



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

Desenvolvimento de um aplicativo de telemedicina

Heron Trindade Fonseca

Monografia apresentada como requisito parcial
para conclusão do Curso de Engenharia de Computação

Orientador
Prof. Dr. Díbio Leandro Borges

Brasília
2021

Dedicatória

Quero dedicar à minha mãe, Helen Trindade Fonseca, ao meu irmão, Hérlon Trindade Fonseca e ao meu filho, Lucas Canelhas Pajuaba Trindade Fonseca; que estiveram sempre presentes na minha jornada até aqui. E deixo a seguinte frase que me motiva a ser alguém fora da curva:

"The people who are crazy enough to think they can change the world, are the ones who do." (Steve Jobs)

Agradecimentos

Gostaria de agradecer ao Prof. Dr. Díbio Leandro Borges que me orientou e possibilitou o desenvolvimento deste projeto. Agradeço, também, à todos os professores da UnB (Universidade de Brasília) que me conduziram a ser o profissional que sou.

Resumo

O projeto tem por objetivo a implementação de um aplicativo para celular que permite a interação entre profissionais de saúde e pacientes por meio da Telemedicina. É necessário que esse aplicativo permita que se realize uma videochamada de maneira segura, a fim de que apenas o paciente e o médico possam participar. Além disso, deve ser possível realizar a marcação de consultas médicas e o armazenamento de arquivos, para que, caso necessários, esses documentos sejam utilizados em um momento oportuno. As ferramentas utilizadas na implementação deste *software*, bem como as funcionalidades e os componentes desse aplicativo são demonstradas ao longo do texto. É importante, também, que se conheça a definição e o contexto da Telemedicina para que o projeto seja devidamente enquadrado neste tipo de serviço. Com isso, este projeto apresentará definições e características da linguagem *JavaScript*, do *framework React Native* e do serviço do Firebase. Por fim, serão apresentados todos os componentes desenvolvidos para a concretização do projeto.

Palavras-chave: Telemedicina, Aplicativo, JavaScript, React Native, Firebase

Abstract

This work presents a software development that is focused on Telemedicine. The purpose of this project is to allow users to make a safe video call, schedule an appointment with healthcare professionals, and upload/download files safely. Therefore, it's important to know some frameworks that will help to develop this work like Javascript, React Native, and Firebase. Then, the paper will explain how this software was developed and how it promoted an easy integration between the users and medical professionals, what is an important thing on Telemedicine.

Keywords: JavaScript, React Native, Firebase, Telemedicine

Sumário

1	Introdução	1
1.1	Telemedicina	1
1.2	Outras aplicações	2
1.3	Objetivos do projeto	3
2	Ferramentas de desenvolvimento	4
2.1	JavaScript	4
2.2	React Native	5
2.3	Firebase	6
2.4	Características do projeto	7
3	Aplicativo de Telemedicina	8
3.1	Fluxo do aplicativo	8
3.2	Configurações do ambiente de teste	9
3.3	Tela de Login	9
3.4	Tela de Cadastro	10
3.5	Tela de Home	11
3.6	Tela de Lista de Médicos	12
3.7	Tela de Descrição do Médico	13
3.8	Tela de Agenda	19
3.9	Tela de descrição da consulta agendada	20
3.10	Tela de ID da vídeo chamada	21
3.11	Tela de chamada de vídeo (médico)	22
3.12	Tela de chamada de vídeo (paciente)	23
3.13	Encerrando a chamada	24
3.14	Tela de arquivos	24
3.15	Escolha do arquivo	25
3.16	Tela de lista de Arquivos	26
3.17	Tela de perfil	27

3.18	Firestore	28
3.18.1	Firestore Authentication	28
3.18.2	Firestore Firestore	29
3.18.3	Firestore storage	30
3.19	Repositório do projeto	31
4	Conclusões	32
	Referências	34

Lista de Figuras

2.1	Ponte entre o código JS e o código nativo.	5
2.2	Tradução no IOS.	6
2.3	Tradução no Android.	6
3.1	Fluxo do aplicativo.	8
3.2	Tela de Login do aplicativo.	10
3.3	Tela de Cadastro no aplicativo.	11
3.4	Tela de Home do aplicativo.	12
3.5	Tela de lista de médicos da especialidade escolhida.	13
3.6	Tela de descrição dos dados do Médico escolhido.	14
3.7	Tela de escolha do dia da consulta.	15
3.8	Tela de escolha do horário da consulta.	16
3.9	Tela de confirmação da data e horário da consulta.	17
3.10	Tela de confirmação de consulta marcada.	18
3.11	Tela de agenda com consultas marcadas.	19
3.12	Tela de dados da consulta agendada.	20
3.13	Tela para ingressar na chamada utilizando o ID.	22
3.14	Tela de início da chamada de Vídeo.	23
3.15	Tela atendimento de telemedicina.	24
3.16	Tela de upload de arquivos.	25
3.17	Tela de escolha de arquivos.	26
3.18	Tela com lista de arquivos salvos pelo usuário.	27
3.19	Tela de perfil do usuário.	28
3.20	Dados de autenticação dos usuários no Firebase.	29
3.21	Coleção de salas para as vídeo chamadas.	30
3.22	Coleção de especialidades médicas.	30
3.23	Coleção de consultas.	31
3.24	Banco de dados do Firebase.	31

Lista de Abreviaturas e Siglas

JS JavaScript.

UI User Interface.

UID ID do usuário.

UX User Experience.

Capítulo 1

Introdução

Este capítulo apresenta uma introdução ao conceito de Telemedicina, além de informações sobre o levantamento feito à respeito de outras aplicações, semelhantes ao que será exposto no texto, e uma explicação sobre a proposta do projeto a ser apresentado.

1.1 Telemedicina

De acordo com o Art. 3o da Lei nº 13.989, de 15 de abril de 2020: Entende-se por telemedicina, entre outros, o exercício da medicina mediado por tecnologias para fins de assistência, pesquisa, prevenção de doenças e lesões e promoção de saúde [1]. Portanto, a telemedicina nada mais é que a prestação de serviços realizada por profissionais de saúde através de meios como computadores, *smartphones*, sistemas de áudio e vídeo, entre outros meios eletrônicos [2]. Este tipo de serviço promove, principalmente, a interação entre profissionais da saúde e pacientes e entre os próprios profissionais da saúde.

Na telemedicina é importante que o profissional tenha alguns cuidados com relação a critérios exigidos neste tipo de serviço como: protocolos de atendimento, informações prévias necessárias, proteção de dados coletados dos pacientes, entre outras questões relacionadas à "Medicina Conectada". Devido à isso, é importante que o especialista na área de saúde esteja sempre bem informado sobre o funcionamento da telemedicina [3].

Levando em conta o atual contexto social, onde as pessoas estão, cada vez mais, tendo um acesso facilitado aos meios eletrônicos cabe afirmar que este serviço merece uma atenção especial visto que consegue atender, a cada dia melhor, a saúde visando o processo de cuidado dos pacientes e dos profissionais, pois com esta tecnologia as fronteiras entre os médicos e pacientes estão sendo minimizadas, além de o profissional poder conseguir acompanhar mais de perto as evoluções dos seus cliente; bem como a gestão dos serviços e sistemas de saúde, já que os processos podem ser cada vez mais centralizados em um único lugar, facilitando o seu gerenciamento.

