if(value > 0 && value < MIN_UNDER){
    digitalWrite(pinLed,HIGH);
  } else {
    digitalWrite(pinLed,LOW);
  }
  sendDataToBluetooth();
}

```

Figura 16 - Métodos condicionais do Led

Os parâmetros de idade foram baseados nos estabelecidos pela Sociedade de Cardiologia após análise dos gráficos, para valores que compreendessem mais e

estivessem mais alinhados, estando assim definida uma mínima de 55 e máxima de 84 para o intervalo de idade de 18 a 59 anos, enquanto temos mínima 57 e máxima 84 para maiores de 60 anos, como exibido na figura 16.

Todas as medidas são enviadas ao aplicativo para ser realizada uma média, mas as idades atuam substancialmente no condicional do led para alertar o paciente.

```
void sendDataToBluetooth() {  
    if (HC05.available() > 0) {  
        HC05.println(BPM);  
    }  
}
```

Figura 17 - Função de envio de dados

E por último na figura 17, a codificação faz o envio por meio do módulo Bluetooth, após verificação de disponibilidade do mesmo, para a aplicação no dispositivo móvel.

3.3. Campo de Interação

Baseado na associação comum que liga a cor verde à área da saúde e hospitais, conforme relatado por Noronha^[34], foi realizado o padrão de cores da figura 18 abaixo para o aplicativo.

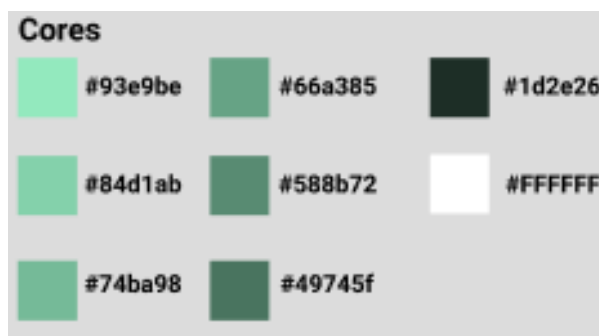


Figura 18 - Esquema de cores para design

Assim como ao elaborar as telas considerando o público-alvo a fonte escolhida foi a Playlist Script pela possível boa compreensão, assim como manteve-se a Roboto, pelo padrão do sistema Android. O tamanho da fonte também foi influenciado pelo mesmo motivo, procurando atender a isto com os tamanhos variando entre 32sp e 18sp.

O fluxo foi pensado para ser enxuto e de fácil acesso. Então colocou-se os dois fluxos principais como escolha da primeira tela. Deste modo, os fluxos possíveis são apresentados a seguir na ordem das telas, nas figuras 19 e 20.



Figura 19 - Fluxo de telas parte 1

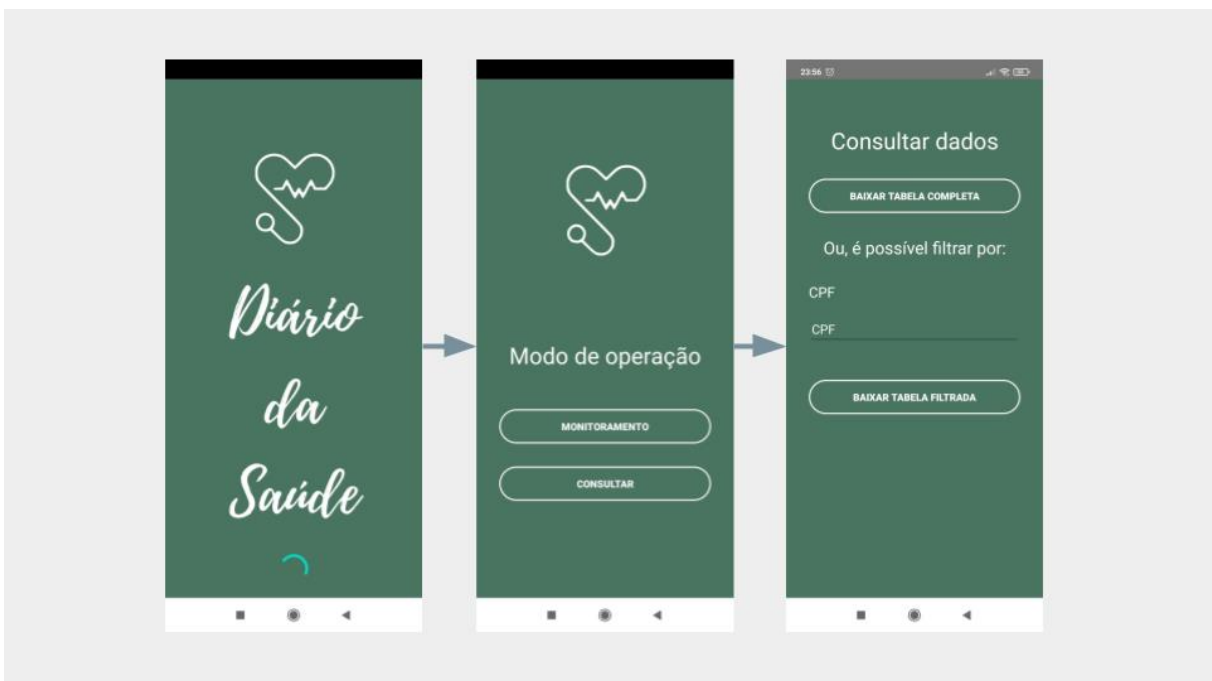


Figura 20 - Fluxo de telas parte 2

Há ainda apresentações de mensagens, para auxiliar o usuário qual procedimento está sendo realizado naquele momento, e assim justificar alguma intercorrência ou espera necessária. Abaixo algumas das principais mensagens notificadas, na figura 21.

