UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO - UFES

CENTRO DE CIÊNCIAS EXATAS, NATURAIS E DA SAÚDE

CURSO DE SISTEMAS DE INFORMAÇÃO

RAPHAEL LEAL CARLETTI

APLICATIVO MOBILE PARA GERENCIAMENTO DE INTERVENÇÕES FÍSICAS DO PROJETO "VIDA SAUDÁVEL"

ALEGRE
2017

RAPHAEL LEAL CARLETTI

APLICATIVO MOBILE PARA GERENCIAMENTO DE INTERVENÇÕES FÍSICAS DO PROJETO "VIDA SAUDÁVEL"

Trabalho de Conclusão de Curso apresentado ao Departamento de Computação do Centro de Ciências Exatas, Naturais e da Saúde da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.
Orientador: Prof. D.Sc. Clayton Vieira Fraga Filho

Alegre
2017

RAPHAEL LEAL CARLETTI

APLICATIVO MOBILE PARA GERENCIAMENTO DE INTERVENÇÕES FÍSICAS DO PROJETO "VIDA SAUDÁVEL"

Trabalho de conclusão de curso apresentado ao Departamento de Computação do Centro de Ciências Exatas, Naturais e da Saúde da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Aprovado em 14 de dezembro de 2017.

COMISSÃO EXAMIDADORA

Prof. D.Sc. Clayton Vieira Fraga Filho
Universidade Federal do Espírito Santo
Orientador

Profa. MSc. Valéria Alves da Silva
Universidade Federal do Espírito Santo

Profa. MSc. Helder de Amorim Mendes
Universidade Federal do Espírito Santo

"Você pode encarar um erro como uma besteira a ser esquecida, ou como um resultado que aponta uma nova direção."

Steve Jobs

AGRADECIMENTOS

Primeiramente queria agradecer a Deus por mais essa conquista e a minha mãe Ana Paula Leal, que sempre me mostrou onde caminhar para conquistar tudo que eu sempre quis e sempre com muito amor e carinho. E ao meu pai Carlos Carletti, que sempre no seu silêncio me mostrava seu amor infinito e que sempre está guardado em meu peito.

Também queria agradecer a todos os meus amigos da faculdade, Caique, Melissa, Silas, e todos da turma 2012/2 que me acompanharam em todas as alegrias e tristezas desses 5 anos de faculdade, e também a família Xinelada, onde pude fazer parte por quase 3 anos e que me trouxeram muitos momentos felizes ao lado da melhor formação possível: Daniel, Marcão, Diego, Hugo, Robson, André. E aos amigos de Alegre que sempre estavam comigo, Lara, Janna, Gabriela, Isabela entre tantos outros.

Obrigado também a todos os professores que ajudaram e contribuíram de alguma forma para o meu crescimento pessoal e profissional, em especial o meu professor orientador Clayton Fraga Filho, que fez com que esse trabalho pudesse tomar forma.

Eu ainda assim não teria a experiência de empresa durante esse tempo, se não fosse pela Click Junior, onde fiz parte durante mais de um ano, e me ensinou muito, e gostaria de agradecer imensamente a quem estava nessa caminhada, em especial Denis, Natália e Wagner, que me ensinavam cada dia um pouco mais.

RESUMO

Dispositivos móveis tem se tornado cada vez mais parte do dia-a-dia do brasileiro, assim, fazendo com que a vida de muitas pessoas se torne mais pratica. Com o objetivo de tornar um processo que antes era completamente feito sem o uso de tecnologia, em algo mais rápido e com informações mais precisas, foi desenvolvido um aplicativo para a plataforma iOS que utiliza do Firebase que implementa uma comunicação entre o professor orientador e os alunos do projeto.

Palavras Chave: Desenvolvimento iOS, Firebase

ABSTRACT

Mobile devices have become more and more part of brazilian life, in that way, making the life of a lot of people easier. With the objective make a process that was completely done without the use of technology in something that is faster and with accurate, was developed an application for iOS operational system that use Firebase to implement the communication between the project teacher and the students.

Keywords: iOS development, Firebase

LISTA DE FIGURAS

Figura 1: MVC (Orlov, 2015) 16

Figura 2: Cocoa MVC (Orlov, 2015) 17

Figura 3: Diagrama de sequência da comunicação do aplicativo com o firebase. Fonte: O autor.....18

Figura 4: Firebase Authentication (Firebase, 2017) 18

Figura 5: Firebase Authentication mostrando os usuários de várias plataformas na mesma tela (Firebase, 2017) 19

Figura 6: Diagrama que mostra como o Firebase Authentication funciona. Fonte: Firebase (2017).....19

Figura 7: Firebase Realtime Database em formato de árvore (Firebase, 2017).....20

Figura 8: Fluxograma das etapas da metodologia 22

Figura 9: Diagrama de caso de uso. Fonte: O autor. 23

Figura 10: Diagrama de classe das classes modelo da aplicação. 24

Figura 11: Digrama de estados da aplicação. 25

Figura 12: Código para requisitar as inforamações de usuários do Firebase e tratar elas para informações do app. Fonte: O autor.....26

Figura 13: Regras de leitura e escrita do banco de dados. Fonte: O autor. 27

Figura 14: Diagrama de desenvolvimento do aplicativo. 29

Figura 15: Diagrama de sequência do caso de uso "cadastrar paciente" 29

Figura 16: Diagrama de classe de cadastrar paciente. 30

Figura 17: Adicionar paciente. Fonte: O autor.....31

Figura 18: Diagrama de sequência de edição de paciente. 32

Figura 19: Diagrama de classe de editar paciente. 32

Figura 20: Diagrama de sequência de remoção de paciente. 33

Figura 21: Diagrama de classe de remover paciente. 34

Figura 22: Tela de pacientes. Fonte: O autor 35

Figura 23: Diagrama de sequência do caso de uso "cadastrar consulta" 36

Figura 24: Diagrama de classe de cadastrar consulta 36

Figura 25: Tela de consultas e adicionar consultas. Fonte: O autor.....37

Figura 26: Diagrama de sequência de remoção de consulta. 37

Figura 27: Diagrama de classe de remover consulta 38

Figura 28: Diagrama de sequência do cadastro de um usuário. 39

Figura 29: Diagrama de classe de cadastrar usuário. 39

Figura 30: Tela de usuários e de adicionar usuário. Fonte: O autor.....40

Figura 31: Diagrama de sequência de edição de usuário. 41

Figura 32: Diagrama de classe de edição de usuário. 41

Figura 33: Diagrama de sequência de login de usuário. 42

Figura 34: Diagrama de classe de logar no sistema. 43

Figura 35: Tela de login. Fonte: O autor.....44

Figura 36: Gráfico do resultado do questionário pós-teste.....46

LISTA DE QUADROS

Quadro 1: Defeitos encontrados no aplicativo. 45

Quadro 2: Caso de teste 001 50

Quadro 3: Caso de teste 002 50

Quadro 4: Caso de teste 003 51

Quadro 5: Caso de teste 004 51

Quadro 6: Caso de teste 005 52

Quadro 7: Caso de teste 006 53

Quadro 8: Caso de teste 007 54

SUMÁRIO

1 INTRODUÇÃO 12
2 OBJETIVOS 14
2.1. Objetivo geral.....14
2.2. Objetivos específicos.....14
3 REVISÃO BIBLIOGRÁFICA 15
3.1. Projeto "Vida Saudável".....15
3.1.1. Obesidade 15
3.1.2. Cardiopatia 15
3.1.3. Diabetes 15
3.2. Desenvolvimento iOS 15
3.3. Arquitetura para desenvolvimento mobile.....16
3.3.1. Model-View-Controller (MVC) 16
3.3.2. Cocoa MVC 17
3.4. Firebase.....17
3.4.1. Firebase Authentication: Serviço de back-end para autenticação . 18
3.4.2. Realtime Database 19
4 METODOLOGIA.....22
4.1. Implementação do Aplicativo 22
4.1.1. Casos de uso..... 23
4.1.2. Diagrama de Classes de Domínio 23
4.1.3. Diagrama de Estados 25
4.2. Integração do Aplicativo com o Firebase.....25
4.3. Configurações do Firebase.....26
5 RESULTADOS OBTIDOS 28
5.1. Arquitetura da aplicação com o Firebase 28
5.2. Diagrama de sequência, diagrama de classe e telas de cada caso de uso 29

5.3. Teste Funcional 44

5.4. Teste de usabilidade.....45

6 CONCLUSÕES 47

7 REFERÊNCIAS BIBLIOGRÁFICAS 48

8 ANEXOS 50

8.1. Anexo A – Teste funcional.....50

8.2. Anexo B – Teste de Usabilidade.....54

8.3. Anexo C – Questionários respondidos sobre o teste de usabilidade.....55

8.4. Anexo D – Casos de uso da aplicação 56

1 INTRODUÇÃO

Segundo dados da Fundação Getúlio Vargas de São Paulo (FGV-SP), citado pela Folha de São Paulo (2017), o número esperado de smartphones no Brasil até o fim de 2016 era de 168 milhões de aparelhos o que seria um aumento de 9% em relação a 2015. Com esse aumento, a distribuição da informação tem sido cada vez mais armazenada virtualmente, que são acessíveis por meio de uma ampla gama de dispositivos. É o que permite que conteúdos sejam alcançáveis de qualquer lugar via equipamentos móveis. (LUSTOSA, 2013)

É possível encontrar nos dias de hoje, dados que mostram que 1 a cada 5 pessoas no Brasil está acima do peso. Segundo o Ministério da Saúde, citado pela Globo G1 (2017), a obesidade cresceu em 60% nos últimos 10 anos, passando de 11,8% para 18,9% da população brasileira em 2016 e o número de brasileiros com diabetes e hipertensão cresceram respectivamente de, 5,5%, em 2006, para 8,9%, em 2016, e de 22,5% para 25,7% no mesmo período.

Como já dito, o aumento da obesidade vem sendo alarmante no Brasil, e existem várias hipóteses e motivos para explicar o grande aumento ter em pouco tempo. Segundo o médico Cláudio Corá, citado por Globo G1 (2017), um dos fatores mais preocupantes sobre esse aumento é dado pela mudança de hábitos alimentares que pode ser notado desde 1970, com a diminuição de tempo hábil para fazer comida em casa, a população passou a preferir comidas mais rápidas e consequentemente mais calóricas.

O motivo para o aumento de pessoas diabéticas no Brasil, segundo Padilha, citado por Portal Saúde (2013), tem sido causado também pela má alimentação que tem sido feita pelos brasileiros. Ele também diz que está sendo feito uma preparação no sistema de saúde para lidar com esse novo quadro.

A hipertensão pode também ter seu aumento referente a mudança de hábitos alimentares e até mesmo com a obesidade propriamente dita, porém, segundo Mekitharian (sem data), a doença pode ser causada por outros diversos fatores como idade, alcoolismo, falta de exercício, dentre outros.

Com base nesses dados, foi criado um projeto na Universidade Federal do Espírito Santo, chamado “Vida Saudável”, para auxiliar e acompanhar o paciente hipertenso, diabético e/ou obeso com a parte nutricional do paciente, e também manter registro de avaliações clínicas e pressórica de tempos em tempos. Com isso é possível avaliar se

houve melhora no quadro no paciente e ajudando-o a fazer isso no seu dia a dia, motivando a prática de exercícios físicos acompanhados de um profissional.

Nesse projeto, as avaliações são todas feitas em formulário que é preenchido a caneta, sem utilização de meios tecnológicos, e em seguida repassado ao professor responsável para que, consulta por consulta, possa ser incluída em um banco de dados. Nesse caso, há um atraso de informação, que afetaria na qualidade de dados, o que pode vir a causar ineficiência operacional, compromete tomada de decisões, e além disso, não teria os dados de forma adequada quando se espera, podendo chegar a perder informações, o que pode impactar mais ainda na decisão. (Loushin, 2001)

Uma solução para esse problema seria o uso de um aplicativo mobile que armazenaria as informações necessárias (que hoje são guardadas em formulários) e poderia ser visualizada por todos os alunos do projeto.

2 OBJETIVOS

2.1. Objetivo geral

Propor um aplicativo para auxiliar no gerenciamento de intervenções físicas realizadas pelos bolsistas do projeto “Vida Saudável” da Universidade Federal do Espírito Santo para auxiliar na visualização do rendimento de cada paciente em intervenções passadas, tanto para o aluno quanto para o orientador.

2.2. Objetivos específicos

São objetivos específicos deste trabalho:

a. Realizar um teste de usabilidade e analisar seus resultados.
b. Realizar testes funcionais com o objetivo de validar o aplicativo.

3 REVISÃO BIBLIOGRÁFICA

3.1. Projeto "Vida Saudável"

Segundo Leopoldo (2016), o projeto "Vida Saudável" tem como objetivo melhorar a qualidade de vida de pessoas, cardiopatas, diabéticas e obesas, prevenindo doenças crônicas não-transmissíveis (DCNTs) com a utilização de exercícios físicos e acompanhamento nutricional.

3.1.1. Obesidade

A obesidade pode ser definida por um excesso de gordura corporal em relação a massa magra, e sobre peso é uma proporção maior de gordura em relação a massa magra (Oliveira et. Al., 2003), porém não grande suficiente a ponto de ser chamado de obesidade.

3.1.2. Cardiopatia

Segundo o Portal São Francisco (2017), a cardiopatia é classificada como toda e qualquer doença relacionada ao coração e sistema sanguíneo, entre elas, as mais conhecidas são o Acidente Vascular Cerebral (AVC) e a Hipertensão Arterial.

3.1.3. Diabetes

A diabetes é uma doença que acontece quando ocorre uma elevação da taxa de glicose no sangue, que pode ocorrer devido a defeitos tanto na secreção quando na falta ou disfunção da insulina, que é produzida pelo pâncreas (Sociedade Brasileira de Endocrinologia e Metabologia, 2017).

3.2. Desenvolvimento iOS

O desenvolvimento nativo para iOS requer a utilização de um Ambiente de Desenvolvimento Integrado ou *Integrated Development Environmen*t (IDE) própria, criada pela própria Apple chamada de Xcode, e apenas é possível criar aplicativos nativos utilizando como linguagem de programação o Objective-C ou o Swift.

O Xcode, segundo a própria empresa, é a IDE que é o centro de toda a experiência de desenvolvimento para aplicações para dispositivos da Apple. Além disso, ela é

integrada com o Cocoa, e o Cocoa Touch, que são frameworks para ajudar na hora de criar aplicações, seja elas para Mac, iOS, Apple TV, entre outros (Apple Developer, 2017). Foi feita para compilar linguagens como: Swift, Objective-C, Objective-C++, C, C++. Também trás consigo ferramentas para ajudar no desenvolvimento, desde simuladores para os dispositivos da Apple, até ferramentas que simula internet de diferentes formas para testes mais profundos.

Já o Swift, também segundo a empresa, é uma das linguagens de programação usadas para criação de aplicativos para iOS, Mac, Apple TV e Apple Watch, além disso, é uma linguagem consistente e intuitiva (Apple Developer, 2017).