Dessa forma, mostrando ser uma ferramenta de interação e integração, onde existe a transferência de dados relacionados à saúde, a telemedicina surge como um instrumento importante na facilitação do acesso à saúde e na interação entre profissionais do meio, seja entre eles mesmos, seja com seus pacientes.

Com relação à aplicação desenvolvida, é necessário entender que existem dois tipos de telemedicina: a síncrona (online), na qual médicos e pacientes se conectam de maneira simultânea e interagem em tempo real; e a assíncrona, na qual os pacientes ou médicos enviam ou recebem informações, registros, documentos ou vídeos em momentos diferentes. O *software* desenvolvido abrange as duas formas de telemedicina, permitindo atendimentos online e o armazenamento de arquivos permitindo que estes sejam baixados para o dispositivo utilizado para que seja posteriormente enviado em um momento oportuno.

Por fim, é importante frisar que, com a situação atual da pandemia causada pelo novo coronavírus (SARS-CoV-2), a telemedicina foi autorizada com caráter temporário, enquanto perdurar a emergência da saúde pública, contudo este método está em constante avanço e já existem estudos que promovem a permanência dessa metodologia de maneira definitiva (mesmo após a pandemia). Portanto, a telemedicina promete ser o futuro da medicina no país e no mundo.

1.2 Outras aplicações

Para o desenvolvimento da aplicação proposta no projeto, verificou-se algumas das aplicações já existentes no mercado atual. Os softwares verificados eram todos voltados para a tecnologia mobile, são eles: Doutor Hoje, Doutor Consulta e Boa Consulta.

O Doutor Hoje [4] é um aplicativo mais voltado à marcação de consultas presenciais. Portanto ele possui algumas funcionalidades para isso. Na tela de início existem as possibilidades de se marcar uma consulta, escolhendo dados como especialidade médica, região e o médico da região desejada; pode-se marcar exames, onde possui uma lista de exames com seus tipos, regiões e clínicas dispostas na região escolhida; e possui uma opção para receber um aconselhamento online de um Doutor à disposição por meio de telefone. Além do citado, o aplicativo possui funcionalidades como agenda, com a listagem de agendamentos realizados no aplicativo; pedidos, onde se lista algum pedido feito no aplicativo e os dados da conta do usuário.

O Doutor Consulta [5] é um software que permite a marcação de consultas presenciais, de telemedicina e de exames. Isso tudo é possível realizar na tela inicial e conforme é selecionado o tipo de serviço a ser agendado visualiza-se uma lista com os tipos de consultas ou exames desejados. Quando é selecionado o serviço e o tipo desejado o usuário obtém acesso aos dados do serviço e dados sobre em quais unidades o serviço poderá ser

realizado. Além das funcionalidades citadas é possível verificar o histórico dos serviços agendados, verificar o resultado de exames e obter acesso ao encaminhamento de notas fiscais e recibos.

Por último, o aplicativo Boa Consulta [6] também permite o agendamento de consultas presenciais e de telemedicina. No caso do agendamento de uma consulta presencial, o usuário pode selecionar, inicialmente, qual o convênio que ele possui. Em seguida é necessário selecionar a especialidade médica para então escolher, no mapa do aplicativo, qual a região e médico irá atendê-lo. Na tela de seleção do médico é possível obter dados desse médico e solicitar o agendamento online. No caso de um agendamento de telemedicina, o usuário deve selecionar a especialidade médica e em seguida aparece uma lista com os médicos disponíveis e com as informações das especialidades, horários de atendimento e tempo médio de resposta.

1.3 Objetivos do projeto

O software desenvolvido neste projeto tem como foco a telemedicina. Desse modo, a proposta do projeto é permitir a interação entre um paciente e seu médico de maneira não presencial. Com isso, a primeira necessidade desse software foi a implementação de uma chamada de vídeo onde o paciente e o médico pudessem se conectar de maneira fácil e segura. Outro ponto deste sistema é proporcionar a possibilidade de um usuário escolher o tipo de especialidade médica e qual profissional irá atendê-lo. Para isso, foi desenvolvido um fluxo que possibilita que o paciente escolha a especialidade, o médico, a data e o horário de atendimento que melhor o atende. Por fim, o aplicativo também tem como propósito permitir que os usuários façam o *upload* e *download* de documentos como exames, anamneses e até receitas para que, tanto o médico quanto o paciente possam, posteriormente, analisar os documentos ou até mesmo compartilhar para uma melhor experiência nas consultas. Outro ponto importante desse projeto é poder oferecer todas essas funcionalidades, acima descritas, de maneira segura. Isto se faz necessário para que nenhuma informação privada do cliente ou do médico seja compartilhada de maneira indevida ou indesejada.

Capítulo 2

Ferramentas de desenvolvimento

Este capítulo descreve sobre as linguagens e ferramentas utilizadas no desenvolvimento do projeto.

2.1 JavaScript

JavaScript® (às vezes abreviado para JS) é uma linguagem leve, interpretada e baseada em objetos com funções de primeira classe, mais conhecida como a linguagem de script para páginas Web, mas usada também em vários outros ambientes sem browser, tais como node.js, Apache CouchDB e Adobe Acrobat . O *JavaScript* é uma linguagem baseada em protótipos, multi-paradigma e dinâmica, suportando estilos de orientação a objetos, imperativos e declarativos (como por exemplo a programação funcional)[7]. Utilizada para aplicações *front-end* e *back-end* essa linguagem se mostra indispensável para quem quer aprender a desenvolver web e mobile *softwares*.

Baseada na análise realizada, em maio de 2021, no site da TIOBE, empresa holandesa especializada em avaliação e rastreamento de qualidade de *software* [8], a linguagem *JavaScript* aparece em sétimo lugar como uma das linguagens de programação mais populares do mundo. Isso é um bom indicativo, pois deve-se levar em conta que quanto mais uma linguagem é utilizada, mais ela possuirá suporte e ferramentas de aprimoramento, o que facilita, cada vez mais, a utilização desta. Além disso, quanto maior a comunidade que utiliza a linguagem, maiores serão as possibilidades de encontrar ferramentas precedentes que auxiliarão no desenvolvimento de novos programas.

Por ser uma linguagem amplamente utilizada, o JavaScript possui diversos frameworks que auxiliam no desenvolvimento de *softwares* tanto no lado do *front-end*, quanto no *back-end*. Dessa forma, pode-se listar algumas dessas ferramentas baseadas na linguagem como: jQuery, D3.js, Babel, Bootstrap, React, entre muitas outras. Neste projeto, por

ser voltado para uma aplicação mobile, foi escolhida a ferramenta *React Native*, que possibilitou todo o desenvolvimeto *front-end* do projeto.

2.2 React Native

Criado pelo *Facebook*, o *React Native* é um *framework*, que possui como linguagem base o JavaScript, utilizado para desenvolver aplicativos mobile para sistemas Android e IOS de forma híbrida e nativa. Segundo a descrição no repositório do *facebok* no github: “o React Native React Native traz a estrutura de UI declarativa do React para iOS e Android. Com o React Native, você usa controles de UI nativos e tem acesso total à plataforma nativa” [9]. Isso significa que apesar da ferramenta utilizar o *JavaScript* como linguagem, ele consegue transformar esse código em código nativo dos sistemas mobile como: Java/Kotlin, no caso do Android, e Swift/Obective-C, no caso do IOS.