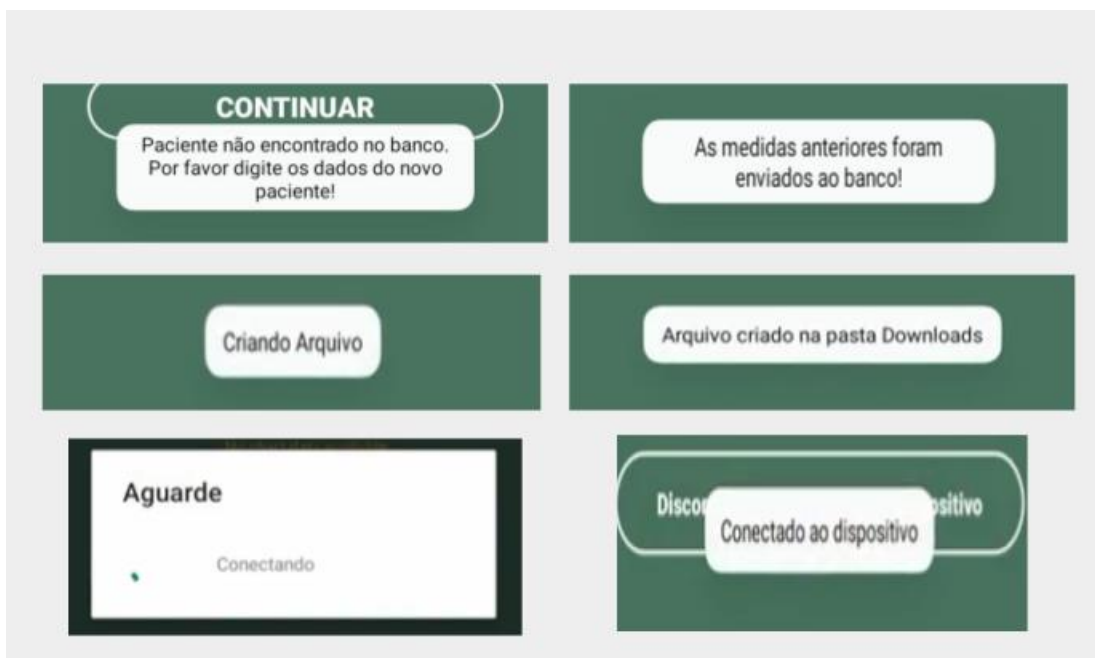


Figura 21 - Mensagens apresentadas ao usuário

Apresenta-se de modo abrangente as entradas e saídas do sistema nas figuras 22 e 23. Para o dispositivo eletrônico, ou hardware, temos o sinal de frequência captado pelo sensor e dados do paciente recebidos pelo bluetooth, e assim envia os valores de frequência em batimentos por minuto (BPM) e também emite o sinal para alerta no LED. Para o aplicativo Android, tem a recepção dos valores de batimentos cardíacos que vem do dispositivo eletrônico, assim como as entradas de modo manual dos dados de paciente, enquanto apresenta em gráfico os valores recebidos, o envio dos dados necessários por Bluetooth para o Arduino e ao encerrar uma sessão há o envio de dados para o Firebase.



Figura 22 - Mensagens apresentadas ao usuário



Figura 23 - Mensagens apresentadas ao usuário

3.4. Software

3.4.1. Configuração do Firebase ao Projeto

Uma vez que um projeto do Firebase atua como container para todos os aplicativos, recursos e serviços a serem utilizados. Deste modo os aplicativos ligados ao mesmo contentor “compartilham e têm acesso aos mesmos recursos e serviços provisionados para o projeto” como indicado na Documentação Oficial do Firebase, pela Google^[35]. Para utilizar o Firebase no projeto é necessário que o aplicativo opere na versão de API no nível 19 ou superior, ou no Android 4.4 ou superior.

O primeiro passo é a criação de um projeto do Firebase no console do mesmo, seguindo os fluxos simples, até o ponto de registro do aplicativo. Para uso deste projeto no aplicativo, na página de visão geral é necessário ‘adicionar’ o aplicativo ao projeto, com a atenção de digitar o nome do pacote conforme o pacote criado no Android Studio.