3.3. Arquitetura para desenvolvimento mobile

3.3.1. Model-View-Controller (MVC)

Segundo Orlov (2015), O padrão MVC é composto principalmente por três elementos:

- ***Model***: responsável pelos dados e a manipulação dos mesmos para que a informação seja entregue a *controller* pronta para uso.
- ***View***: responsável pela camada de visualização, onde é implementada a interface da aplicação.
- ***Controller***: responsável por solicitar os dados que se encontram na model, para que seja mostrado na interface para o usuário.

Esse padrão tem como vantagem a divisão de responsabilidade, o que facilita no entendimento na hora da leitura de código, melhora o reuso das views e models, e melhora, com a separação de responsabilidades mais bem definida (Orlov, 2015).

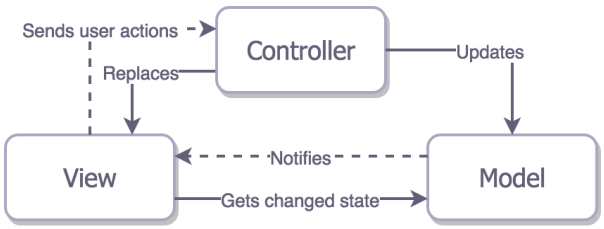

Figura 1: MVC (Orlov, 2015)

O funcionamento do MVC, como visto no diagrama da figura 1, é dado da seguinte maneira: o usuário, utilizando a *view,* manda uma ação para a *controller*, que por sua vez atualiza ou consulta a *model* que, então, notifica a *view* que ouve mudança, e assim a view pega o estado da *model* para atualizar os elementos da *view*.

Além disso tudo é uma boa arquitetura para dividir o trabalho em equipes diferentes mexendo no mesmo módulo e para realizar o trabalho assincronamente.

3.3.2. Cocoa MVC

O Cocoa MVC é um padrão de arquitetura de software muito ultilizado em aplicações iOS, que inclusive foi criado pela Apple com base no MVC apresentado na seção 3.3.1 e pode ser visto na Figura 2.

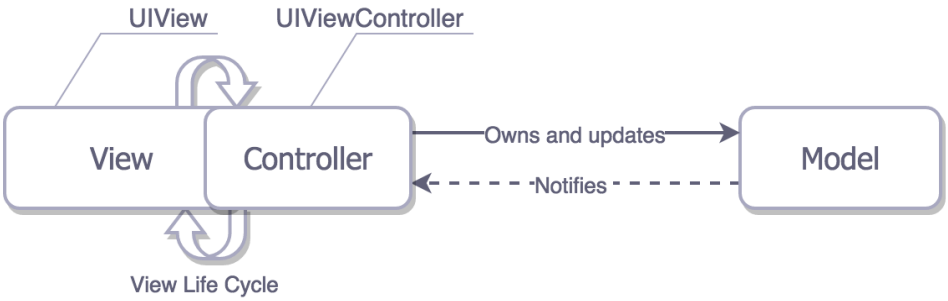

Figura 2: Cocoa MVC (Orlov, 2015)

Como mostra na figura 2, o Swift mantém no seu ciclo de vida a view e a controller sempre unidas, o que faz com que ambas sejam configuradas juntas tendo a divisão com a model, para que sejam buscados os dados da aplicação e mostrada na interface para o usuário.

Por isso, segundo Wenderlich (2016) usar esse padrão para aplicações com muitas funcionalidades, ou até mesmo muito complexas, podem vir a ser um problema, tendo em vista que a *ViewController* começa a acumular mais responsabilidades do que deveria. Com isso, cria-se uma dificuldade de fazer testes unitários, já que tudo está acoplado e uma mesma classe responde por muitas responsabilidades.

3.4. Firebase

O Firebase é um serviço de backend completo disponível via web, que pode ser usado em aplicações web e mobile. Ele é dividido em vários módulos que podem ou não ser utilizados, sendo eles o Authentication, Realtime Database, Functions, Cloud Storage, Crashlytics, entre outros (FIREBASE, 2017). Nesta seção será explicado sobre o Authentication e o Realtime Database, que foram escolhidos para serem utilizados nesse projeto.

Figura 3: Diagrama de sequência da comunicação do aplicativo com o firebase. Fonte: O autor.

3.4.1. Firebase Authentication: Serviço de back-end para autenticação

O Firebase Authentication fornece serviços de backend e um kit de desenvolvimento de software para ser usado na sua aplicação, com ele é possível autenticar, criar, e atualizar um usuário de várias formas diferente, entre elas estão: Facebook, Google+, GitHub, e-mail e senha.

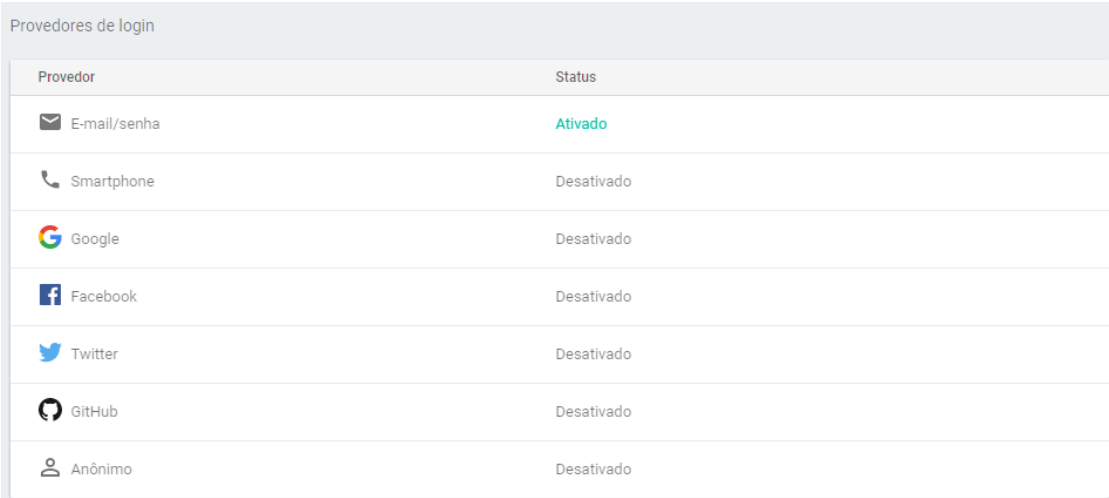

Figura 4: Firebase Authentication (Firebase, 2017)

Como pode ser visualizado nas figuras 4 e 5, ele permite que você ative ou não os métodos de login que desejar usar em sua aplicação e mantém todas as informações de usuário em um mesmo local, não precisando checar em várias plataformas ou usuários existentes.

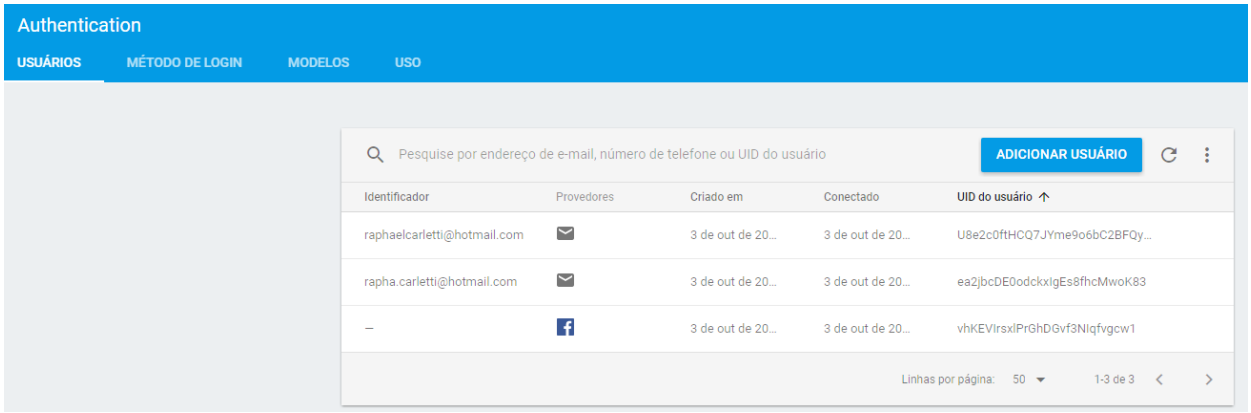

Figura 5: Firebase Authentication mostrando os usuários de várias plataformas na mesma tela (Firebase, 2017)

Como mostra a figura 6, o usuário faz uma requisição ao servidor que verifica credenciais e as informações de usuário no banco de dados, depois o Firebase Authentication verifica se as informações combinam com as que ele tem armazenado e, caso seja um serviço de terceiros, faz uma autenticação com os respectivos provedores (Google, Facebook, Twitter ou GitHub) assim, retornando ao aplicativo o UID do usuário

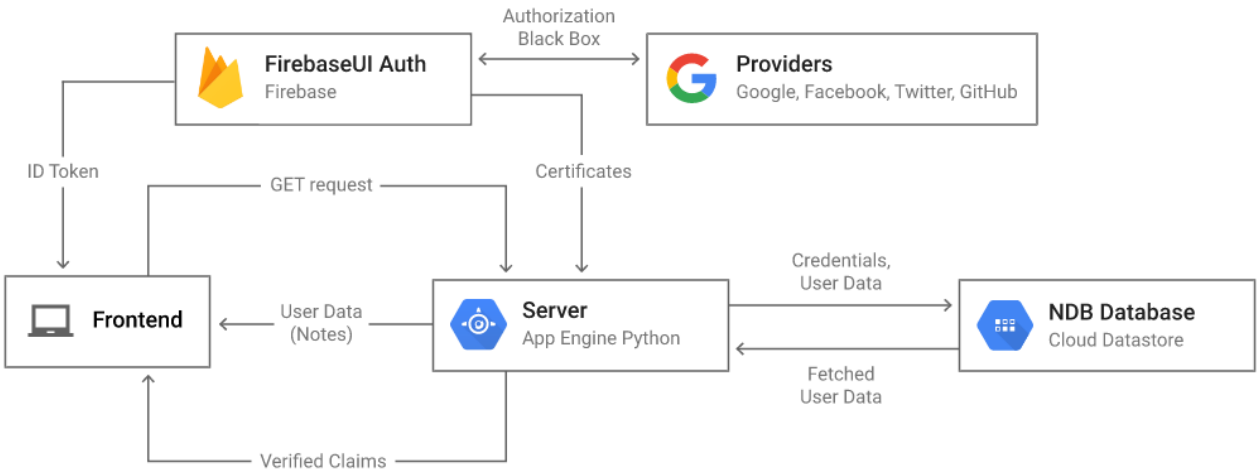

Figura 6: Diagrama que mostra como o Firebase Authentication funciona. Fonte: Firebase (2017)

3.4.2. Realtime Database

O Firebase Realtime Database é um banco de dados hospedado em nuvem que é no NoSQL, ou seja, é um banco não relacional de alto desempenho, que mostra as informações no formato de da estrutura de dados do tipo árvore e retorna as consultas

feitas a ele em formato JSON (JavaScript Object Notation) para que seja tratado pela aplicação, assim como no modelo abaixo, apresentado pela figura 7.

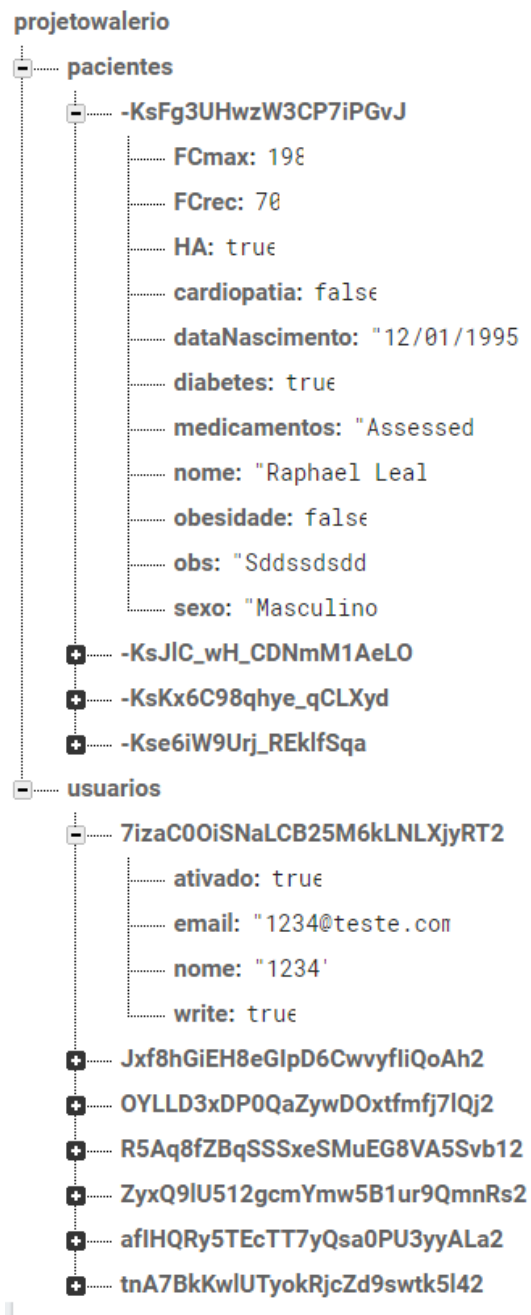

Figura 7: Firebase Realtime Database em formato de árvore (Firebase, 2017)

Como o próprio nome diz, ele é um banco de dados em tempo real, ou seja, tudo que for modificado no banco de dados é atualizado no mesmo instante no site do Firebase.

Além disso, também é possível criar regras de acesso no formato JSON para configurar o que e quem pode escrever no banco de dados ou ler, e também pode-se

configurar como validar um valor, e até mesmo manter uma ordem de forma a organizar informações de forma específica e automatizada pelo próprio Realtime Database.

É possível ainda usar de recursos da Authentication, que guarda em memória o estado do usuário (autenticado ou não) para checar se o usuário está autenticado no sistema para fazer escrita no banco de dados, o que protege o mesmo de ataques, caso o invasor consiga burlar de alguma forma o aplicativo, o Firebase tem formas de impedir que haja roubo de informações privadas.

4 METODOLOGIA

Nessa seção será apresentado a metodologia utilizada para realização do trabalho que será divida em implementação do aplicativo, configurações do Firebase e a integração do aplicativo com o Firebase.