Para que o *framework* pudesse funcionar de forma nativa, foi criada uma arquiteutura que utliza uma *bridge* (ponte) que permite a comunicação bidirecional e assíncrona entre o código nativo e o código *JavaScript* [10] como demonstra a Figura 2.1.

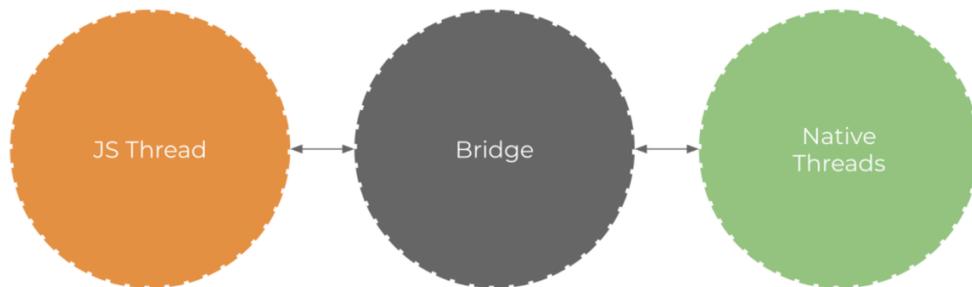


Figura 2.1: Ponte entre o código JS e o código nativo.

Dessa maneira, os comandos descritos em *JavaScript* são processadas e transformadas em JSON. Após as transformação, a bridge recebe os dados e envia a mensagem para o código nativo. Com isso, o código nativo, recebe a mensagem e processa construindo o código recebido na linguagem do sistema mobile (Android ou IOS).

O fluxo dessa tradução da linguagem JS para a linguagem nativa do sistema pode ser representado pelos dois fluxos representados na Figura 2.2 e na Figura 2.3:

Além do citado, o *React Native* também é conhecido por ser um *framework* baseado em componentes. Dessa maneira, é possível criar componentes encapsulados que gerenciam o próprio estado e, assim, pode-se combinar vários desses componentes a fim de desenvolver UIs complexas, aumentando a capacidade do *software* de modo a acrescentar facilidades e ferramentas para o projeto. Devido a isso, existem inúmeras bibliotecas que auxiliam no



Figura 2.2: Tradução no IOS.



Figura 2.3: Tradução no Android.

desenvolvimento dos programas a fim de agilizar e contribuir com desenvolvimento destes. Como exemplo pode-se citar as seguintes bibliotecas: *React Material UI*, que possui diversas ferramentas de UI que auxiliam na construção de telas e páginas dos aplicativos; *React Native Mapview*, que auxilia na visualização e utilização do Google Maps API; *Uniform*, que ajuda no desenvolvimento de formulários; *Redux*, responsável por reunir e gerenciar variáveis de estado nos projetos; entre muitas outras bibliotecas que permitem que esta ferramenta baseada em JS seja uma das ferramentas mais completas e mais fáceis de se utilizar nos dias atuais [11] [12] [13].

Neste projeto de telemedicina, o *React Native* se fez importante para o desenvolvimento de toda a parte *front-end*. Para isto, foram utilizadas algumas das ferramentas e bibliotecas disponibilizadas por este *framework*.

2.3 Firebase

Com a utilização do *framework React Native* para a parte *front-end* do sistema, se fez necessário achar uma alternativa para o *back-end* do aplicativo. Com isso, optou-se por utilizar a base disponibilizada na plataforma *Firebase*.

Firebase é uma plataforma de desenvolvimento mobile (e web) adquirida pela Google em 2014. Com foco em ser um *back-end* completo e de fácil usabilidade, essa ferramenta disponibiliza diversos serviços diferentes que auxiliam no desenvolvimento e gerenciamento de aplicativos [14]. Dentre os serviços disponibilizados, podemos citar alguns como: *Realtime Database*, banco de dados que permite a sincronização de todos os dispositivos

conectados em tempo real; *Storage*, o qual armazena dados dos usuários permitindo o *upload* e *download* desses dados de uma maneira segura; *Authentication*, que possibilita a autenticação de usuários na plataforma através de métodos como *login* pelo google, facebook, github, entre outros. Esses são alguns dos serviços que o firebase proporciona àqueles que pretendem ter um serviço de *back-end* de maneira segura e sem se preocupar em desenvolver toda a estrutura de banco de dados e de conexões de um aplicativo.

Cabe citar que o Firebase trabalha com banco de dados não relacional, ou seja, não utiliza SQL(Linguagem de Consulta Estruturada). Desse modo, o modelo utilizado é o NoSQL, que não utiliza uma estrutura de tabelas e registros, no caso do Firebase a estrutura utilizada é a de uma árvore JSON. Ao adicionar dados à árvore, eles se tornam um nó na estrutura JSON com uma chave associada. Dessa maneira é possível fornecer suas próprias chaves, como códigos de usuário e nomes semânticos, ou gerá-las utilizando comandos como o *push()* [15].

2.4 Características do projeto

Com o exposto neste capítulo, pode-se entender quais tipos de ferramentas foram utilizadas para o desenvolvimento do projeto. A linguagem *JavaScript* serviu como base para a utilização da ferramenta *React Native*, que permitiu o desenvolvimento de toda a parte *front-end* do aplicativo. Com isso, todas as telas e toda a parte de UI que será visto no próximo capítulo foi construída com esses mecanismos de *front-end*. No *back-end* foram utilizados os serviços do Firestore, onde desenvolveu-se em cima das ferramentas de *Realtime database*, *storage* e *authentication*. Dessa forma, pôde-se armazenar dados em tempo real, armazenar dados estáticos e autenticar os usuários do *software* desenvolvido neste projeto.

Capítulo 3

Aplicativo de Telemedicina

Este capítulo desmonstra todo o aplicativo desenvolvido no projeto. As telas de *front-end* e informações do *back-end* serão ilustradas e comentadas seguindo o fluxo de usabilidade do *software*.

3.1 Fluxo do aplicativo

Para o início do desenvolvimento do aplicativo, foi construído um fluxograma das telas do projeto, como demonstrado na Figura 3.1, de forma que se pudesse entender qual seria o fluxo de desenvolvimento a ser seguido e como seria a usabilidade do usuário.