Após o registro, na área de Android das configurações de projeto, será possível baixar o arquivo ‘google-services.json’, o qual após completo o *download*, deverá ser movido para o diretório raiz do módulo do aplicativo criado pela IDE. Para tornar comunicável o arquivo recentemente adicionado aos SDKS, também é adicionado ao arquivo *gradle* de nível de projeto, o *plug-in* de serviços da Google, como uma dependência *buildscript*, do mesmo modo, também ao *gradle* de nível de app. E por fim, no *gradle* de nível aplicativo, são adicionadas as dependências dos produtos Firebases que serão utilizados.

3.4.2. Aplicativo Diário da Saúde

Para o projeto foi selecionada a versão Android 7.0, API 24 por abranger cerca de 91,7% de usuários do sistema. Para configuração do projeto, dentre as principais bibliotecas utilizadas como dependência é possível citar, conforme já citado, ligadas aos serviços do Google e do Firebase, *MPAndroidChart*, responsável pela exibição

dos gráficos no app e *Poi-HSSF*, que auxiliou na criação de arquivos em excel dos dados baixados, dentre outras.

O nome do pacote foi escolhido anteriormente à decisão de nome do projeto, mas com a compatibilidade com o projeto criado no Firebase. Para a estrutura foi escolhido o padrão de arquitetura MVVM, sendo assim organizadas as pastas, conforme a figura 24 abaixo.

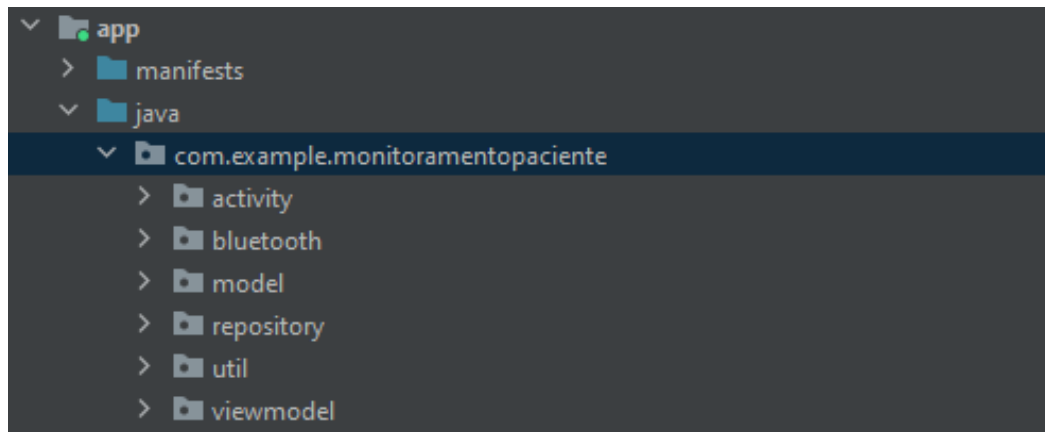


Figura 24 - Diretório do projeto no Android Studio

Enquanto em usualmente na programação é utilizado o método `main()`, o sistema Android inicia o código em uma instância `Activity`, na qual, segundo documentação do Android ^[36], é possível invocar funções equivalentes a estágios específicos do ciclo de vida de um aplicativo. Dessa forma, a pasta `activity` armazena arquivos correspondentes à classe `Activity`, que trazem a interação inicial de um usuário com o app, sendo assim nossa `View`, no sistema MVVM.

Para o `bluetooth`, foram separadas classes para cada funcionalidade a ser realizada, como conexão, desconexão, leitura e envio de dados, sendo necessário uso de algumas variáveis em *companion object* para comunicação entre diferentes pacotes. Essencial destacar também que foi utilizado o ID padrão de porta serial para a conexão entre dispositivo e módulo HC-05. Também é necessário que permissões a serem requisitadas estejam configuradas no *Android Manifest* do projeto.

Na pasta `model`, temos as classes de modelos utilizados para criação dos objetos que serão orientados durante o decorrer do aplicativo. A classe *PatientData*, na figura 25, guarda os principais dados do paciente a serem utilizados, utilizando o CPF como identificador único e no formato `String`, temos o nome, idade, gênero e a lista de objetos de registros por dia.

```

data class PatientData(
    var CPF: String?,
    var name:String?,
    var age: Int?,
    var gender: Boolean?,
    var dateRegister: ArrayList<MeasureDate>?,
) : Serializable

```

Figura 25 - Modelo de Dados do Paciente

Já a classe *MeasureDate*, figura 26, é voltada para a identificação da data, sendo esta definida como String para uma melhor organização do mesmo dentro do sistema Firebase, e a lista que serão armazenadas naquela data específica, sendo estas definidas pelo o outro objeto.

```

data class MeasureDate(
    var dateRegister: String?,
    var listBeats: MeasureHeartBeats
):Serializable

```

Figura 26 - Modelo de registro de datas

E por último arquivo da pasta *model*, temos o *MeasureHeartBeats*, figura 27, que traz as três medidas armazenadas por dia, sendo estas a média de todas as medidas armazenadas, a maior frequência registrada e a menor também, para o profissional estar ciente das mesmas.

```

data class MeasureHeartBeats(
    var average: Int?,
    var great: Int?,
    var minor: Int?