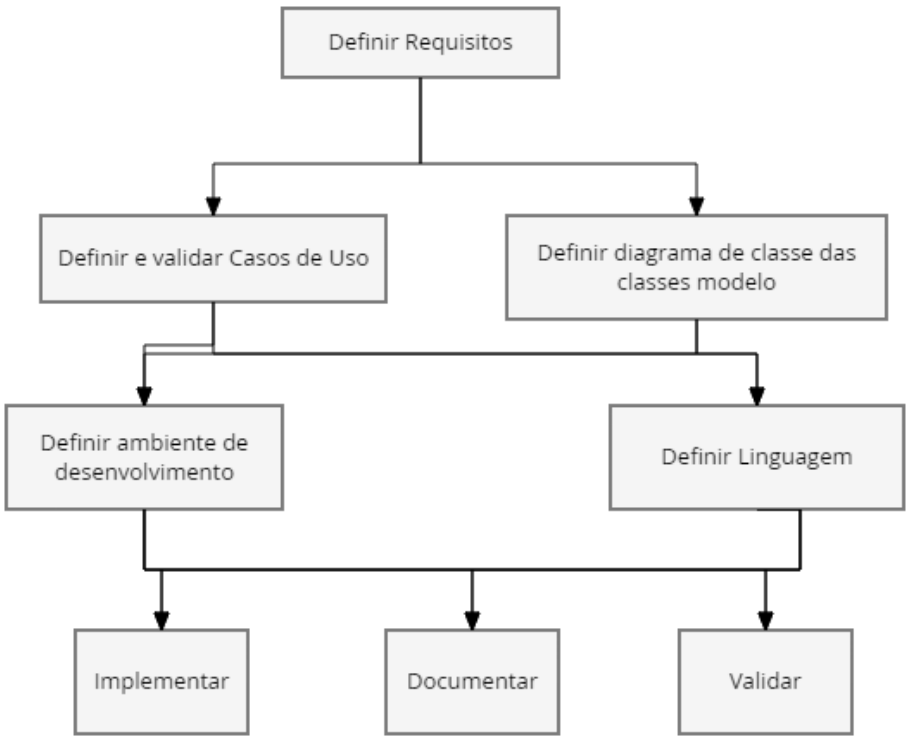

Figura 8: Fluxograma das etapas da metodologia

Na figura 8 é possível ver as etapas da metodologia utilizada nesse trabalho, onde primeiro foi definido os requisitos com um dos alunos do projeto, depois disso foi possível definir e validar os casos de usos com base nos requisito e assim criar um diagrama de classe do modelo que ia ser implementado, após isso foi definido que linguagem utilizar e o ambiente de desenvolvimento que melhor condizia com o trabalho, e por fim foi possível implementar, documentar e validar o trabalho.

4.1. Implementação do Aplicativo

Para a implementação desse aplicativo foi necessário fazer a integração com o Firebase para guardar os dados na nuvem e um front-end desenvolvido no Xcode com a linguagem Swift 3.

O objetivo desse aplicativo é manter todos os dados dos pacientes e consultas do projeto "vida saudável" que funciona atualmente na Universidade Federal do Espírito Santo, no campus de vitória, para que tudo seja guardado online e facilite a comunicação entre o professor orientador do projeto e os respectivos alunos que fazem as consultas com as pessoas cardiopatas, diabéticas e/ou obesas que o projeto ajuda. E além disso, seria possível evitar de perder dados de consultas, que são documentos importantes, com o tempo, já que agora tudo ficaria armazenado no Firebase, podendo ser consultado no aplicativo mobile.

4.1.1. Casos de uso

No diagrama apresentado na figura 9 é possível ver as funcionalidades que foram implementadas no aplicativo para que haja um controle completo dos alunos que o estão utilizando e dos pacientes e consultas a serem armazenados.

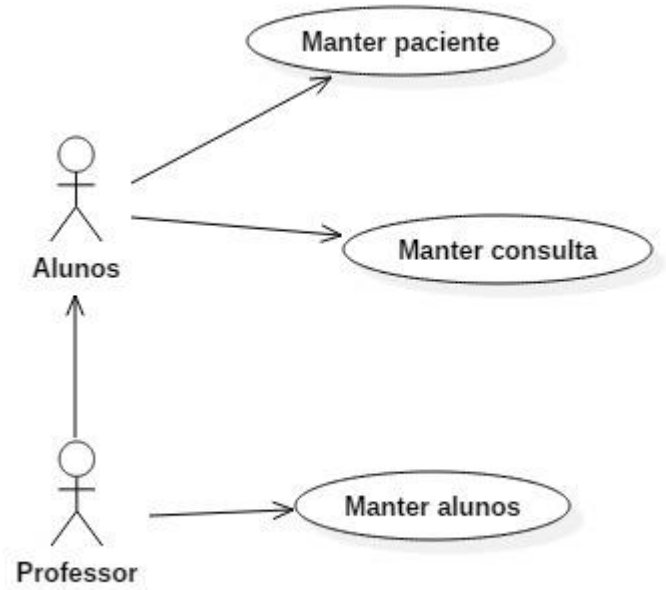

Figura 9: Diagrama de caso de uso. Fonte: O autor.

4.1.2. Diagrama de Classes de Domínio

Dessa forma, a partir dos casos de uso modelados, foi possível definir seis classes: Paciente, Consulta, Repouso, Durante, Final e Recuperação. Onde um paciente tem um relacionamento de "um para muitos" com consulta, e consulta tem um relacionamento de

"um para um" com os demais, de modo que repouso, durante, final e recuperação só existem se estiverem inseridos na consulta.

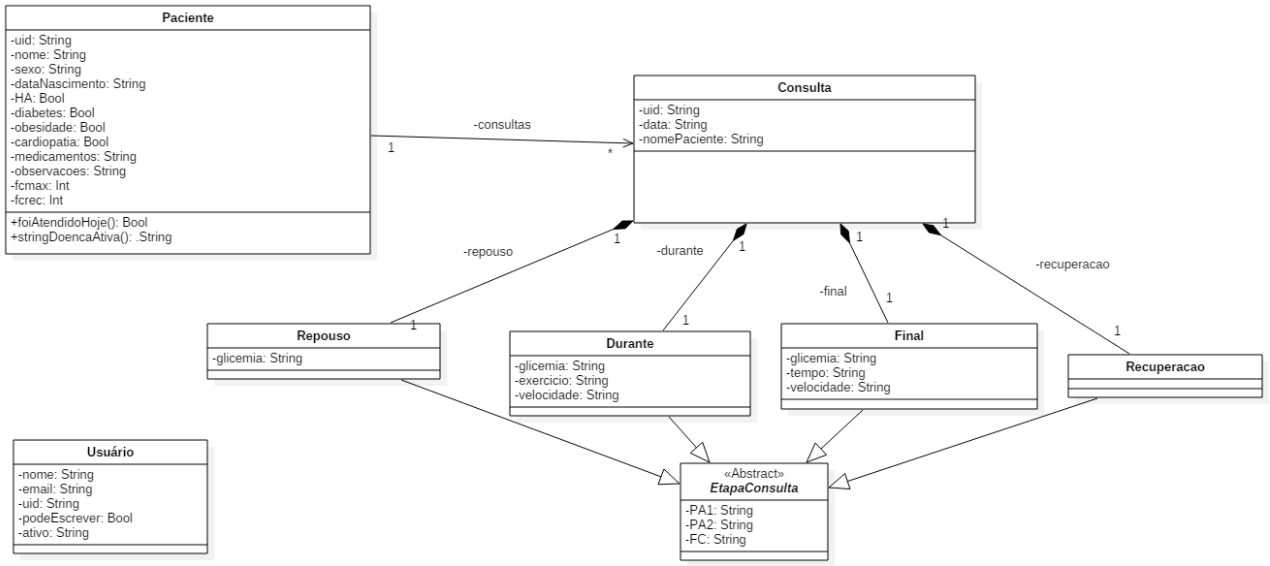

Figura 10: Diagrama de classe das classes modelo da aplicação.

O diagrama de classes da figura 10 é um diagrama das classes modelos da aplicação, que são as classes responsáveis pelos dados e a manipulação dos mesmos.

A classe Paciente tem os seguintes atributos: UID (Identificados Único), nome, sexo, data de nascimento, se o paciente tem: hipertensão, diabetes, obesidade, cardiopatia, quais os medicamentos que ele faz uso, observações adicionais, frequência cardíaca máxima e em repouso. Além disso, um paciente pode ter várias consultas, que são armazenadas na própria classe.

Já a classe Consulta tem os seguintes atributos: repouso, durante, final, recuperação (que são os estados da consulta), UID, data da consulta e o nome do paciente.

E por fim, as quatro classes que compõem uma consulta são repouso, durante, final e recuperação, onde todas têm os atributos: pressão arterial (PA1 e PA2) e frequência cardíaca (FC). Em repouso, durante e final também é informado a glicemia, caso o paciente tenha diabetes. Durante e no final do exercício é informado qual a velocidade do exercício em cada estado. Durante também se informa qual o exercício está sendo executado e no final quanto tempo a atividade levou para ser concluído.

4.1.3. Diagrama de Estados

Foi utilizado como base para a criação das telas o diagrama de estado descrito na figura 11.

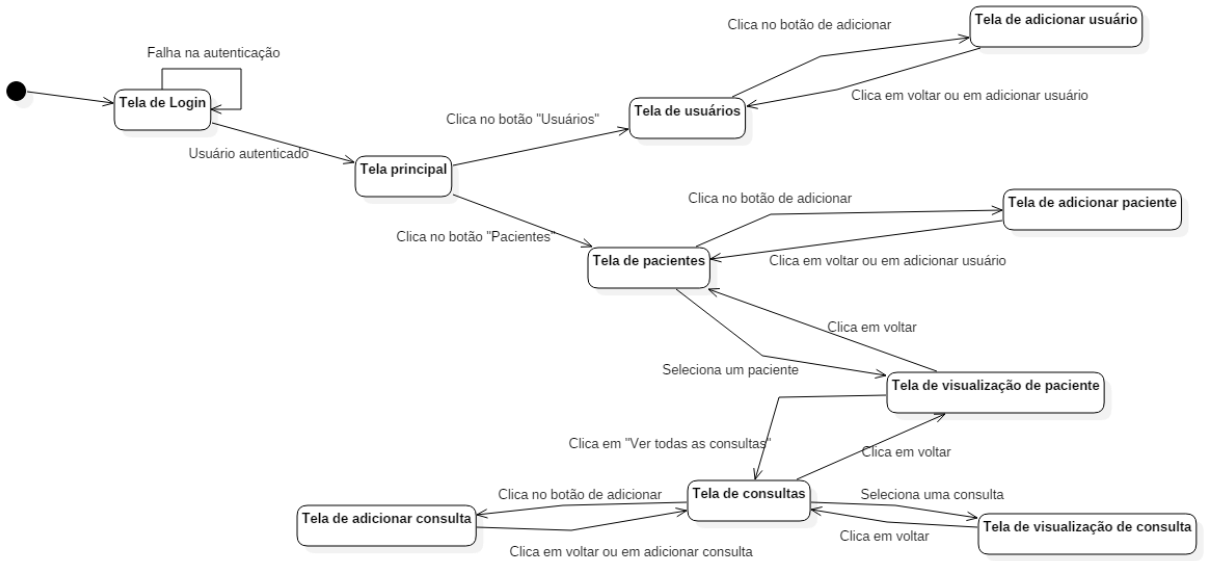

Figura 11: Digrama de estados da aplicação.

No diagrama de estados apresentado na figura 11, pode-se observar todo o possível fluxo do aplicativo e todas as telas nele envolvido, onde o ponto inicial é a tela de login, que o usuário precisa se autenticar para entrar no aplicativo. Depois ele vai para a tela principal onde é possível ver a aba de pacientes e de usuários (caso tenha permissão para isso) e então, na de usuário é possível adicionar um novo usuário.

Já na tela de pacientes é possível adicionar um novo paciente e selecionando um já existente e possível visualizar esse paciente e clicando em “Ver todas as consultas” é encaminhado para a tela onde são apresentadas as consultas referentes ao paciente em questão. É possível também, saindo desta tela, ir para a tela onde se adiciona novas consultas para esse paciente ou, caso queira visualizar uma consulta, basta seleciona a consulta que desejar.

4.2. Integração do Aplicativo com o Firebase

Para fazer a integração do aplicativo criado com o Firebase é necessária a utilização das próprias bibliotecas que é fornecida pela Google, que é o proprietário do Firebase, para que possa ser feito o compartilhamento de dados.

E para a adição de bibliotecas em um projeto no XCode foi preciso fazer uso do CocoaPods, que é um gerenciador de dependências para desenvolvimento iOS, permitindo o uso de biblioteca de terceiros.

Como mostrado na figura 12, a aplicação faz uma requisição para o Firebase que executa o que foi solicitado e logo em seguida faz o retorno dos dados necessários na aplicação e assim acontece o compartilhamento de informações entre eles.

```
let db = Database.database().reference().child("usuarios")
db.observe(DataEventType.childAdded, with: { (snapshot) in
    let dados = snapshot.value as! NSDictionary
    let user = Usuario(nome: dados["nome"] as! String,
                       email: dados["email"] as! String,
                       uid: snapshot.key,
                       write: (dados["write"] as? Bool)!,
                       ativo: (dados["ativado"] as? Bool)!)
    if snapshot.key != userID{
        self.usuarios.append(user)
    }
    self.usuarios.sort(by: { (usuario1, usuario2) -> Bool in
        return usuario1.nome < usuario2.nome
    })
    self.tableView.reloadData()
    self.activityIndicator.stopAnimating()

})
```

Figura 12: Código para requisitar as inforamações de usuários do Firebase e tratar elas para informações do app. Fonte: O autor.

A figura 12 mostra as duas primeiras linhas do código fazem a requisição ao Firebase, e as linhas dentro da função são executadas como complemento, assim que o Firbase retornar a requisição para a aplicção.

4.3. Configurações do Firebase

Para a execução nesse trabalho foi preciso fazer algumas configurações, tanto no *authentication* quanto no *realtime database*.

No *authentication* foi preciso ativar as formas de *login* e cadastro que poderiam ser utilizadas para o aplicativo, e desativar tudo o que não poderia. Pensando no problema, onde é necessário que apenas o professor cadastre os alunos do projeto, foi utilizado apenas o cadastro e *login* utilizando o e-mail e senha, para que seja mais fácil para o professor manter esses alunos no sistema.

Já no *realtime database*, para que fosse permitido ou proibidos alguns acessos, leitura e escrita, foi criado uma base de dados de regras que utiliza o formato JSON para que o próprio firebase verifique antes de aceitar alguma escrita ou leitura.

```
1  {
2    "rules": {
3      ".read":"auth.uid != null",
4      "usuarios" : {
5        ".write" : "data.child(auth.uid).child('write').val() == true",
6        ".read" : true,
7      },
8      "pacientes" : {
9        ".write" : true,
10       ".read" : true
11     }
12   }
13 }
```

Figura 13: Regras de leitura e escrita do banco de dados. Fonte: O autor.

Na figura 13 acima é possível visualizar as regras que foram feitas no banco de dados, no nó de usuários apenas é possível escrever quem tem o valor de escrita como verdadeiro, já para leitura todos podem, e no nó pacientes, todos os usuários podem escrever e ler.