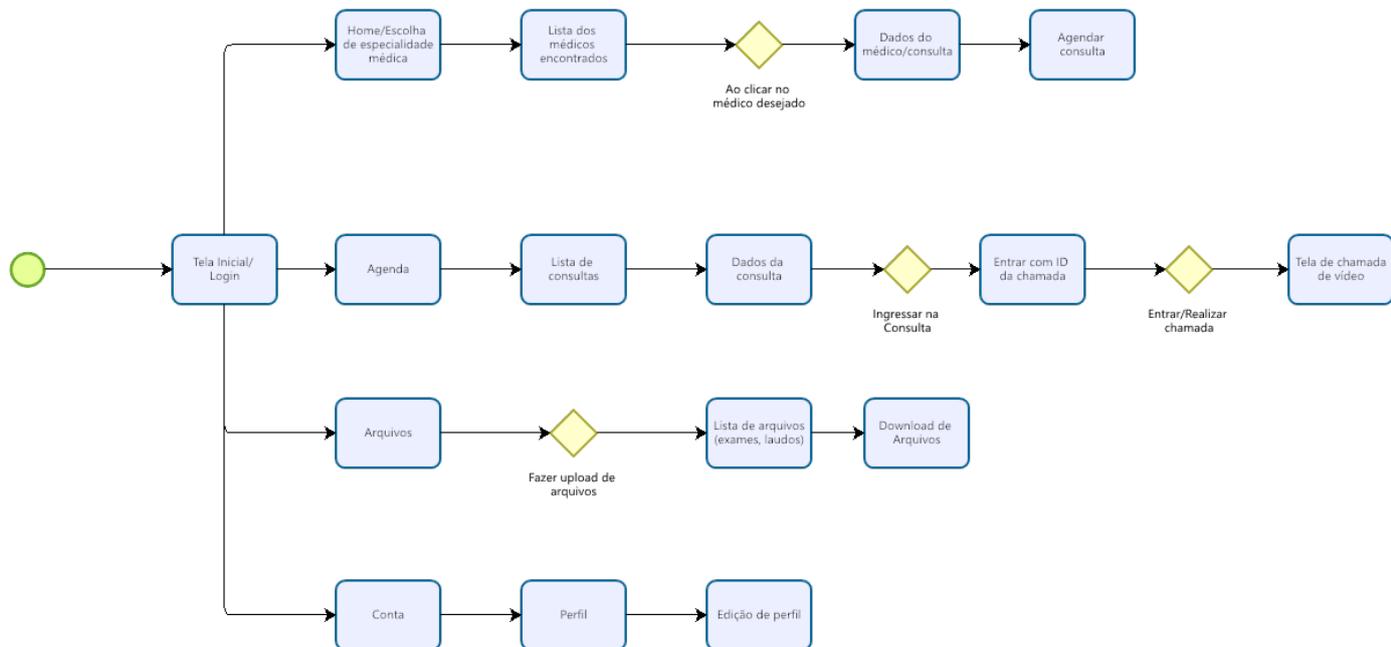


Figura 3.1: Fluxo do aplicativo.

3.2 Configurações do ambiente de teste

A execução do aplicativo foi realizada no ambiente do *Android Virtual Device Manager*, que é proporcionado pelo *Android Studio*. Dentro desse ambiente, foi configurado um dispositivo virtual com as seguintes características:

- *Name: Nexus 6 API 28;*
- *Resolution: 1440 x 2560: 560dpi;*
- *API: 28;*
- *Target: Android 9.0 (Google APIs);*
- *CPU/ABI: x86;*
- *Size on Disk: 4,6 GB.*

3.3 Tela de Login

Na tela de Login é apresentado um Logo do aplicativo e contém campos como “usuário” e “senha” para que o usuário possa entrar com seus dados e fazer o *login* no aplicativo. Além desses componentes, existe também o botão para entrar no aplicativo. Caso os dados inseridos nos campos estejam corretos, o usuário é redirecionado para a tela “home” do aplicativo. Porém, se os dados estiverem incorretos, aparece uma mensagem de erro e o usuário não será redirecionado para a próxima tela. No caso de o usuário não possuir um cadastro no aplicativo, nesta tela existe o botão de “Cadastrar” que irá redirecionar para uma tela de cadastro de usuários.

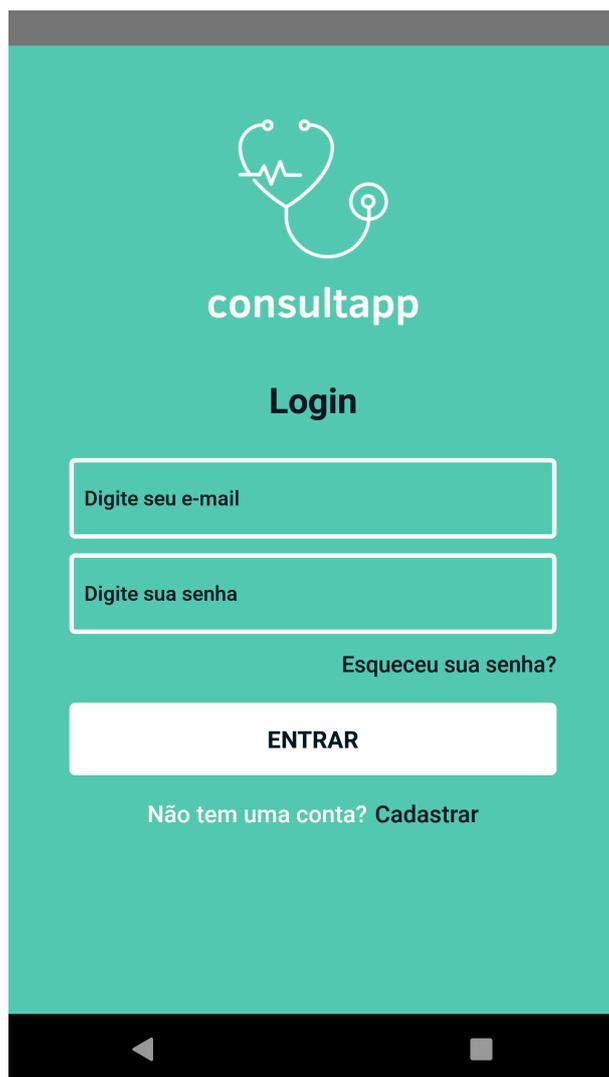


Figura 3.2: Tela de Login do aplicativo.

3.4 Tela de Cadastro

Na tela de Cadastro é apresentado ao usuário um formulário com os campos de “Nome”, “Email”, “Senha” e “Confirmação de Senha”. Dessa forma, é necessário preencher o formulário para a criação de uma conta no aplicativo. Caso os dados sejam preenchidos todos de maneira correta, ao pressionar o botão “Enviar”, o usuário irá criar uma nova conta e será redirecionado diretamente para a tela “Home” do aplicativo. Se algum dado do formulário não for preenchido, uma mensagem de “Preencha todos os campos” será exibida e não ocorrerá o redirecionamento de tela. Este caso ocorre, também, se não houver a concorrência dos campos de “Senha” e “Confirme sua senha”.

← Cadastrar

Informe seu Nome:
name

Informe seu Email:
Email

Informe sua Senha:
Senha

Confirme sua senha:
Confirme sua senha

ENVIAR

Figura 3.3: Tela de Cadastro no aplicativo.

3.5 Tela de Home

Uma vez logado no aplicativo, o usuário visualizará, primeiramente a tela de “Home”. Nesta tela, é exibida uma lista com todas as especialidades médicas que podem ser selecionadas a fim de procurar o profissional desejado para uma marcação de consulta na telemedicina. Pode-se observar um menu na base da tela do aplicativo, o qual permite que o usuário navegue para outras telas como: Agenda, Arquivos, e Perfil; Além da tela de “Home”, o qual o usuário já se encontra.



Figura 3.4: Tela de Home do aplicativo.