):Serializable

```

Figura 27 - Modelo do registro das medidas

O repositório gerencia o consumo de dados independente da origem deles, como banco local ou servidor online, no entanto, somente uma destas fontes por vez.

Na pasta *repository*, temos a classe que comunica diretamente com o realtime database do Firebase, tendo assim, funções como criação de pacientes, obtenção de lista, envio de dados e outras. Além disto também tem uma sealed class *ResultCall*, para mapear entre sucesso, progresso e falha de cada tentativa de chamada efetuada.

A pasta *util* armazena a classe com métodos utilizados entre outros pacotes. É onde é realizada a criação dos *SharedPreferences*, assim como o armazenamento e limpeza destes. Também contém formatações necessárias de dados e os cálculos de média em uma lista numérica, assim como descobrir o maior e menor valor nesta, para assim salvar estes dados e deixá-los prontos para envio ao banco.

No *viewmodel* temos uma parte lógica da interface, como citado a relação entre a *view* e o modelo, com a lógica de negócios. Esta classe também comunica-se com o repositório, como apresentado na figura 28, por meio de solicitações e tem retornos de chamadas observadas para envio de respostas para as *activities*.

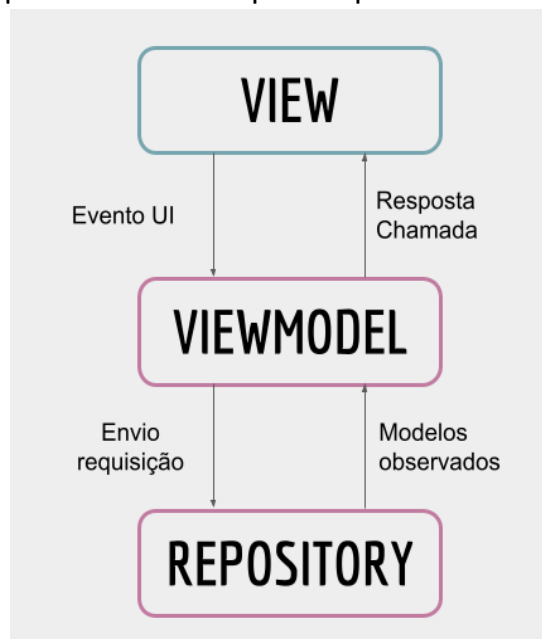


Figura 28 - Comunicação do *viewModel* do projeto

Deste modo, para resumir a atuação do aplicativo na sua versão final, a figura 29 abaixo apresenta as principais telas e principais operações possíveis de serem realizadas nestas.

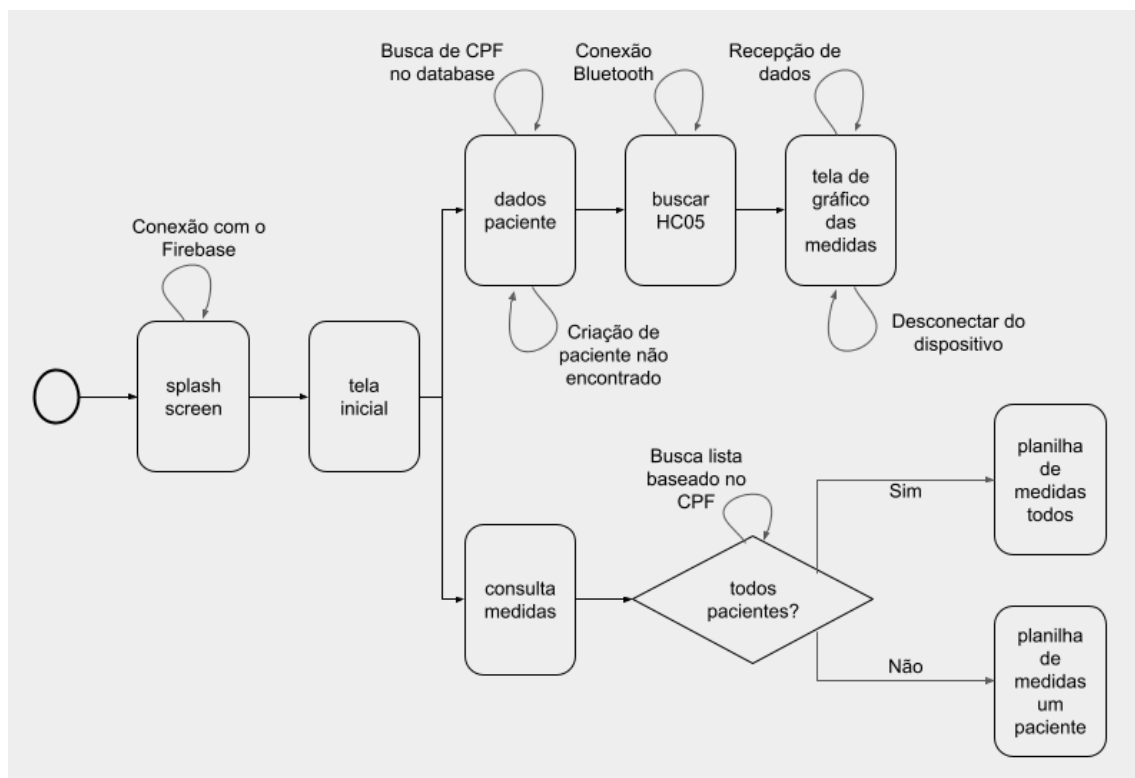


Figura 29 - Fluxo geral de telas e serviços do sistema

A atuação do dispositivo em conjunto com o aplicativo foi pensada prioritariamente para o acompanhamento domiciliar dos pacientes que utilizam serviços de profissionais de saúde. O dispositivo eletrônico estaria ligado de modo fixo ao paciente, com o sensor de frequência conectado em algum dos pontos comuns para medição dos batimentos, como pulso, dedos ou mesmo lóbulos. Sendo assim realizadas as leituras, mesmo sem a conexão com algum celular, tendo o LED do dispositivo como informativo ao paciente se a leitura indica alguma alteração. Em seguida, se o profissional de saúde precisar do acompanhamento simultâneo poderá conectar por meio bluetooth o arduino e a aplicação, na opção de monitoramento do paciente pela qual receberá os dados, tendo a limitação de estar abaixo de dez metros de distância em decorrência do bluetooth. No entanto, caso o profissional necessite posteriormente dos dados obtidos anteriormente de qualquer paciente, ele poderá copiar do banco de dados do Firebase, necessitando apenas de conexão com a internet, mas sem precisar estar conectado com o dispositivo físico.