O banco de dados é dividido em nós de modo que tenhamos pacientes, que contém dentro dele todos os pacientes cadastrados e dentro desses pacientes, todos os dados e também consultas feitas. Já no nó de usuários é capaz de visualizar as informações de cada usuário do sistema, contendo informações como: e-mail, nome, se ele pode escrever no banco e se é uma conta ativa, ou seja, que pode usar o sistema.

5 RESULTADOS OBTIDOS

Tendo em vista a metodologia apresentada foi feito o desenvolvimento do aplicativo para o projeto "Vida Saudável". Após o desenvolvimento foi realizado alguns testes para medir a usabilidade e a funcionalidade do mesmo, com base em um teste funcional proposto no anexo A e um teste de usabilidade proposto no anexo B.

Para o teste funcional, foi utilizado casos de teste, baseando-se em cenários que foram retirados dos casos de usos apresentados na seção 4.1.1 e nas funcionalidades do aplicativo, e foi realizado pelo autor para medir o quão funcional o aplicativo ficou no término do desenvolvimento e também foi relatado possíveis problemas que foram encontrados durante os testes que devem ser corrigidos em trabalhos futuros.

Para o teste de usabilidade, foram criados cenários com base nos casos de uso principais, cadastrar paciente e cadastrar consultam, para ser executado pelos voluntários para representar possíveis usuários, que depois de fazer os passos dos casos de teste apresentados, responderam um questionário que pode ser encontrado no apêndice C.

5.1. Arquitetura da aplicação com o Firebase

Nessa seção será apresentada o diagrama de desenvolvimento do aplicativo, mostrando a comunicação entre o aplicativo e o Firebase.

Na figura 14 é apresentada a comunicação entre o aplicativo móvel e o Firebase, que sempre é chamado pelo Firebase Core que esse chama o Firebase Authentication ou o Firebase Realtime Database, dependendo do que foi requisitado e da necessidade da funcionalidade.

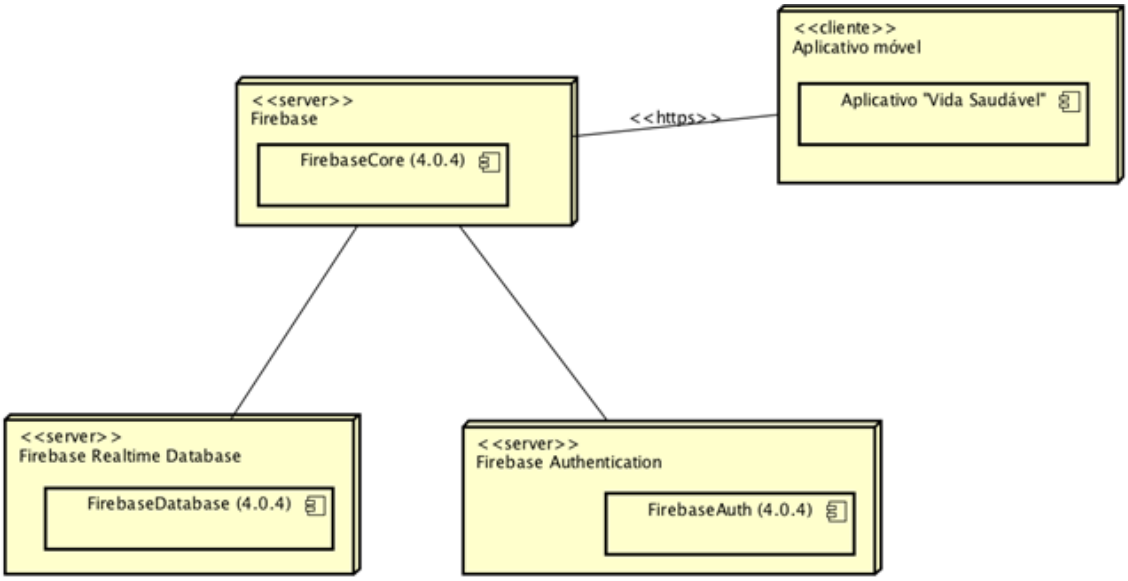

Figura 14: Diagrama de desenvolvimento do aplicativo.

5.2. Diagrama de sequência, diagrama de classe e telas de cada caso de uso

- Cadastrar Paciente

O diagrama de sequência para o caso de uso cadastrar paciente apresenta as interações entre o ator, a aplicação e o Firebase.

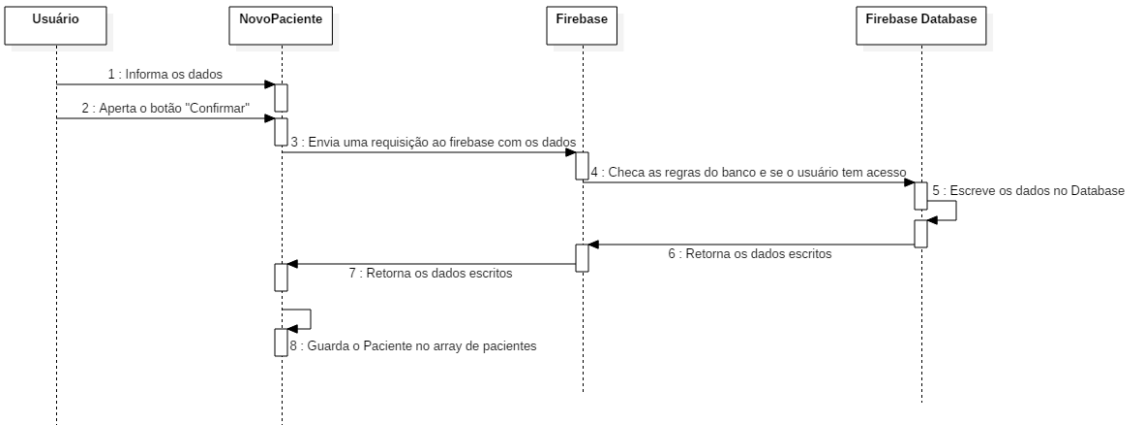

Figura 15: Diagrama de sequência do caso de uso "cadastrar paciente"

Após o ator preencher as informações e clicar no botão, a aplicação requisita a gravação das informações de paciente no banco de dados do Firebase, que retorna as informações que foram gravadas e a aplicação guardar na memória o paciente que acabou de ser adicionado.

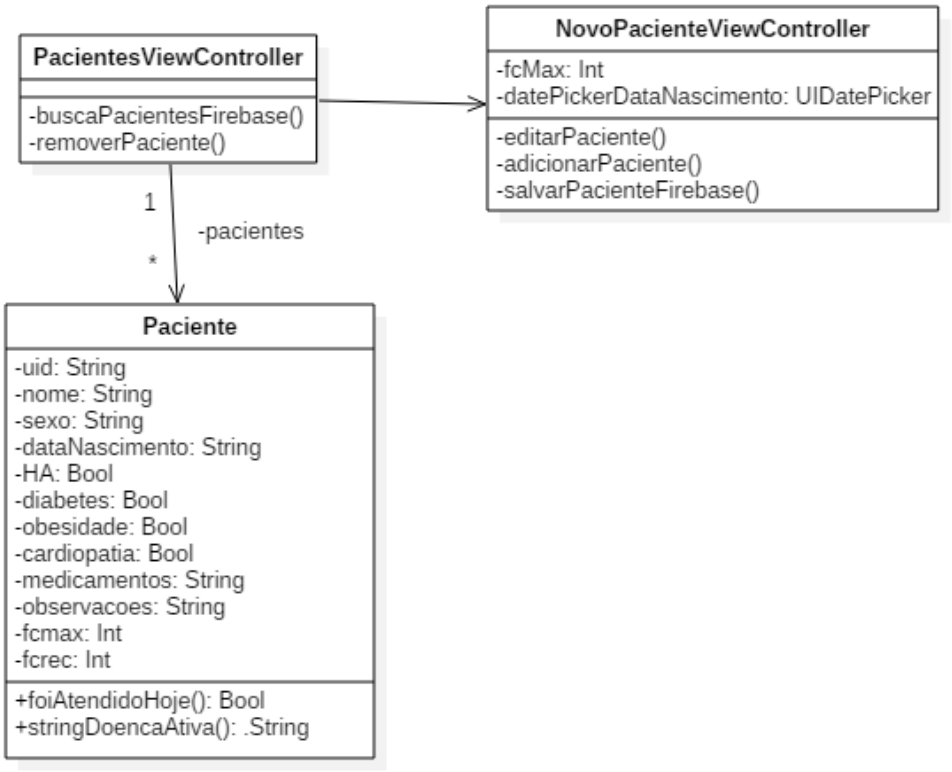

Figura 16: Diagrama de classe de cadastrar paciente.

O diagrama da figura 16 mostra as classes utilizadas no caso de uso de cadastrar paciente, onde é usado o modelo de Paciente e as classes de controle e visualização Pacientes e Novo Paciente, que é onde fica a lógica de cadastro.

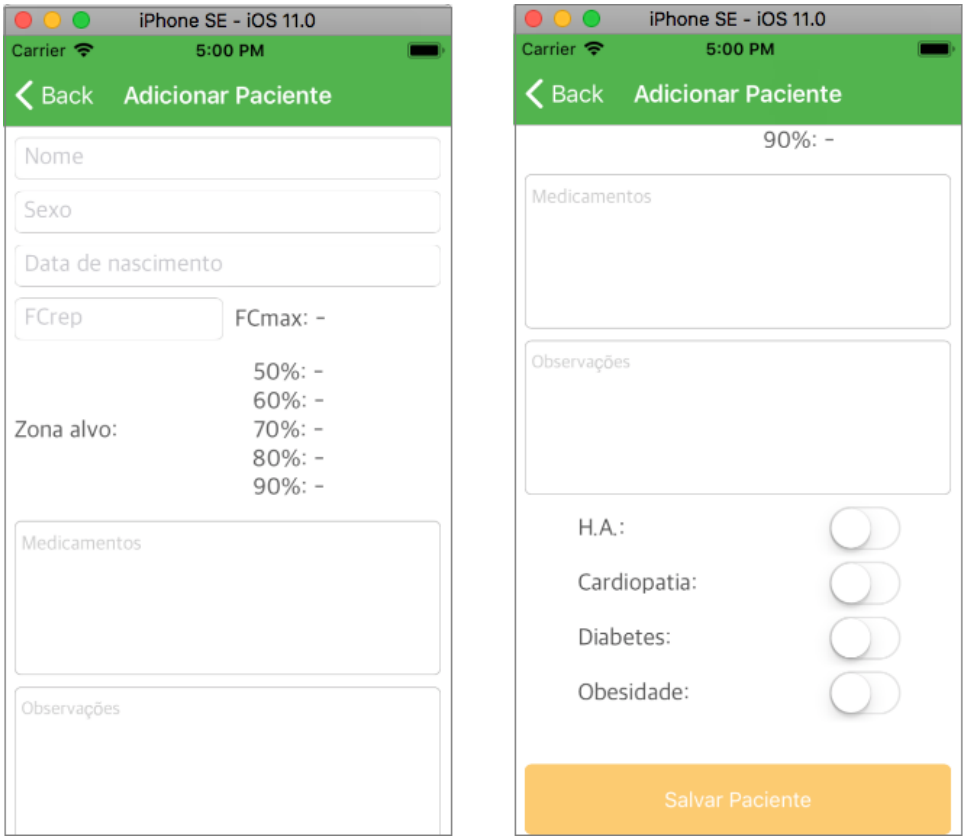

Figura 17: Adicionar paciente. Fonte: O autor.

Na figura 17 é possível visualizar a tela de adicionar paciente, onde pede-se para o aluno do projeto adicionar as informações do paciente, como: nome, sexo, data de nascimento, frequência cardíaca em repouso, entre outras. Depois disso ele pode salvar o paciente.

- Editar Paciente

O diagrama de sequência a seguir mostra as interações entre o Usuário, Editar Paciente, Firebase e Firebase Database.

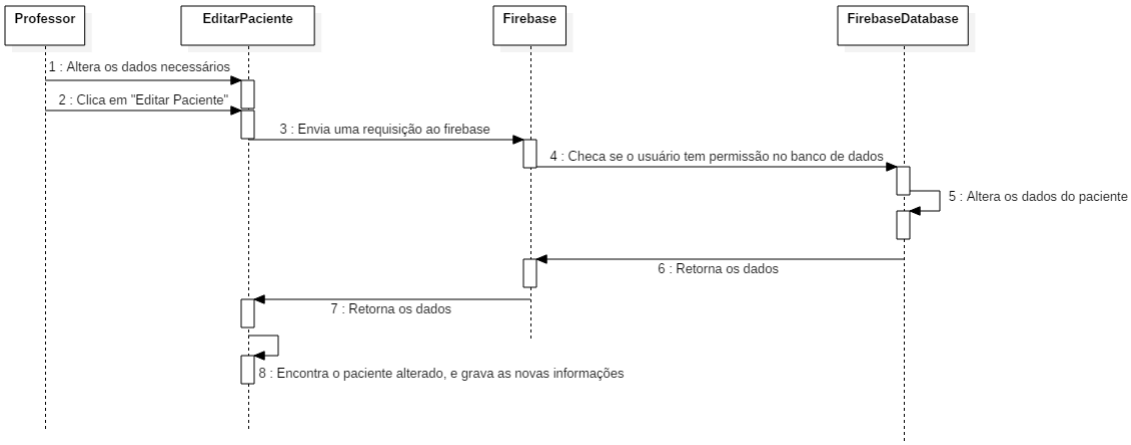

Figura 18: Diagrama de sequência de edição de paciente.

Depois de o usuário informar as alterações do paciente e clicar em "Editar Paciente" o aplicativo envia uma requisição ao firebase, que esse checa as regras de acesso ao banco de dados, que caso seja aceito, altera os dados do paciente e retorna os dados para a aplicação, que guarda as alterações no array de pacientes.

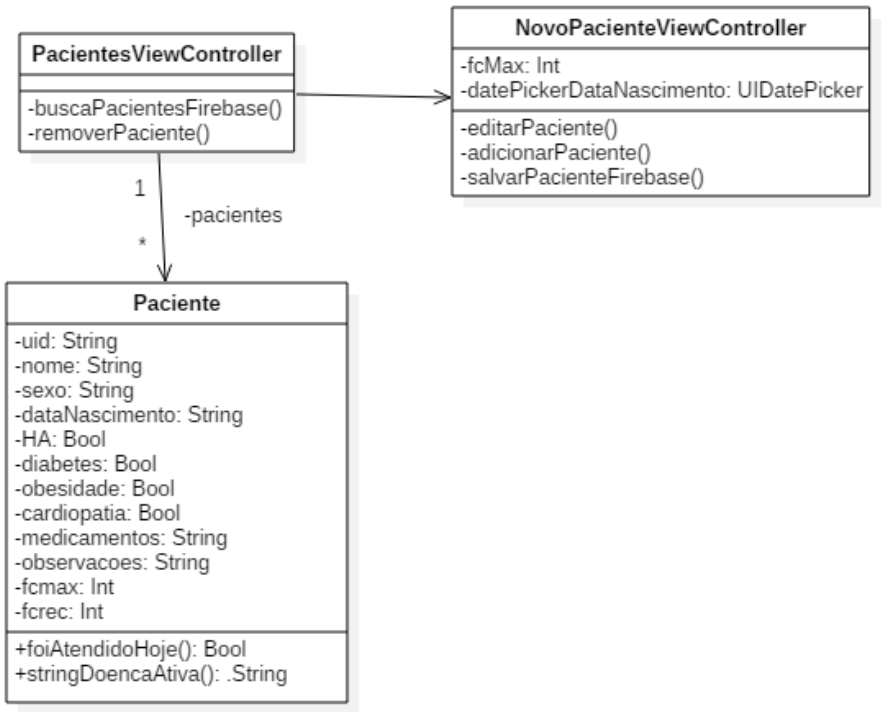

Figura 19: Diagrama de classe de editar paciente.