3.6 Tela de Lista de Médicos

Nesta tela, é exibido ao usuário uma lista de médicos da especialidade profissional inicialmente selecionada. No exemplo da Figura 3.5, a especialidade selecionada foi “Ortopedia”, portanto a lista de médicos apresentada, mostra quais médico desta área estão disponíveis para uma possível marcação de consulta.

Em cada um dos cartões que apresenta os dados do médico, observa-se os seguintes dados: Nome do médico, especialidade e horários que estão disponíveis para o atendimento.

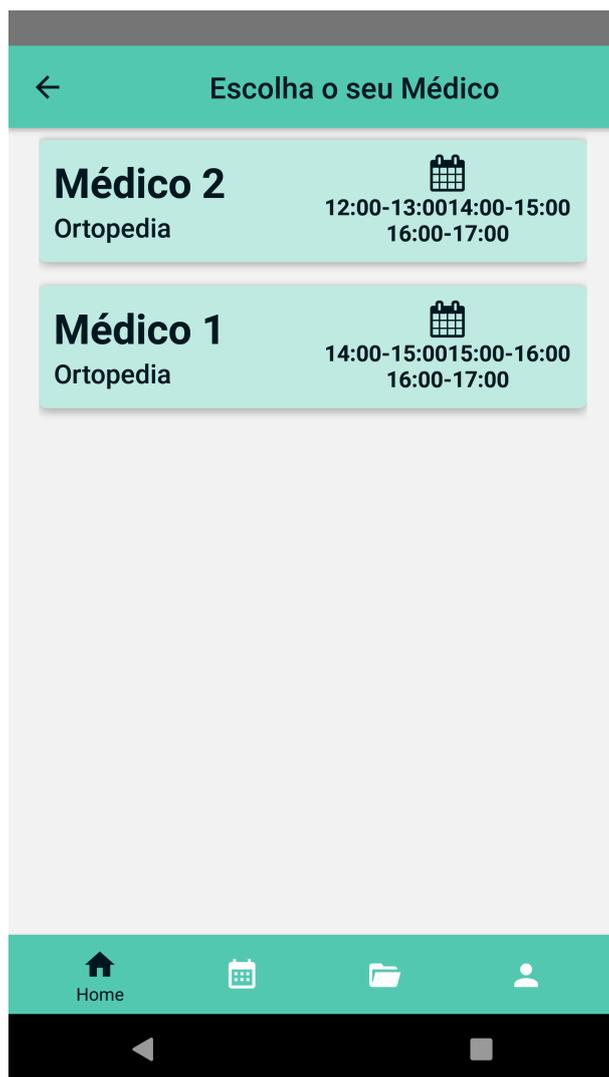


Figura 3.5: Tela de lista de médicos da especialidade escolhida.

3.7 Tela de Descrição do Médico

Após a seleção do médico para o agendamento da consulta, é apresentado ao usuário, na Figura 3.6, uma breve descrição de alguns dados do profissional selecionado. Esse dados são os seguintes: Foto do médico, nome, CRM do profissional e horários de atendimento.

Além dos dados apresentados, são exibidos dois botões para que o usuário escolha a data o qual pretender marcar a consulta com o médico apresentado. No primeiro botão é possível escolher o dia da consulta, como apresentado na Figura 3.7.

No segundo botão, o usuário pode selecionar o horário que deseja marcar a consulta, como demonstrado na Figura 3.8.

Em seguida, depois de escolher o dia e o horário, ao apertar o botão “Agendar Consulta”, é exibida uma modal apresentando os dados selecionados pelo paciente e um botão



Figura 3.6: Tela de descrição dos dados do Médico escolhido.

para que seja confirmada a marcação da consulta, como demonstrado na Figura 3.9.

Por fim, ao confirmar o agendamento, o paciente recebe uma mensagem confirmando que sua consulta foi agendada, de acordo com a Figura 3.10.

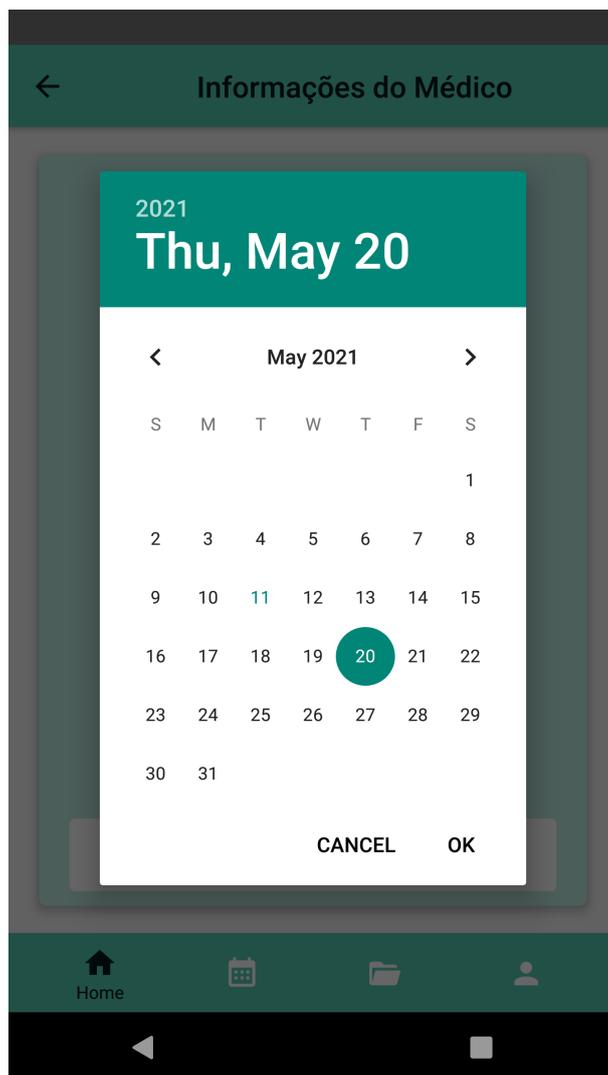


Figura 3.7: Tela de escolha do dia da consulta.

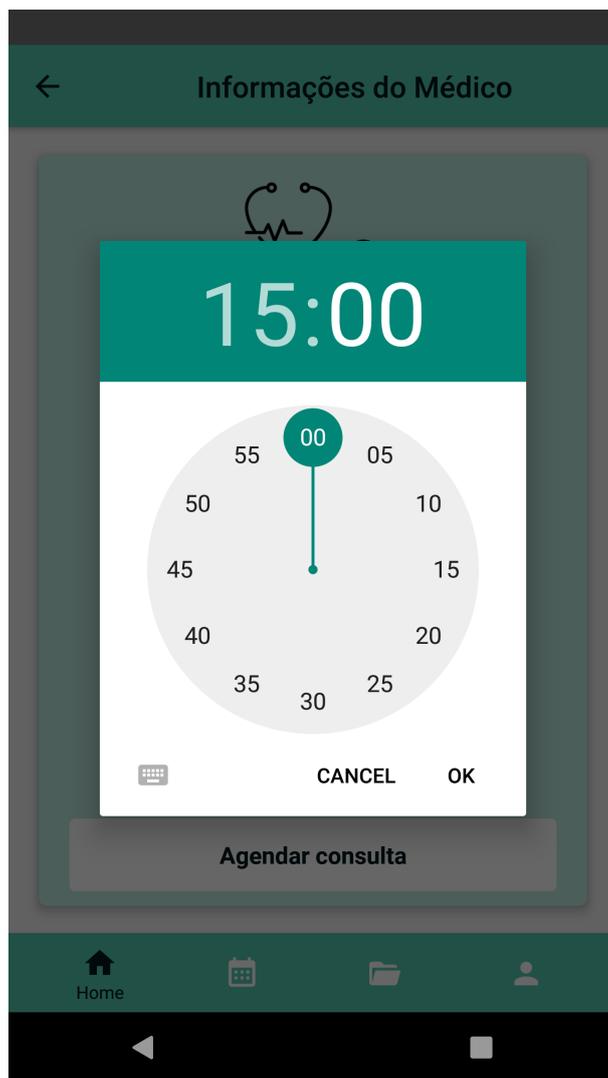


Figura 3.8: Tela de escolha do horário da consulta.