Capítulo 4

Conclusão

4.1. Sobre o Trabalho Realizado

Após o desenvolvimento do aplicativo, o conjunto do trabalho foi testado com dispositivos físicos, uma vez que os emuladores apresentados na IDE Android Studio não permitem o funcionamento do Bluetooth e portanto não haveria comunicação entre os dispositivos. As versões iniciais do app mostraram caminhos para as mudanças necessárias que foram realizadas para melhor desempenho do aplicativo.

Uma das principais alterações ocorrida foi no modelo de MeasureHeartBeats. O modelo consistia anteriormente em uma lista de medidas recebidas a cada três segundos, mantendo um registro mais completo de cada paciente por dia. No entanto, ao passar pela fase de testes, foi observado que o sistema exigiria bastante da thread principal e com o envio constante para o Firebase, causaria uma lentidão na performance do aplicativo. Desse modo, surgiu o atual modelo, com a média, o máximo e o mínimo.

Como constatado, os testes foram realizados considerando apenas um paciente por vez para o registro por dispositivo, assim como somente um profissional efetuando as medidas. Seguido a isto, a seção eletrônica e do Arduino apresentou a velocidade necessária tanto durante a recepção de dados, quanto para o envio das informações captadas.

Para a aplicação, no caso de o usuário necessitar esperar a finalização de algum serviço ou delay, foram exibidas mensagens de interação, compartilhando o processo. Ainda assim, conexões como o Firebase, assim como realização de ações dentro da plataforma com comandos oriundos do app, como por exemplo criação de novos pacientes, mostraram-se mais rápidas e fluídas do que as conexões Bluetooth entre módulo e aplicativo.

Para as versões acima do Nougat e com conexão Bluetooth compatível com protocolo versão 2.0 ou acima é prevista atuação regular do aplicativo. No entanto, apesar do bom desempenho apresentado, são necessários mais testes em ambiente característico para entender e classificar o comportamento do app e do dispositivo com diferentes variáveis de entrada e versões de Android.

Ainda assim, o projeto cumpre com os objetivos estabelecidos anteriormente, tornando possível à um profissional de saúde ao utilizar os dispositivos, receber dados via bluetooth de um sensor de frequência cardíaca captando os batimentos do paciente. Também é válido destacar que entre as matérias da grade curricular do curso, destacam-se as disciplinas de Arquitetura de Sistemas Digitais (ASD) e Projeto de Programas (PP) para realização do trabalho. Na disciplina de ASD, tanto sua teoria quanto prática em laboratório auxiliaram na compreensão de criações de sistemas entre as três camadas de hardware, firmware e software, o que foi essencial para este projeto. Enquanto em PP, foram abordados os princípios e primeiros passos para o desenvolvimento de aplicações no sistema operacional Android, tendo a base para os principais conceitos empregados na elaboração do aplicativo.

4.2. Sobre Trabalhos Futuros

Para continuidade do trabalho apresentado é possível dividir as propostas em quatro principais caminhos. O primeiro pensando no armazenamento offline. O segundo refere-se ao aumento do número de clientes e de usuários no aplicativo. O terceiro voltado para o dispositivo de entrada de dados, não sendo especificamente o estabelecido neste trabalho. E por fim, seria a inserção de novas medidas e de outras áreas de registro a serem armazenados no app.