O diagrama da figura 19 mostra as classes utilizadas no caso de uso de editar paciente, onde é usado o modelo de Paciente e as classes de controle e visualização Pacientes e Novo Paciente, que é onde fica a lógica de edição.

A tela da figura 17 também é utilizada para a edição de um paciente, onde os campos vem preenchidos com os dados do paciente selecionado e depois da alteração deve-se clicar em salvar para manter as alterações.

- Remover Paciente

  O diagrama a seguir mostra as interações entre o Usuário, Pacientes, Firebase e Firebase Database.

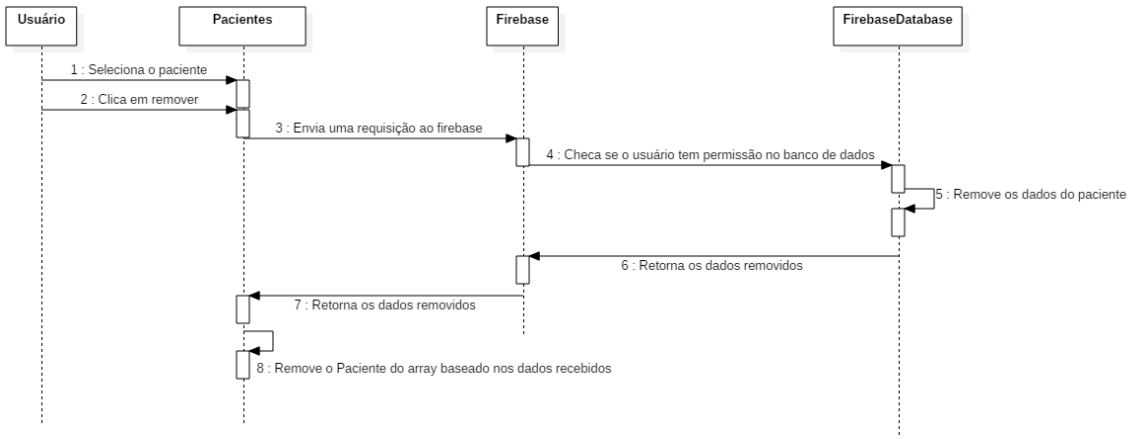

Figura 20: Diagrama de sequência de remoção de paciente.

Depois de selecionar o paciente e clicar em remover a aplicação requisita ao firebase que checa a permissão no banco de dados, caso seja aceito, remove os dados do paciente e retorna esses dados a aplicação que remove o paciente do array de pacientes.

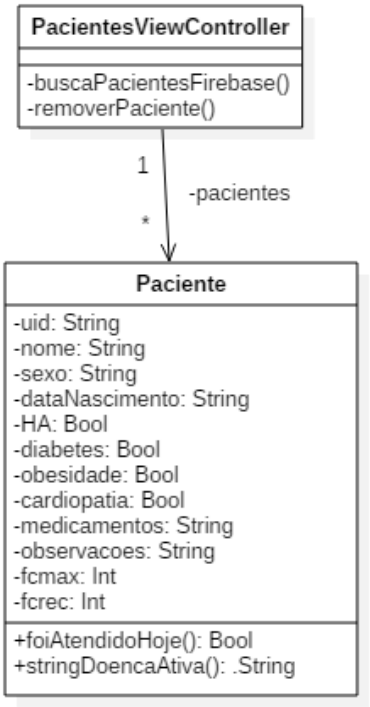

Figura 21: Diagrama de classe de remover paciente.

O diagrama da figura 21 mostra as classes utilizadas no caso de uso de remover paciente, onde é usado o modelo de Paciente e as classes de controle e visualização Pacientes, que é onde fica a lógica de remoção.

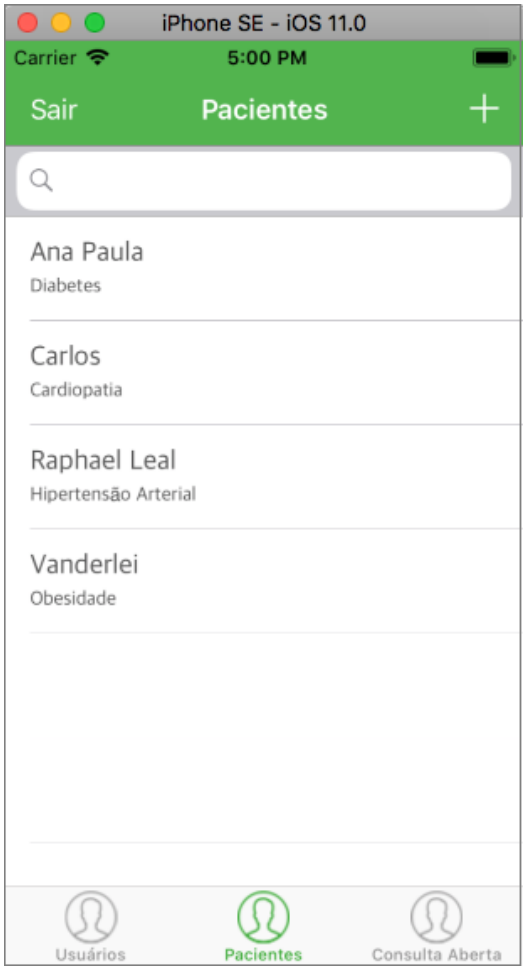

Figura 22: Tela de pacientes. Fonte: O autor

Pode ser visualizada na figura 22 a tela de pacientes, nela é listado todos os pacientes cadastrados por todos os usuários, acima da tabela onde é feita a listagem há um campo de busca onde pode ser feito uma filtragem dos pacientes por nome, facilitando assim a busca de um paciente específico. Além disso, nessa tela, se escolher um paciente e deslizar para a esquerda aparece a opção de remoção do paciente, que ao clicar remove o paciente da lista e as consultas referentes a ele.

- Cadastrar Consulta

O diagrama de sequência para o caso de uso cadastrar consulta apresenta as interações entre o ator, a aplicação e o Firebase.

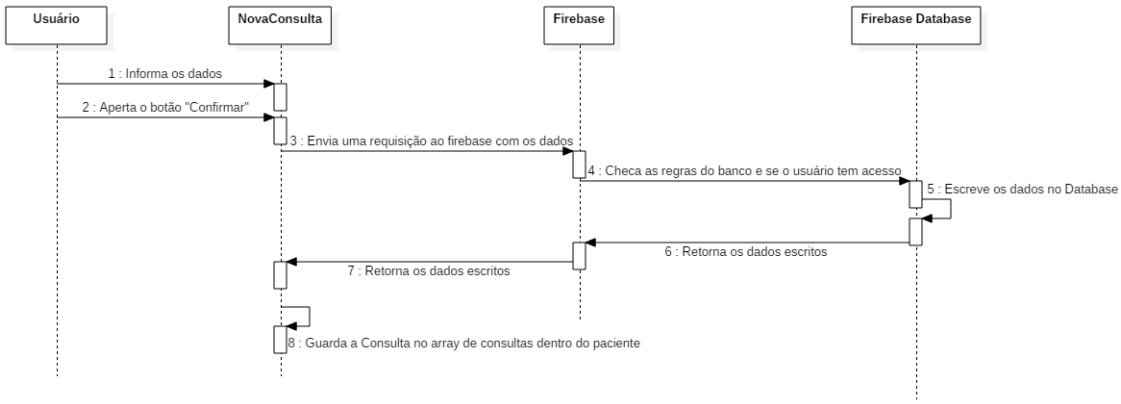

Figura 23: Diagrama de sequência do caso de uso "cadastrar consulta"

O ator preenche todas as informações necessárias e clica no botão "Confirmar", em seguida a aplicação requisita à gravação das informações no banco de dados do Firebase, que retorna as informações gravadas para a aplicação que nesse momento adicionado na memória a consulta que foi gravada.

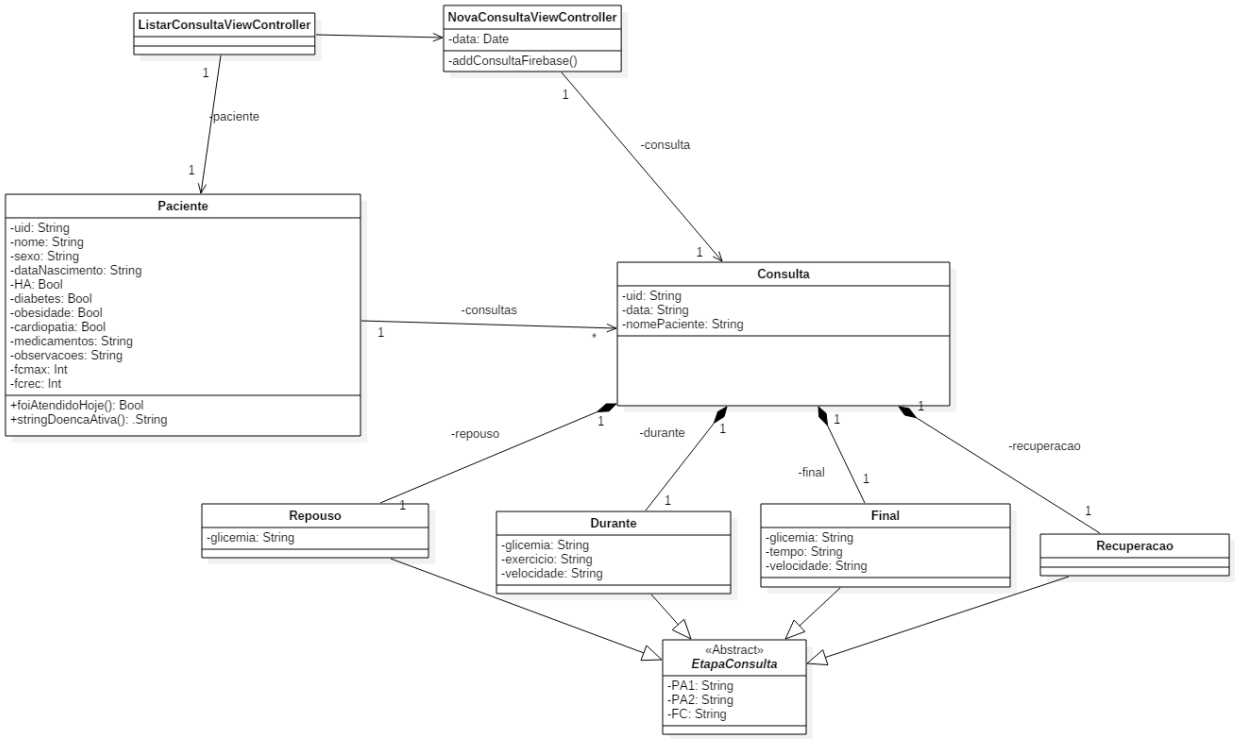

Figura 24: Diagrama de classe de cadastrar consulta

O diagrama da figura 24 mostra as classes utilizadas no caso de uso de cadastrar consulta, onde é usado o modelo de Paciente, Consulta, Repouso, Durante, Final, Recuperação e EtapaConsulta e as classes de controle e visualização Listar Consultas e Nova Consulta, que é onde fica a lógica de cadastro.

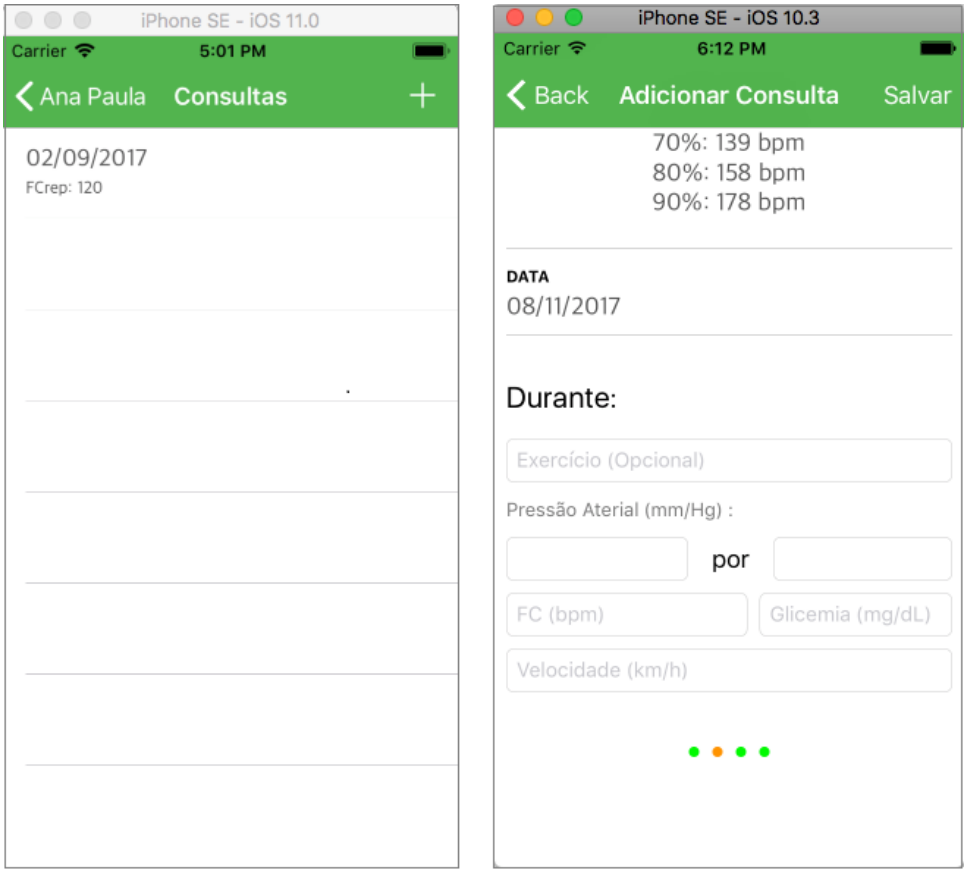

Figura 25: Tela de consultas e adicionar consultas. Fonte: O autor.

A tela da esquerda da figura 25 lista todas as consultas de determinado paciente. Já a tela ao lado é utilizada para criação de uma nova consulta desse paciente específico.