Figura 3.9: Tela de confirmação da data e horário da consulta.

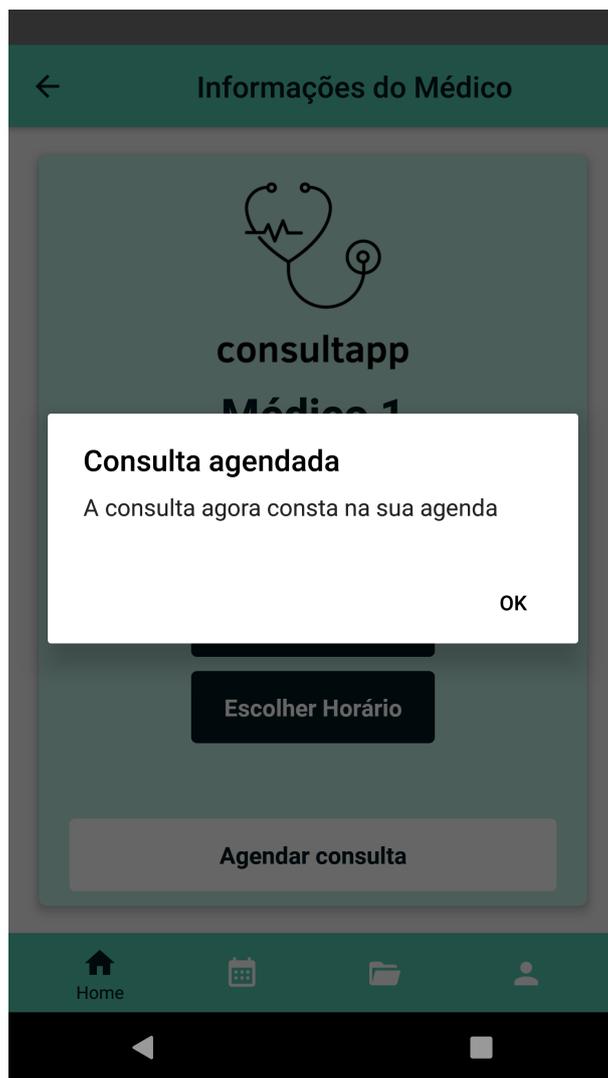


Figura 3.10: Tela de confirmação de consulta marcada.

3.8 Tela de Agenda

A tela de Agenda é a responsável por demonstrar todos os agendamentos realizados pelo usuário. Contudo, além de exibir as consultas marcadas, o paciente pode entrar na consulta, ver os dados e ingressar numa chamada de vídeo para que ocorra a telemedicina. Esta é a tela mais importante do aplicativo, pois é a partir dela que vai ocorrer a interação entre o profissional da saúde e o paciente. Portanto, inicialmente, nesta tela é apresentado um cartão com dados como: Especialidade médica, nome do e, por fim, a data e horário marcados para esta consulta, como pode-se observar na Figura 3.11.

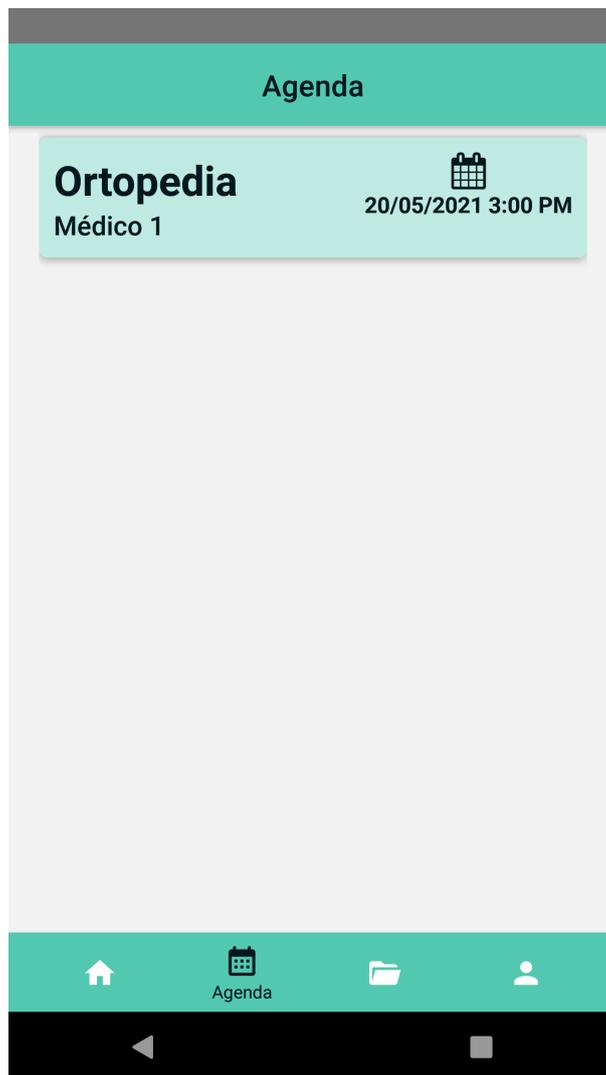


Figura 3.11: Tela de agenda com consultas marcadas.

3.9 Tela de descrição da consulta agendada

Ao selecionar a consulta agendada, é apresentada uma tela contendo a descrição dos dados como: Foto do médico, nome, data e hora da consulta e ID da chamada. Esta última informação é muito importante, pois ela que permite a conexão, tanto do paciente, quanto do médico na vídeo chamada da consulta marcada. Esse ID é o que torna esta consulta em uma chamada privada e segura. Além das informações descritas acima, no final da página, contém um botão que permite o redirecionamento para a tela onde insere o ID da chamada e se ingressa na consulta.



Figura 3.12: Tela de dados da consulta agendada.

3.10 Tela de ID da vídeo chamada

Nesta tela é possível observar um espaço de *input* onde o usuário insere um ID para ingressar ou iniciar uma chamada. Após inserir o ID o médico deve iniciar a chamada apertando o botão “Realizar chamada”. Dessa forma, com a chamada iniciada, o profissional será redirecionado para a próxima tela, onde poderá abrir a câmera do aplicativo e então iniciar de fato a chamada de vídeo.

É importante entender que o ID da chamada de vídeo garante a segurança da chamada, uma vez que só poderá ter acesso a uma determinada chamada o usuário que possui o ID específico desta ligação. Este ID é gerado individualmente para cada consulta marcada de forma que apenas quem tiver o acesso à essa consulta poderá entrar na chamada.