Pensando no registro de medidas enquanto o profissional não está presente, seria necessário um armazenamento de modo offline para que estes dados não sejam perdidos. Seria necessário utilizar a memória EEPROM do arduino para isto. Deste modo, mesmo desligando a alimentação, os dados estariam salvos, uma vez que a EEPROM é uma memória não-volátil. Deste modo, se fosse necessário seria possível o paciente continuar com o dispositivo, e quando o profissional retornasse, seria feita a comunicação do aplicativo com a memória, tendo assim todos os registros feitos durante sua ausência. Com isto, seria substancial a alteração de algumas funcionalidades e telas de apresentação, para acolher esta nova opção de obtenção de dados.

Uma vez que foi testado apenas para uma única empresa e com ênfase em um profissional por vez, a primeira possível melhoria a ser trabalhada envolveria uma alteração nos modelos e conseqüentemente no banco de dados do projeto. Deste modo, far-se-ia necessário a presença de login para controle de acesso de cada usuário para cada empresa correspondente.

Seriam criados três modelos a mais para o aplicativo: o de empresa, o de profissional e o de paciente. De modo que, o perfil de cliente, ou empresa, teria funcionalidades de cadastro, com a adição do CNPJ ou correspondente para validação da criação do perfil, login, possibilidade de adição de profissionais ligados à empresa e gerenciamento destes perfis. O modo profissional de saúde teria seu cadastro com validação possível por dois meios, caso uma empresa já houvesse criado seu perfil, podendo assim acessar por meio deste, e efetuar os registros e cadastros de seus pacientes, ou de modo autônomo, mas com a verificação do número registrado no CRM salvo durante o cadastro, com as mesmas funcionalidades de atuação. Por fim, para o perfil de paciente não seria necessária nenhuma validação, e o público alvo seriam para pessoas que preferem realizar e armazenar registros próprios de medidas, com adição de funcionalidade de envio de planilha, facilitando assim, caso necessário o repasse de informações para familiares, possíveis cuidadores ou até consultas.

Com os novos tipos de objeto citados acima, seria possível manter o controle de todas as medidas por paciente, que poderia estar ligado ou não a um profissional de saúde habilitado, que conseqüentemente, também poderia estar conectado ou não a uma empresa responsável. Com a ideia principal de fazer identificadores para cada modelo e realizar a relação entre diferentes objetos dentro do banco de dados para eventuais buscas, verificações de informações, edições e downloads de arquivos que se comunicam entre si. A maioria dos atuais softwares disponíveis atuam no ambiente específico de uma empresa ou no modo de perfil cliente.

Quanto ao dispositivo que efetuará o envio de dados de entrada para o aplicativo, seria possível uma segunda opção, um dispositivo *wearable*. Os *smartwatches* são pequenos, usados no pulso e podem alcançar desempenho semelhante aos smartphones. A operação do relógio usualmente é por meio de uma tela sensível ao toque e/ou botões físicos dependendo do modelo e marca. Regularmente tem acoplados medidores de frequência, pedômetros e outros, sendo assim, muito utilizados para acompanhamentos médicos.

Deste modo, qualquer paciente que já utilizasse um smartwatch compatível com o celular poderia ter suas medidas registradas sem a necessidade do dispositivo eletrônico descrito no projeto. No entanto, para isto seria necessário a criação de um aplicativo *wearable* para atuação do lado do relógio inteligente.

O aplicativo dentro do relógio teria que receber primeiramente o sinal de conexão entre o relógio e o aplicativo específico do projeto, e a partir disso iniciar o envio de dados capturados do mesmo. Também sendo possível encerrar a atividade assim que houver a desconexão pela parte do app. Enquanto no aplicativo do celular não haveria grandes alterações a serem realizadas, uma vez que a conexão e recepção de dados dar-se-ia do mesmo modo.

Referente à questão de novas medidas e áreas, a pesquisa citada no capítulo 3 pode ser lembrada. Os profissionais assinalaram quais medidas são consideradas principais para um registro frequente de um paciente a ser observado. Dentre as mencionadas, destacam-se: pressão arterial, batimentos cardíacos, saturação, temperatura e glicose.

A frequência cardíaca foi trabalhada neste projeto, sendo assim, as medidas que auxiliam os profissionais construírem uma melhor base de informações seriam as restantes. Além destas, também foi citada durante o processo da pesquisa a importância de ter ao menos um resumo do histórico médico do paciente e a possibilidade de realizar observações ao decorrer dos preenchimentos.

Ainda é importante destacar informações sobre o estado mental de quem está sendo acompanhado. Dentre os estudos na área, destacam-se as palavras de Maeda, Saiki e Nakamura ^[37] no estudo sobre “Registros de estados de saúde com chatbot pelo celular”, qual falam sobre como o monitoramento utilizando sensores limita-se a eventos observáveis externamente, sendo necessário também reconhecer o estado interno do idoso, para um melhor cuidado e acompanhamento domiciliar.

Deste modo, a última adição a uma proposta futura seria a adição dos registros psicológicos do paciente. Para o perfil de um profissional, a opção seria de uma entrada manual do acompanhamento realizado a ser armazenado. Enquanto para o perfil paciente, seria apresentado um pequeno fluxo conversacional programado para acompanhamento. O fluxo provavelmente seria definido utilizando o Dialogflow para gerenciamento da interface conversacional, utilizando, após pesquisa, perguntas básicas frequentemente utilizadas para este modelo de monitoramento.