- Remover Consulta

O diagrama a seguir mostra a interação entre o Usuário, Consultas, Firebase e Firebase Database.

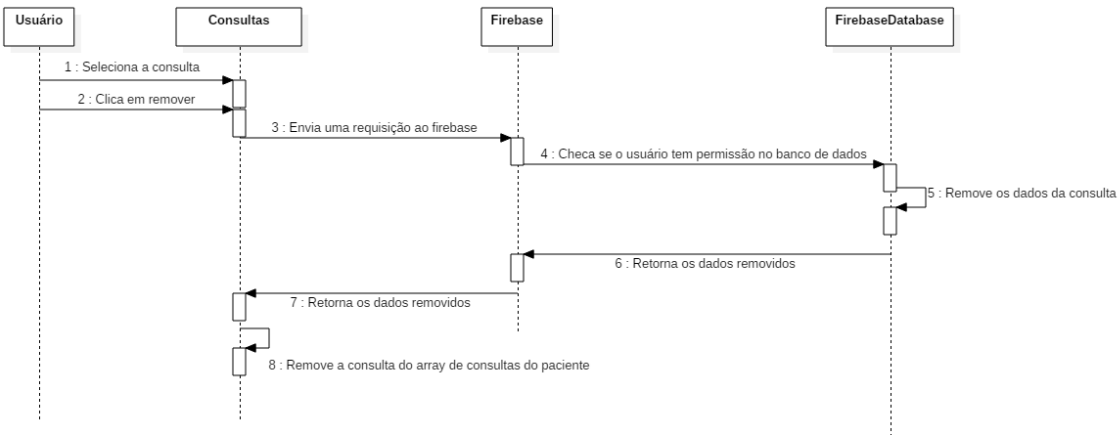

Figura 26: Diagrama de sequência de remoção de consulta.

Depois de selecionar a consulta e clicar para remover, o aplicativo faz uma requisição ao firebase que checa a permissão do banco de dados, caso seja aceito, remove os dados da consulta do paciente, e retorna esses dados ao aplicativo, que remove a consulta do array de consultas daquele paciente.

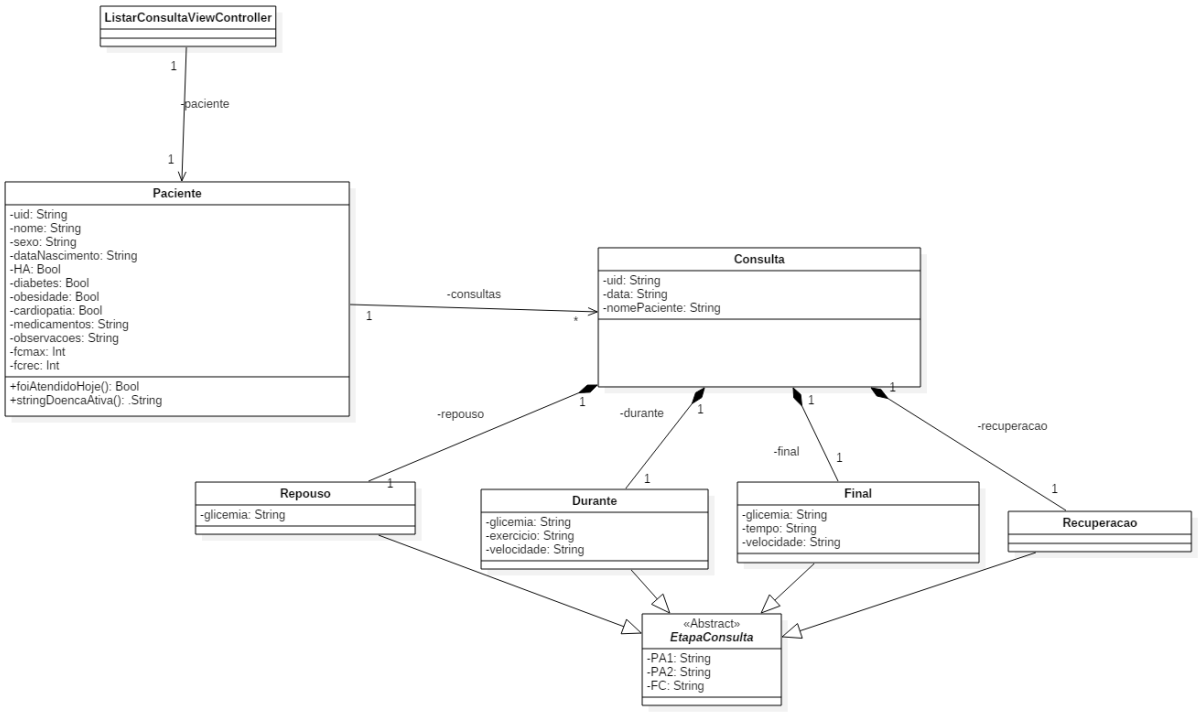

Figura 27: Diagrama de classe de remover consulta

O diagrama da figura 27 mostra as classes utilizadas no caso de uso de remover consulta, onde é usado o modelo de Paciente, Consulta, Repouso, Durante, Final, Recuperação e EtapaConsulta e as classes de controle e visualização Listar Consultas, que é onde fica a lógica de remoção.

A tela da esquerda da figura 25 também é utilizada para a remoção de uma consulta. Seguindo a mesma lógica da remoção do paciente, deve-se deslizar a consulta que deseja remover para a esquerda, onde vai aparecer a opção de remoção e clicando nela a consulta será removida da lista.

- Cadastrar Usuário

  O diagrama a seguir mostra a interação entre o Professor, NovoUsuário, Firebase, Firebase Database.

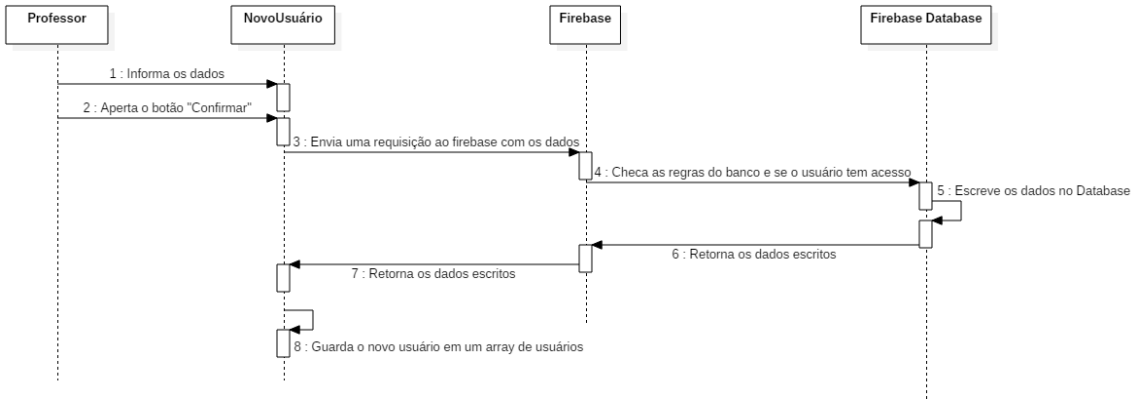

Figura 28: Diagrama de sequência do cadastro de um usuário.

Depois de informar os dados do usuário e clicar em confirmar, o aplicativo requisita o firabase o acesso ao banco de dados, que por sua vez confere as regras do banco de dados, caso seja aceito, escreve os dados no banco de dados e retorna para a aplicação, que guarda esse usuário no array de usuários.

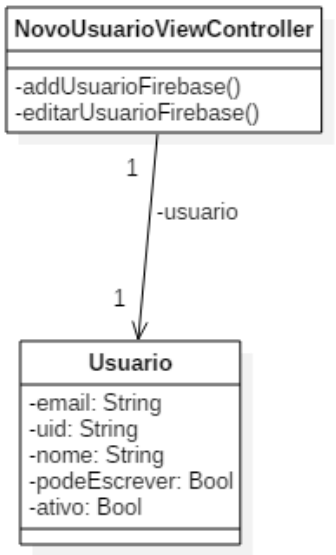

Figura 29: Diagrama de classe de cadastrar usuário.

O diagrama da figura 29 mostra as classes utilizadas no caso de uso de cadastrar usuário, onde é usado o modelo de Usuário e as classes de controle e visualização Novo Usuário, que é onde fica a lógica de cadastro.

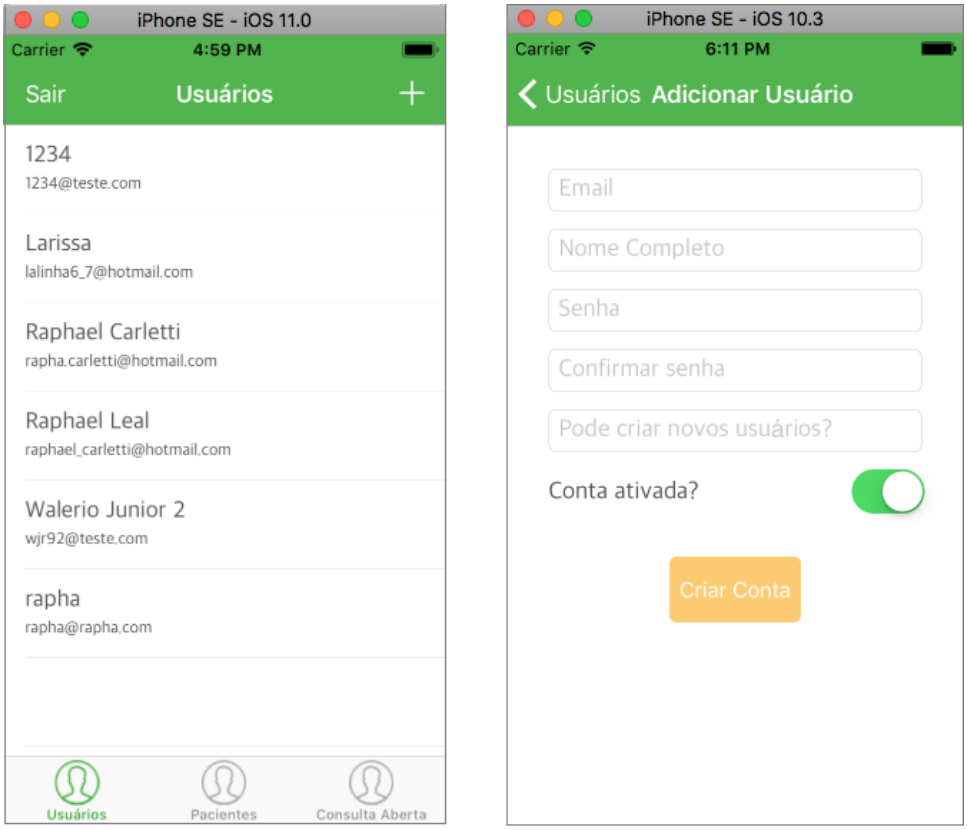

Figura 30: Tela de usuários e de adicionar usuário. Fonte: O autor.

Na tela da esquerda da figura 30 é possível observar a tela de usuários, onde são listados todos os usuários e caso seja selecionado algum seria mostrada todas as informações do mesmo em uma outra tela. Já a tela de adicionar usuário, assim como a de listagem de usuários, só é mostrada para alguém que tenha permissão de escrita, e sua função é para adicionar um usuário incluindo algumas informações.

- Editar Usuário

O diagrama a seguir mostra a interação entre o Professor, EditarUsuário, Firebase, Firebase Database.

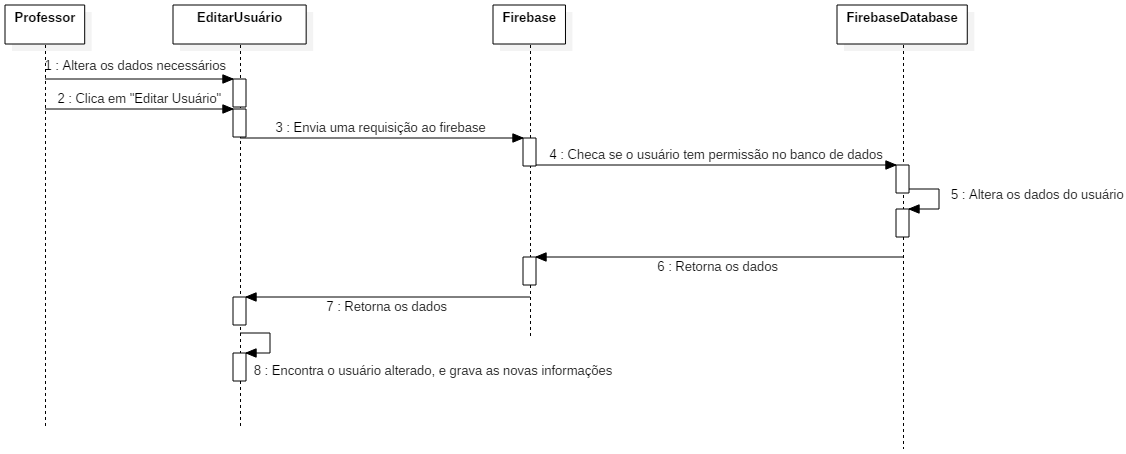

Figura 31: Diagrama de sequência de edição de usuário.

Depois de alterar os dados necessários, o aplicativo envia uma requisição ao firebase que checa a permissão do banco de dados e altera os dados do usuário, caso seja permitido, e retorna os dados para a aplicação que encontra o usuário e modifica os dados na classe Usuário.

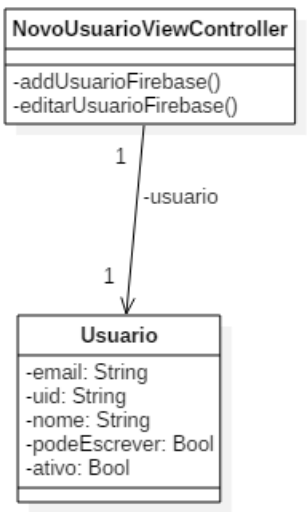

Figura 32: Diagrama de classe de edição de usuário.

O diagrama da figura 32 mostra as classes utilizadas no caso de uso de edição de usuário, onde é usado o modelo de Usuário e a classe de controle e visualização Novo Usuário, que é onde fica a lógica de edição.

A tela da direita da figura 30 também é utilizada para a edição do usuário, em que a tela vem com as informações do usuário seleciona já preenchidas para que pode ser feito a alteração e o botão de criar conta muda a escrita para “salvar” que será onde pode-se clicar para manter as alterações.

- Logar no sistema

O diagrama a seguir mostra as interações entre Usuário, Login, Firebase, Firebase Authentication, Firebase Database, TelaPrincipal.