Outro fator que auxilia na segurança da chamada é o fato de a ligação por vídeo possuir uma configuração que permite que apenas duas pessoas possam se conectar simultaneamente à ligação, de maneira que ninguém, além do paciente e do médico, possa ter acesso à chamada enquanto ela acontece.

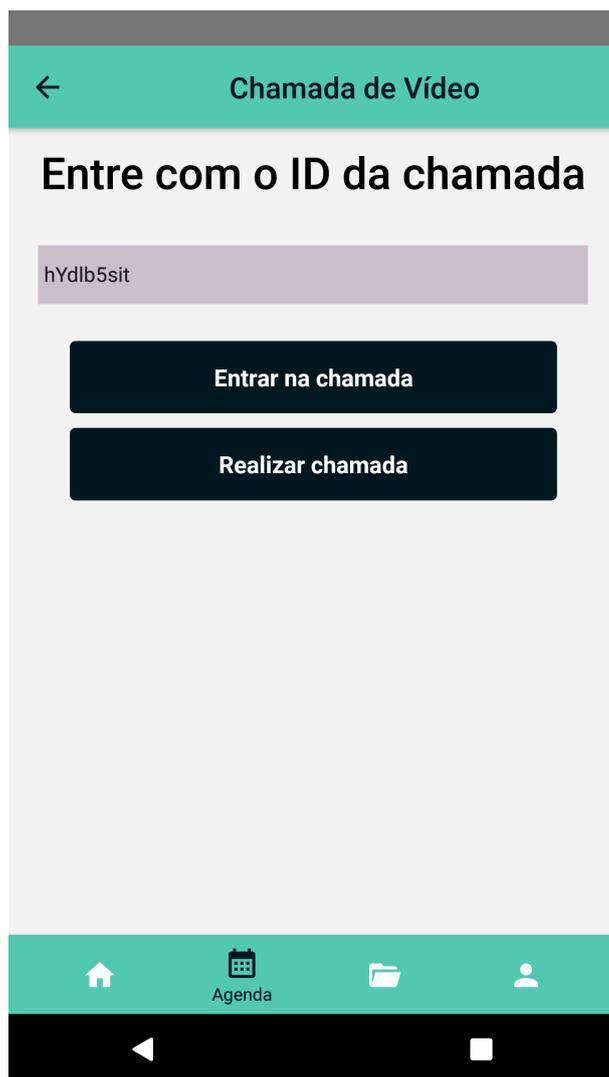


Figura 3.13: Tela para ingressar na chamada utilizando o ID.

3.11 Tela de chamada de vídeo (médico)

Após inserido o ID da chamada de vídeo e ao apertar o botão “Realizar chamada”, na tela anterior; o médico será direcionado para esta tela, onde irá abrir sua câmera e iniciar a chamada, para então aguardar a entrada do seu paciente. Nesta tela ainda existem as algumas opções como: mutar a chamada, virar a câmera, encerrar e iniciar a chamada, como demonstrado na Figura 3.14. Além disso, cabe atentar que é exibido no topo da tela o ID da chamada que está ocorrendo. Dessa forma é possível que o profissional verifique se está na chamada correta.

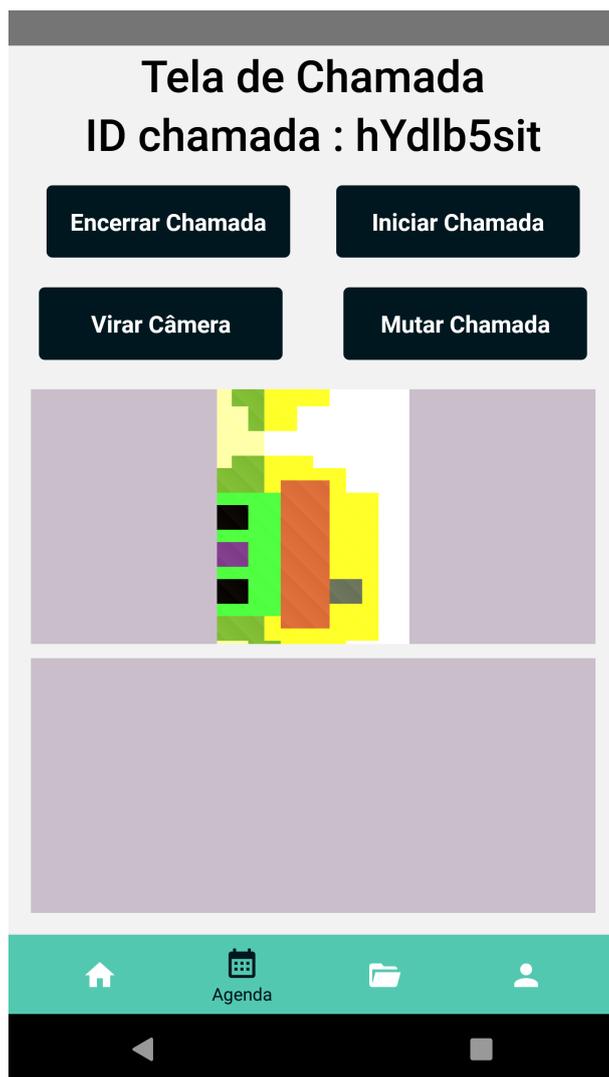


Figura 3.14: Tela de início da chamada de Vídeo.

3.12 Tela de chamada de vídeo (paciente)

No lado do usuário como paciente, existirá a mesma tela de ID da chamada, como na Figura 3.13. Contudo, ao inserir o ID da chamada da consulta, o paciente irá apertar no botão “Entrar na chamada”. Dessa maneira ele também irá para a tela da Figura 3.14 e deverá abrir a câmera. Após os passos anteriores o paciente deve pressionar o botão “Entrar na chamada”, como é demonstrado na Figura 3.15. Com isso, agora tanto médico quanto paciente estarão na chamada e poderão iniciar o atendimento por meio da telemedicina.

Cabe atentar que as imagens representadas nas câmeras da videochamada, na Figura 3.14 e na Figura 3.15, são imagens meramente representativas que o próprio emulador de celular reproduz.