Com as possíveis melhorias citadas, este projeto possivelmente teria um alcance maior, trazendo um modo de armazenamento de registro mais amplo e completo.

Referências Bibliográficas

[1] IBGE. Diretoria de Pesquisas, Coordenação de trabalho e Rendimento, Pesquisa Nacional por Amostras de Domicílio Contínua . Características gerais dos moradores 2019. Rio de Janeiro, RJ. 2020.

[2] IBGE. SIMÕES, Celso Cardoso da Silva, Coordenação de População e Indicadores Sociais. Relações entre as alterações históricas na dinâmica demográfica brasileira e os impactos decorrentes do processo de envelhecimento da população. Rio de Janeiro, RJ. 2016. pg 95-96.

[3] Ministério da Saúde. Caderneta de Saúde da Pessoa Idosa. Manual de Preenchimento. 5ª Ed. Brasília, DF. 2018.

[4] THOMSEN, Adilson. O que é Arduino, para que serve e primeiros passos. Disponível em: <<https://www.makerhero.com/blog/o-que-e-arduino/>>. Acesso em: 20 out. 2022.

[5] MCROBERTS, Michael. Arduino Básico. Novatec Editora. São Paulo, SP. 2011.

[6] Arduino Uno R3. Datasheet: Product Reference Manual. Last publication: 2023

[7] World Famous Electronics,LLC. Pulse Sensor, Easy to use heart rate sensor & kit. Nova York, EUA.

[8] World Famous Electronics, LLC. Pulse sensor getting started Guide. Nova York EUA.

[9] Blog Eletrogate. Módulos Bluetooth HC-05 e HC-06 para comunicação com dispositivos móveis com arduino. Disponível em: <<https://blog.eletrogate.com/modulos-bluetooth-hc05-e-hc06-para-comunicacao-com-dispositivos-moveis-com-arduino/>>. Acesso em: 27 out. 2022.

[10] Laboratório Mobilis. Tutorial Android para comunicação bluetooth. Disponível em: <<http://www2.decom.ufop.br/imobilis/tutorial-android-comunicacao-bluetooth/>>. Acesso em: 15 out. 2022.

[11] ITeadStudio. Datasheet: Bluetooth to Serial Port Module. 2010.

[12] My mob. História do android. Disponível em: <<https://mymob.com.br/blog/historia-do-android.html>>. Acesso em: 15 out. 2022

[13] ALMENARA, Igor. CIRIACO, Douglas. Qual o sistema operacional de celular mais usado no mundo. Canal Tech. 2022. Disponível em: <<https://canaltech.com.br/software/qual-o-sistema-operacional-de-celular-mais-usado-do-mundo-223862/>>. Acesso em: 15 out. 2022.

- [14] GRIFFITHS, Dawn. GRIFFITHS, David. Head First Android Development. O'Reilly Media Inc. Califórnia, CA, EUA.2015. Chapter 1.
- [15] Android. O android é para todos. Disponível em: <https://www.android.com/intl/pt-BR_br/everyone/facts/>. Acesso em: 15 out.2022.
- [16] Devmedia. Conceitos básicos para programar para Android. Disponível em: <<https://www.devmedia.com.br/conceitos-basicos-para-programar-para-android/40112>>. Acesso em: 15 out. 2022.
- [17] Android Developer. Arquitetura da plataforma. Disponível em: <<https://developer.android.com/guide/platform?hl=pt-br>>. Acesso em: 15 de out. 2022.
- [18] JUNIOR, Luiz Fernando Duarte. Meu primeiro App Android. LuizTools. 2016.
- [19] Android Developer. Conheça o Android Studio. Disponível em: <<https://developer.android.com/studio/intro?hl=pt-br>>. Acesso em: 30 out. 2022.
- [20] TOLEDO. Lucas Henrique Bento. Desenvolvimento de Aplicações Android Adaptáveis utilizando a linguagem Kotlin. Universidade Federal de Ouro Preto, Instituto de Ciências Exatas. Ouro Preto, MG. 2019.
- [21] Geeks for geeks. Introduction to Kotlin. Disponível em: <<https://www.geeksforgeeks.org/introduction-to-kotlin/>>. Acesso em: 30 out. 2022.
- [22] Kotlin Language Documentation 1.8.0. Kotlin Lang. 2023.
- [23] KUMAR S, Ashok. Mastering Firebase for Android Development: Build real-time, scalable, and cloud-enabled Android apps with Firebase. Packt Publishing. 2018.
- [24] Google Developers. Monetize seu Negócio: Usando o Firebase para Lucrar mais. Disponível em: <<https://developers-br.googleblog.com/2016/11/monetize-seu-aplicativo-usando-o-firebase.html>>. Acesso em: 30 out. 2022.
- [25] Firebase Google. Entenda os projetos Firebase. Disponível em: <<https://firebase.google.com/docs/projects/learn-more>>. Acesso em: 30 out. 2022.