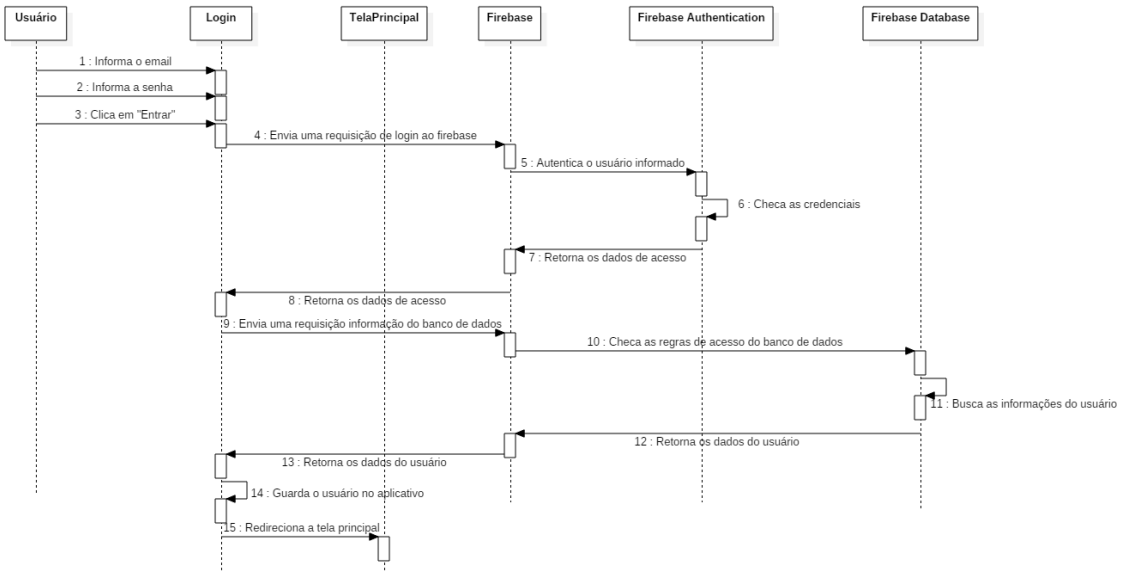

Figura 33: Diagrama de sequência de login de usuário.

Depois de informado email e senha, o usuário clica em "Entrar", que requisita uma autenticação ao firebase, que autentica o usuário e checa as credencias no Firebase Authentication, depois retorna os dados de acesso a aplicação. Após isso, a aplicação faz uma requisição para buscar informações no banco de dados que checa as regras de acesso antes, e caso seja aceito, busca as informações do usuário e as retorna para a aplicação que guarda esses dados na classe Usuário e redireciona o usuário para a tela principal do sistema.

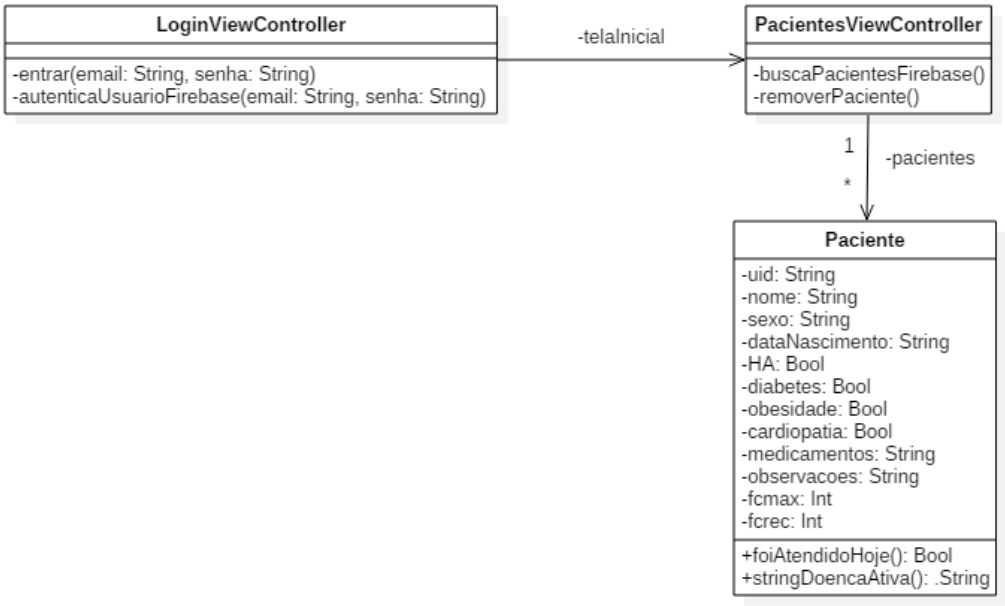

Figura 34: Diagrama de classe de logar no sistema.

O diagrama da figura 34 mostra as classes utilizadas no caso de uso de logar no sistema, onde é usado a classe de controle e visualização Login, que é onde fica a lógica de autenticação.

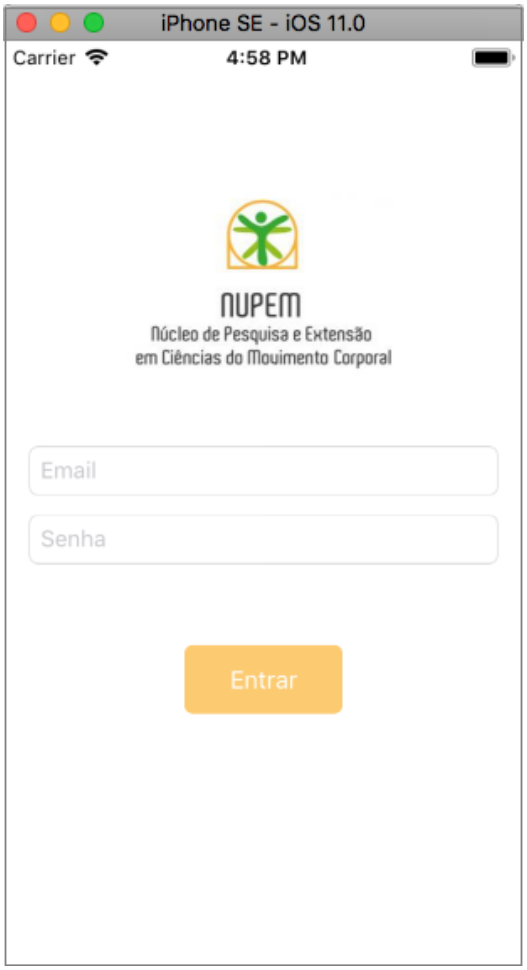

Figura 35: Tela de login. Fonte: O autor.

A tela de login da figura 35 mostra que é necessário utilizar de um e-mail e senha válidos para autenticação do usuário no aplicativo que deve ser cadastrado pelo professor, dentro do aplicativo.

5.3. Teste Funcional

No teste funcional foram executados os 7 casos de teste apresentados no apêndice A, e então foram identificados alguns possíveis defeitos que foram encontrados durante a execução, listados abaixo:

ID defeito	CT	Descrição do defeito	Severidade do defeito
01	002	Após a criação de um novo usuário, o aplicativo não retorna para a tela principal.	médio
02	005	Após a criação de uma nova consulta, o aplicativo retorna a tela inicial, quando na verdade deveria retorna a tela de consultas daquele paciente	médio
03	005	Em algumas vezes aparece uma barra preta no canto inferior da tela	baixo

Quadro 1: Defeitos encontrados no aplicativo.

Como resultado, foi possível perceber que se teve uma média de aproximadamente 0,42 erro por caso de teste, medida que pode ser usado como base para novas versões do aplicativo ou para novos requisitos que podem vir a surgir.

Também foi possível perceber que não houve defeitos graves que atrapalhe a experiência do usuário a ponto de impedir que alguma funcionalidade aconteça. Desses erros, 2 foram de severidade média, que atrapalham o usuário de uma forma, mas não impede o uso do aplicativo e nem de nenhuma funcionalidade e 1 erro de severidade baixa, que seria erros relacionados a estética.

5.4. Teste de usabilidade

No teste de usabilidade foram utilizadas 2 simulações de uso da aplicação que foram elaboradas para testar a aceitação e a facilidade que um usuário que nunca tivesse utilizado o aplicativo teria nas principais funcionalidades do sistema.

Para o questionário pós-teste foram utilizadas as seguintes perguntas:

- Conseguiu concluir todas as tarefas sem grandes dificuldades?
- Em algum momento teve dificuldade sobre qual ação deveria ser realizada?
- Em algum momento ficou sem saber como proceder com a tarefa?
- O aplicativo deixa claro qual o próximo passo para realizar a tarefa?
- Controles e botões se distinguem do restante do *layout*, deixando evidente que são clicáveis?

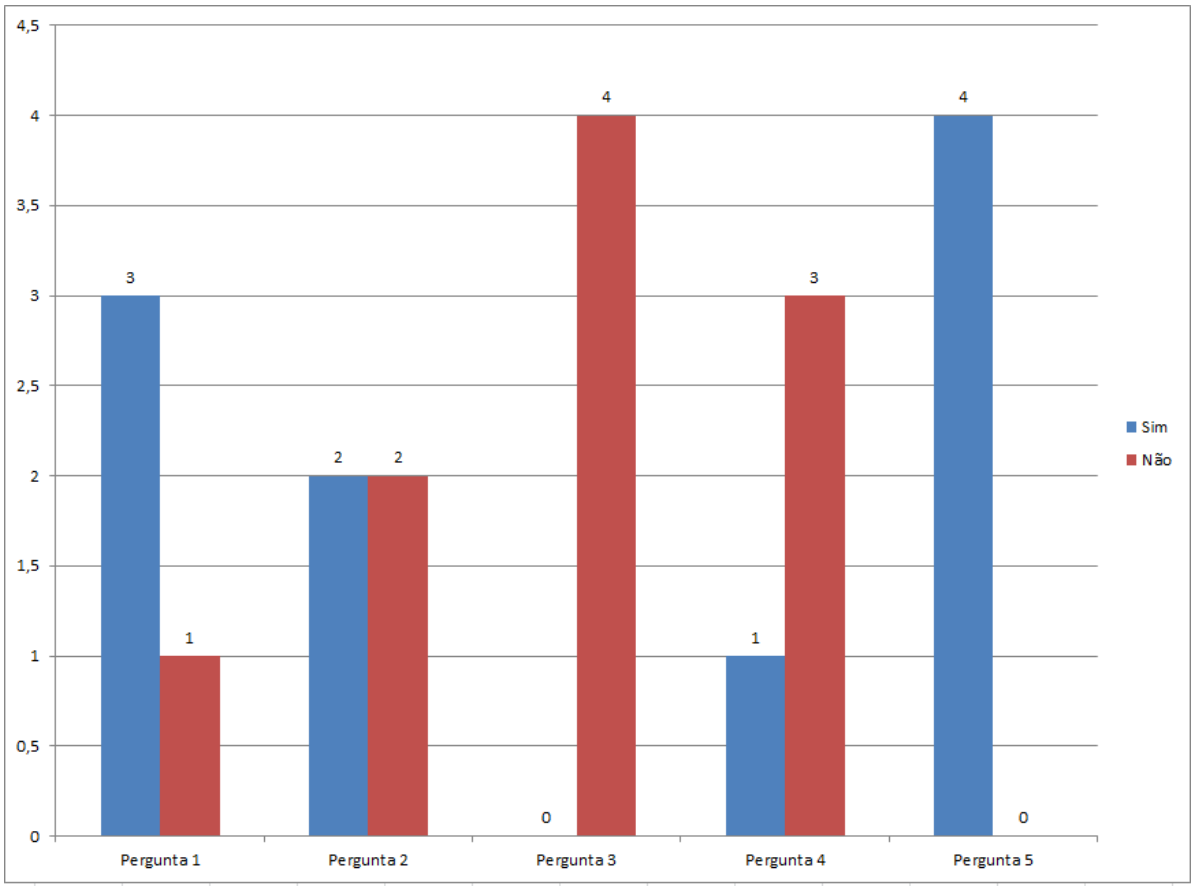

Figura 36: Gráfico do resultado do questionário pós-teste

Como pode ser visto no anexo C, das 4 pessoas que realizaram o teste, apenas 1 não conseguiu executar todas as tarefas sem problemas. Porém, das 3 pessoas que executaram sem grandes problemas, 1 delas teve alguma dificuldade sobre a ação.

Um problema que pode ser observado é que em uma das funcionalidades, os usuários sentiram falta de clareza no que deveria ser feito em seguida. Isso pode ser observado em 3 testadores.

E por fim os participantes, por unanimidade, disseram que os botões se distinguem bem do layout, o que mostra claramente que são clicáveis.

6 CONCLUSÕES

Neste trabalho foi desenvolvido um aplicativo móvel para que possa armazenar as intervenções físicas realizadas no projeto "Vida Saudável", o que permite a visualização remota dessas informações.

O aplicativo contém 2 funcionalidades principais, sendo elas: registro de paciente e registro de consulta de um paciente.

Como constatado pelos testes realizados neste trabalho, o aplicativo se mostrou funcional para o que foi proposto, realizando bem as tarefas necessárias, atingindo dessa forma os objetivos desse trabalho.

7 REFERÊNCIAS BIBLIOGRÁFICAS

APPLE DEVELOPER. **Xcode**. Disponível em: <https://developer.apple.com/xcode/ide/>. Acesso em: 29 nov. 2017.

FIREBASE. **Firebase**. Disponível em: <https://firebase.google.com/?hl=pt-br>. Acesso em: 29 nov. 2017.

FOLHA DE SÃO PAULO. **Número de smartphones em uso no brasil chega a 168 milhões, diz estudo**. Disponível em: <http://www1.folha.uol.com.br/mercado/2016/04/1761310-numero-de-smartphones-em-uso-no-brasil-chega-a-168-milhoes-diz-estudo.shtml>. Acesso em: 29 nov. 2017.

GLOBO G1. **Por que há uma explosão de obesidade no brasil?**. Disponível em: <https://g1.globo.com/bemestar/noticia/por-que-ha-uma-explosao-de-obesidade-no-brasil.ghtml>. Acesso em: 29 nov. 2017.

LEOPOLDO A. S.. **Projeto "vida saudável"**. Disponível em: <http://www.cefd.ufes.br/projeto-vida-saudavel>. Acesso em: 29 nov. 2017.

LOUSHIN, D. **Enterprise knowledge management**: the data quality approach.. [S.L.: s.n.], 2001. 493 p.

LUSTOSA, DANUBIA RAMOS. **Rede social geo educacional - geoeducar**. UNIVERSIDADE DO ESTADO DO AMAZONAS - UEA: [s.n.], 2013. 55 p.

Mekhitarian, P. **Hipertensão: sintomas, tratamentos e causas**. Disponível em: <http://www.minhavida.com.br/saude/temas/hipertensao>. Acesso em: 29 nov. 2017.

OLIVEIRA, A. M. A. D. et al. Sobrepeso e Obesidade Infantil: Influência de Fatores Biológicos e Ambientais em Feira de Santana, BA. **Arquivos Brasileiros de Endocrinologia & Metabologia**, Departamento de Saúde, Universidade Estadual de Feira de Santana (UEFS), Feira de Santana, BA., v. 47, n. 2, abr. 2017. Disponível em: <http://www.scielo.br/pdf/abem/v47n2/a06v47n2.pdf>. Acesso em: 29 nov. 2017.

ORLOV, B. **Ios architecture patterns**. Disponível em: <https://medium.com/ios-os-x-development/ios-architecture-patterns-ecba4c38de52>. Acesso em: 29 nov. 2017.

PORTAL SÃO FRANCISCO. **Cardiopatia**. Disponível em: <http://www.portalsaofrancisco.com.br/saude/cardiopatia>. Acesso em: 29 nov. 2017.

PORTAL SAÚDE. **Número de pessoas com diabetes aumenta em 40% em seis anos.** Disponível em: <http://www.brasil.gov.br/saude/2013/11/numero-de-pessoas-com-diabetes-aumenta-40-em-seis-anos>. Acesso em: 29 nov. 2017.

WENDERLICH, R. **Model-view-controller (mvc) in ios: a modern approach.** Disponível em: <https://www.raywenderlich.com/132662/mvc-in-ios-a-modern-approach>. Acesso em: 29 nov. 2017.

SOCIEDADE BRASILEIRA DE ENDOCRINOLOGIA E METABOLOGIA. **O que é diabetes?**. Disponível em: <https://www.endocrino.org.br/o-que-e-diabetes/>. Acesso em: 29 nov. 2017.

8 ANEXOS

8.1. Anexo A – Teste funcional

Número do CT	001
Caso de Teste	Testar o login do usuário
Procedimento	1. Usuário abre o aplicativo 2. Usuário digita o e-mail e senha válidos 3. Clica no botão “Entrar”
Resultado esperado	Usuário loga no aplicativo com sucesso e aparece a tela inicial

Quadro 2: Caso de teste 001

Número do CT	002
Caso de Teste	Testar a criação de um usuário
Pré-condição	Estar logado em uma conta com permissão para criação de novos usuários
Procedimento	1. Usuário clica na aba “Usuários” 2. Usuário clica no símbolo de mais (+) no canto superior direito 3. Usuário insere um e-mail válido do novo usuário 4. Usuário insere o nome completo desse novo usuário 5. Usuário insere a senha desse usuário com mais de 6 caracteres 6. Usuário repete a senha anterior no campo de “Confirmar senha” 7. Usuário escolhe se o usuário novo tem permissão para criar novos usuários 8. Usuário escolhe se a conta está ativada 9. Usuário clica em “Criar Conta”
Resultado esperado	É criado um novo usuário com essas informações que foram inseridas e que pode ser usado para fazer login no aplicativo, além disso, o usuário retorna a tela de “Usuários”

Quadro 3: Caso de teste 002

Número do CT	003
Caso de Teste	Testar a criação de um paciente
Pré-condição	Estar logado em uma conta
Procedimento	1. Usuário clica na aba “Pacientes” 2. Usuário clica no símbolo de mais (+) no canto superior direito 3. Inserir nome do paciente 4. Escolher sexo do paciente 5. Inserir data de nascimento 6. Inserir sua frequência cardíaca em repouso (FCrep) 7. Informar quais medicamentos ele utiliza, caso houver algum. 8. Informar observações, caso for necessário. 9. Marcar entre as opções em quais doenças ele se enquadra 10. Clicar em “Salvar paciente”
Resultado esperado	O usuário retorna a tela de pacientes, onde é possível ver todos os pacientes cadastrados, inclusive o que ele acabou de cadastrar.

Quadro 4: Caso de teste 003

Número do CT	004
Caso de Teste	Testar a visualização de paciente
Pré-condição	Estar logado em uma conta
Procedimento	1. Usuário clica na aba “Pacientes” 2. Clica em algum paciente cadastro 3. Abre uma nova tela para a visualização de todas as informações do paciente
Resultado esperado	O usuário visualiza uma tela com todas as informações do paciente, sem que haja alguma faltante.

Quadro 5: Caso de teste 004

Número do CT	005
Caso de Teste	Testar a criação de uma consulta
Pré-condição	Estar logado em uma conta e já ter selecionado a visualização de um paciente
Procedimento	1. Usuário clica no botão “Ver todas as consultas” 2. Abre uma nova tela com uma lista de consultas cadastradas (Caso não tenha, mostra uma lista vazia) 3. Clica no botão com símbolo de mais (+) no canto superior direito 4. Abre uma nova tela para preenchimento dos dados da consulta 5. Insere os dados do paciente em repouso. 6. Insere os dados do paciente durante. 7. Insere os dados do paciente no final do exercício. 8. Insere os dados do paciente em recuperação (pós-exercício) 9. Clica no botão “Salvar”
Resultado esperado	O usuário retorna a tela de consultas, onde é possível ver todas as consultas cadastradas para esse paciente, inclusive o que ele acabou de cadastrar.

Quadro 6: Caso de teste 005

Número do CT	006
Caso de Teste	Testar a exibição de uma consulta
Pré-condição	Estar logado em uma conta e já ter selecionado a visualização de um paciente
Procedimento	1. Usuário clica no botão “Ver todas as consultas” 2. Abre uma nova tela com uma lista de consultas cadastradas (Caso não tenha, mostra uma lista vazia) 3. Clica em alguma consulta da lista
Resultado esperado	Abre uma nova tela com as informações da consulta preenchidas corretamente

Quadro 7: Caso de teste 006

Número do CT	007
Caso de Teste	Testar a exibição de uma consulta aberto e salvar essa consulta
Pré-condição	Estar logado em uma conta e ter iniciado uma consulta e voltado para a tela inicial
Procedimento	1. Usuário clica na aba “Consulta Aberta” 2. Abre a aba com a lista de todas as consultas que o usuário abriu e ainda não foi finalizada 3. Clica em uma das consultas em aberto 4. Na tela que abriu com as informações já preenchidas, terminar de preencher o restante dos dados da consulta e clicar no botão “Salvar consulta” 5. A consulta não aparece mais na aba “Consulta aberta” 6. Clica na aba “Pacientes” 7. Seleciona o paciente que foi salva a consulta 8. Clica em “Ver todas as consultas” 9. A consulta está inclusa na tabela
Resultado	O usuário deverá visualizar a consulta que foi salva da

esperado	aba “Consulta aberta” na tela “Ver todas as consultas” do paciente relacionado a consulta salva.

Quadro 8: Caso de teste 007

8.2. Anexo B – Teste de Usabilidade

Tarefa – Adicionar e visualizar um novo paciente

Após abrir o aplicativo:

1. Criar um novo paciente com os seguintes dados:
    - Nome: Raphael Leal Carletti
    - Sexo: Masculino
    - Data de nascimento: 12/01/1995
    - FCrep: 95
    - Medicamentos: Nenhum
    - Observações: Nenhuma
    - Tem Obesidade
2. Visualizar esse paciente adicionado.

Tarefa – Adicionar e visualizar uma nova consulta

Após abrir o aplicativo:

1. Criar uma nova consulta para o paciente “Raphael Leal Carletti” com os seguintes dados:
    a. Repouso:
        - Pressao arterial: 12 por 8
        - Frequencia cardiaca: 90
        - Glicemia apenas para diabeticos
    b. Durante
        - Exercicio: Bicicleta
        - Pressao arterial: 16 por 7
        - Frequencia cardiaca: 160
        - Glicemia apenas para diabeticos
        - Velocidade: 10 km/h
    c. Final:
        - Pressao arterial: 16 por 10
        - Frequencia arterial: 170
        - Velocidade: 10 km/h
        - Tempo: 60 minutos
    d. Recuperacao:
        - Pressao arterial: 13 por 9

- Frequencia cardiaca: 100

2. Visualizar essa consulta adicionada.

8.3. Anexo C – Questionários respondidos sobre o teste de usabilidade

Nome: Testador 1

Assinale com um X a resposta que mais condiz com a realidade.

- Conseguiu concluir todas as tarefas sem grandes dificuldade?
  (X) SIM ( ) NÃO
  1. Em algum momento teve dificuldade sobre qual ação deveria ser realizada?
  (X) SIM ( ) NÃO
  1. Em algum momento ficou sem saber como proceder com a tarefa?
  ( ) SIM (X) NÃO
  1. O aplicativo deixa claro qual o próximo passo para realizar a tarefa?
  (X) SIM ( ) NÃO
  1. Controles e botões se distinguem do restante do *layout*, deixando evidente que são clicáveis?
  (X) SIM ( ) NÃO

Nome: Testador 2

Assinale com um X a resposta que mais condiz com a realidade.

6. Conseguiu concluir todas as tarefas sem grandes dificuldade?
(X) SIM ( ) NÃO
7. Em algum momento teve dificuldade sobre qual ação deveria ser realizada?
( ) SIM (X) NÃO
8. Em algum momento ficou sem saber como proceder com a tarefa?
( ) SIM (X) NÃO
9. O aplicativo deixa claro qual o próximo passo para realizar a tarefa?
( ) SIM (X) NÃO
10. Controles e botões se distinguem do restante do *layout*, deixando evidente que são clicáveis?
(X) SIM ( ) NÃO

Nome: Testador 3

Assinale com um X a resposta que mais condiz com a realidade.

11. Conseguiu concluir todas as tarefas sem grandes dificuldade?
( ) SIM (X) NÃO

12. Em algum momento teve dificuldade sobre qual ação deveria ser realizada?
(X) SIM ( ) NÃO
13. Em algum momento ficou sem saber como proceder com a tarefa?
( ) SIM (X) NÃO
14. O aplicativo deixa claro qual o próximo passo para realizar a tarefa?
( ) SIM (X) NÃO
15. Controles e botões se distinguem do restante do *layout*, deixando evidente que são clicáveis?
(X) SIM ( ) NÃO

Nome: Testador 4

Assinale com um X a resposta que mais condiz com a realidade.

16. Conseguiu concluir todas as tarefas sem grandes dificuldade?
(X) SIM ( ) NÃO
17. Em algum momento teve dificuldade sobre qual ação deveria ser realizada?
( ) SIM (X) NÃO
18. Em algum momento ficou sem saber como proceder com a tarefa?
( ) SIM (X) NÃO
19. O aplicativo deixa claro qual o próximo passo para realizar a tarefa?
( ) SIM (X) NÃO
20. Controles e botões se distinguem do restante do *layout*, deixando evidente que são clicáveis?
(X) SIM ( ) NÃO

8.4. Anexo D – Casos de uso da aplicação

[UC01]: Cadastrar paciente

Descrição: Permite que o ator cadastre um novo paciente no sistema

Fluxo Principal:

[FP00]: O caso de uso se inicia no momento em que o ator clica no botão para ir a tela de cadastro de paciente

[FP01]: O ator digita as informações do paciente.

[FP02]: O ator seleciona a opção “Salvar paciente” para salvar o paciente descrito.

[FP03]: O sistema encaminha o ator para a página inicial.

Fluxo Alternativo:

[FA01]: No [FP02] o ator não selecionou um dos campos obrigatórios, retornar para [FP01] sinalizando que o cadastro não pode ser feito.

[UC02]: Remover paciente

Descrição: Permite que o ator remova um paciente do sistema

Fluxo Principal:

[FP00]: O caso de uso se inicia no momento em que o ator clica no botão para ir a tela de pacientes

[FP01]: O ator arrasta a célula do paciente que quer remover para a esquerda.

[FP02]: O ator seleciona o botão “Remover” para remover o paciente.

[FP03]: O sistema remove o paciente da lista.

[UC03]: Editar paciente

Descrição: Permite que o ator edite um paciente no sistema

Fluxo Principal:

[FP00]: O caso de uso se inicia no momento em que o ator clica no botão para ir a tela de edição do paciente em questão.

[FP01]: O ator edita as informações necessárias.

[FP02]: O ator seleciona a opção “Salvar paciente” para salvar o paciente descrito.

[FP03]: O sistema encaminha o ator para a página inicial.

Fluxo Alternativo:

[FA01]: No [FP02] o ator não selecionou um dos campos obrigatórios, retornar para [FP01] sinalizando que o cadastro não pode ser feito.

[UC04]: Cadastrar consulta

Descrição: Permite que o ator cadastre uma nova consulta para um usuário do sistema

Fluxo Principal:

[FP00]: O caso de uso se inicia no momento em que o ator clica no botão para ir a tela de cadastro de consulta em um paciente

[FP01]: O ator digita as informações da consulta.

[FP02]: O ator seleciona a opção “Salvar” para salvar a consulta nas consultas do paciente selecionado anteriormente.

[FP03]: O sistema encaminha o ator para a página inicial.

Fluxo Alternativo:

[FA01]: No [FP02] o ator não preencheu um dos campos obrigatórios, retornar para [FP01] sinalizando que o cadastro não pode ser feito.

[UC05]: Remover consulta

Descrição: Permite que o ator remova uma consulta de um paciente do sistema

Fluxo Principal:

[FP00]: O caso de uso se inicia no momento em que o ator clica no botão para ir a tela de consultas do paciente selecionado

[FP01]: O ator arrasta a célula da consulta que quer remover para a esquerda.

[FP02]: O ator seleciona o botão “Remover” para remover o paciente.

[FP03]: O sistema remove o paciente da lista.

[UC06]: Cadastrar usuário

Descrição: Permite que o ator cadastre um novo usuário no sistema para utilização do mesmo

Fluxo Principal:

[FP00]: O caso de uso se inicia no momento em que o ator clica no botão para ir a tela de cadastro de usuário

[FP01]: O ator digita as informações do usuário.

[FP02]: O ator seleciona a opção “Criar conta” para salvar o usuário descrito.

[FP03]: O sistema encaminha o ator para a página inicial.

Fluxo Alternativo:

[FA01]: No [FP02] o ator não preencheu um dos campos obrigatórios, retornar para [FP01] sinalizando que o cadastro não pode ser feito.

[UC07]: Editar usuário

Descrição: Permite que o ator edite um usuário no sistema

Fluxo Principal:

[FP00]: O caso de uso se inicia no momento em que o ator clica no botão para ir a tela de edição do usuário em questão.

[FP01]: O ator edita as informações necessárias.

[FP02]: O ator seleciona a opção “Editar conta” para salvar o paciente descrito.

[FP03]: O sistema encaminha o ator para a página inicial.

Fluxo Alternativo:

[FA01]: No [FP02] o ator não selecionou um dos campos obrigatórios, retornar para [FP01] sinalizando que o cadastro não pode ser feito.

[UC08]: Logar no sistema.

Descrição: Permite que o ator tenha acesso as funcionalidades do sistema.

Fluxo Principal:

[FP00]: O caso de uso se inicia no momento em que o ator abre a aplicação.

[FP01]: O ator digita suas credenciais.

[FP02]: O ator seleciona a opção “Entrar” para enviar seus dados.

[FP03]: O sistema verifica os dados digitados.

[FP04]: O sistema encaminha o ator para a pagina inicial.

Fluxo Alternativo: [FA01]: No [FP03] o ator Professor digitou Login, Senha ou o cadastro não existe, retornar para [FP01] sinalizando que o Login não foi efetuado.