- [26] VICE, Ryan. SIDDIQI, Muhammad Shujaat. MVVM Survival Guide for Enterprise Architectures in Silverlight and WPF. Packt Publishing. Birmingham - Mumbai. 2012
- [27] BARBOSA, Adriley Samuel Ribeiro. Análise Comparativa entre os padrões MVC, MVP, MVVM e MVI na Plataforma Android. Instituto Federal de Educação, Ciência e Tecnologia Goiano - Campus Urutaí. 2022.
- [28] TERENCE, Ana Cláudia Fernandes. FILHO, Edmundo Escrivão. Abordagem quantitativa, qualitativa e a utilização da pesquisa-ação nos estudos organizacionais. Encontro Nacional de Engenharia de Produção - ENEGEP. Fortaleza, CE. 2006.

[29] CNN Brasil. No Brasil mais de 230 mil pessoas morreram por doenças cardiovasculares em 2021. Disponível em: <<https://www.cnnbrasil.com.br/saude/no-brasil-mais-de-230-mil-pessoas-morreram-por-doencas-cardiovasculares-em-2021/>>. Acesso em: 03 nov. 2022.

[30] Secretaria de Vigilância em saúde. Boletim epidemiológico: Mortalidade de idosos no Brasil em 2000, 2009 e 2019. Ministério da Saúde. Volume 53. 2022.

[31] World Health Organization. The Top 10 causes of death. Geneva: WHO. 2020.

[32] Sociedade Brasileira de Cardiologia. Aumenta o número de mortes por doenças cardiovasculares no primeiro semestre de 2021. Disponível em: <<https://www.portal.cardiol.br/post/aumenta-o-n%C3%BAmero-de-mortes-por-doen%C3%A7as-cardiovasculares-no-primeiro-semester-de-2021>>. Acesso em: 03 nov. 2022

[33] Github. Pulse Sensor Amped Arduino. Disponível em: <https://github.com/WorldFamousElectronics/PulseSensor_Amped_Arduino>. Acesso em: 27 out. 2022.

[34] NORONHA, Bruna. A psicologia das cores e sua relação com o UX Design. UX Collective BR. 2020

[35] Firebase Google. Adicione o Firebase ao seu projeto Android. Disponível em: <<https://firebase.google.com/docs/android/setup?hl=pt-br>>. Acesso em: 22 nov. 2022.

[36] Android Developer. Introdução a atividades. Disponível em: <<https://developer.android.com/guide/components/activities/intro-activities>>. Acesso em: 22 nov. 2022.

[37] MAEDA, Haruhisa; SAIKI, Sachio; NAKAMURA, Masahide; YASUDA, Kiyoshi. Recording Daily Health Status with Chatbot on Mobile Phone – A preliminary study. Graduate School of Systems Informatics, Kobe University; Faculty of Informatics, Osaka Institute of Technology.

Anexos

Anexo 1

Questionário

1. Qual seu nome?
2. Qual sistema operacional do seu celular?
 - a) iOS
 - b) Android
 - c) Outros
3. Qual sua profissão?
4. Na sua profissão, é necessário o registro de medidas, como pressão arterial, batimentos cardíacos, entre outros?
 - a) Sim
 - b) Não
5. Se sim, como é feito este registro?
 - a) Manualmente, escrevo tudo do modo mais organizado que consigo.
 - b) Digitalmente, uso planilhas pelo notebook para organizar.
 - c) Digitalmente, mas costumo utilizar mais o celular para anotações.
6. Costuma fazer os atendimentos de forma:
 - a) Domiciliar
 - b) Hospital/Clínica
 - c) Outros
7. Qual é a faixa etária dos atendimentos que realiza? (Ex. : 0-1, 1-18 anos, 18-40 anos...)
8. Quais medidas são mais importantes para registrar? Marque todas que achar necessário.
 - a) Pressão arterial
 - b) Batimentos cardíacos
 - c) Temperatura
 - d) Saturação
 - e) Peso
 - f) Glicose
 - g) Outros
9. Tem/Teve que transformar estes registros para o formato digital?
 - a) Sim, onde trabalho preciso enviar os relatórios em formato de planilha.
 - b) Sim, precisei dos registros digitais para melhor organização.
 - c) Não, posso manter meus relatórios manuais.
10. Já utilizou aplicativos de celular para realizar estes registros?
 - a) Sim, sempre procuro alguns para facilitar
 - b) Não, ainda não utilizo o celular para fazer os registros

11. O que considera que não poderia faltar em um aplicativo para estes registros?
- a) Todas as medidas citadas anteriormente
 - b) Histórico do paciente
 - c) Sintomas do dia
 - d) Acompanhamento básico de psicológico
 - e) Exportação dos dados
 - f) Agenda de remédios
 - g) Espaço para observações adicionais
 - h) Outros
12. Você usaria um aplicativo que tivesse as opções selecionadas acima?
- a) Sim, com certeza, facilitaria bastante o processo
 - b) Acho que sim, dependendo se realmente for útil
 - c) Acho que não, já tenho o meu modo de organizar
 - d) Não.
13. Pode usar este espaço para comentários que considerar necessários.