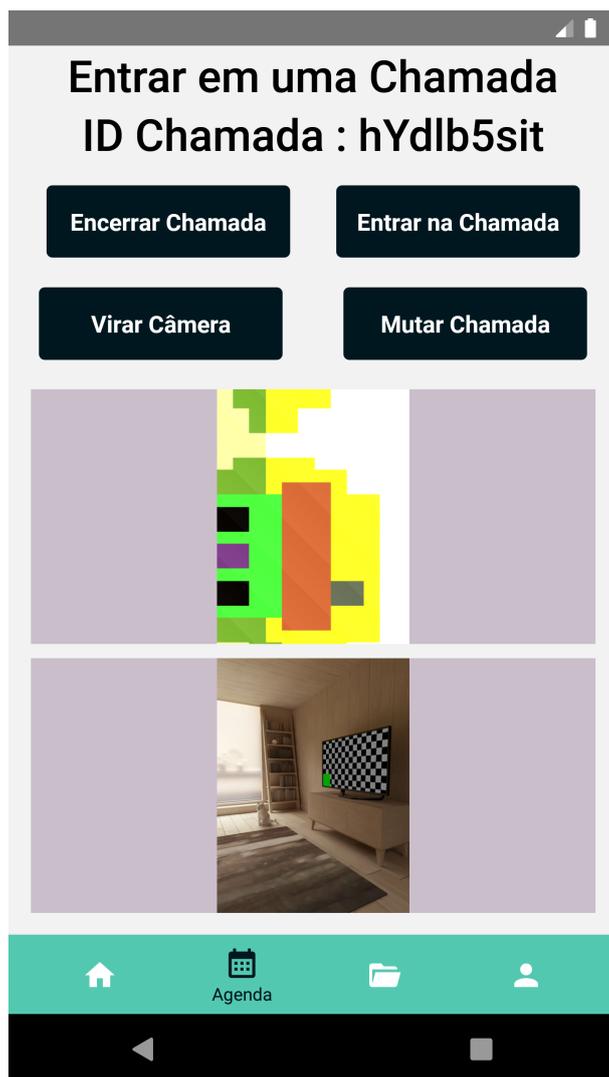


Figura 3.15: Tela atendimento de telemedicina.

3.13 Encerrando a chamada

Ao pressionar o botão “Encerrar a chamada” presente nas telas da Figura 3.14 e da Figura 3.15, o usuário será redirecionado para a tela da Figura 3.13. Desse modo, ele poderá ingressar em alguma outra chamada inserindo um novo ID, ou então, poderá navegar para outras telas do aplicativo.

3.14 Tela de arquivos

Neste aplicativo, é possível também, armazenar e baixar arquivos de interesses da consulta. Com isso, existe a tela de arquivos, onde é possível carregar e fazer o upload de documentos. Nesta tela é possível observar, na Figura 3.16, que existem três botões:

“Escolha o Arquivo”, onde o usuário será redirecionado para a tela do smartpone onde contém os arquivos dele; “Faça o upload do arquivo”, onde o usuário enviará esses documentos para o banco de dados do aplicativo; por fim, “Ir para a página de arquivos”, onde pode-se observar uma lista contendo os arquivos salvos pelo usuário.

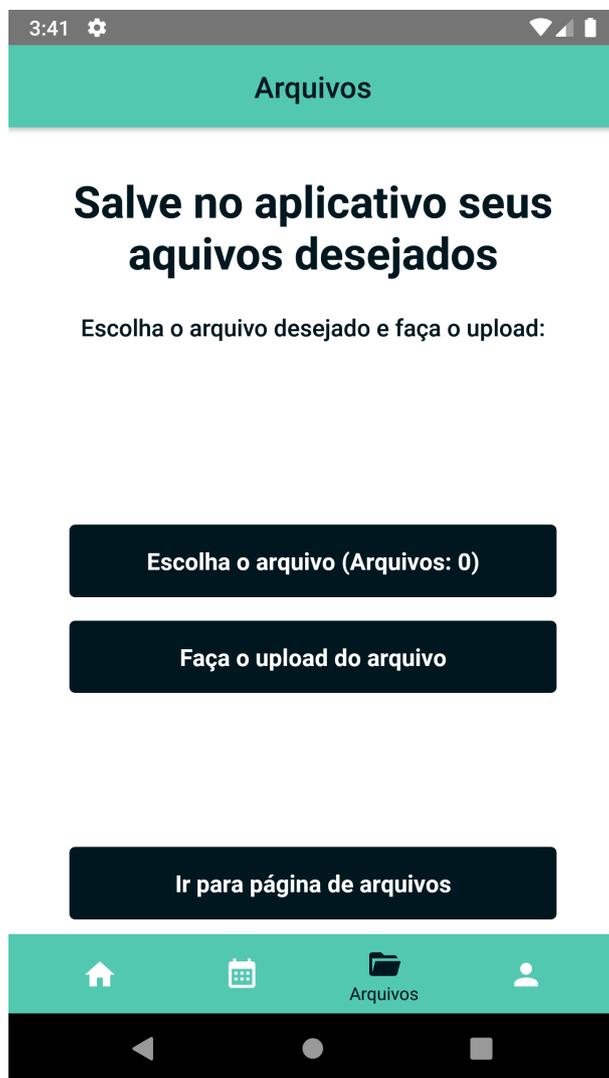


Figura 3.16: Tela de upload de arquivos.

3.15 Escolha do arquivo

Como descrito anteriormente, ao pressionar o botão “Escolha seu arquivo” na tela da Figura 3.16, o usuário será redirecionado para uma tela onde irá constar todos os documentos presentes no celular. Desse modo, ao selecionar o documento desejado, o arquivo estará pronto para ser inserido no banco de dados do aplicativo.

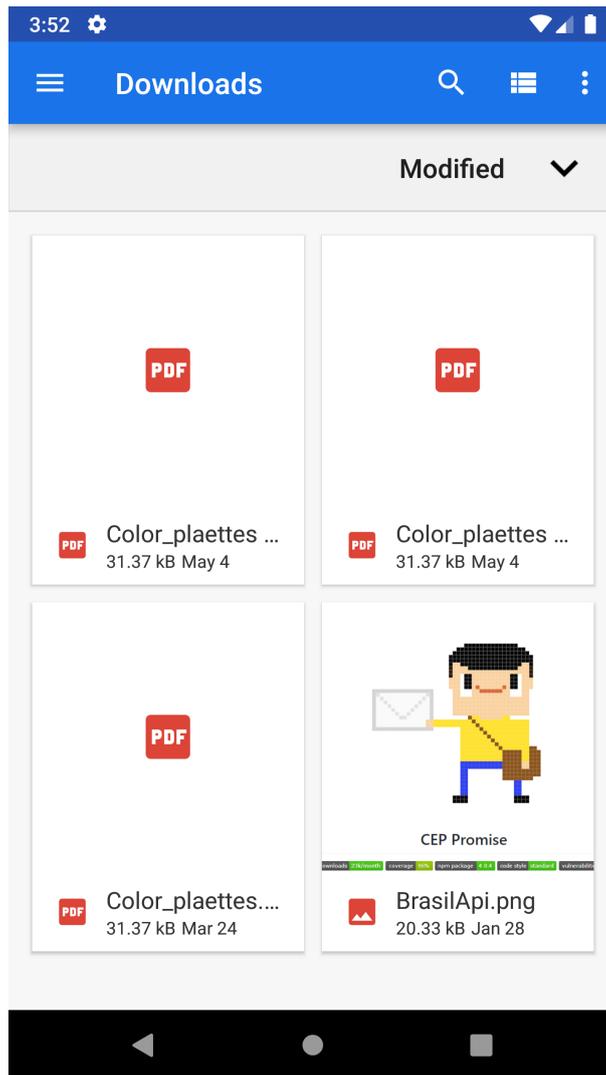


Figura 3.17: Tela de escolha de arquivos.

3.16 Tela de lista de Arquivos

Após realizar o upload do arquivo desejado, o usuário poderá clicar no botão “Ir para página de arquivos” e então poderá visualizar uma tela contendo uma lista com todos os arquivos salvos por ele, como consta na Figura 3.18. Tendo posse dessa lista, o usuário poderá selecionar o arquivo desejado e decidir se deseja fazer, ou não, o *download* deste arquivo.

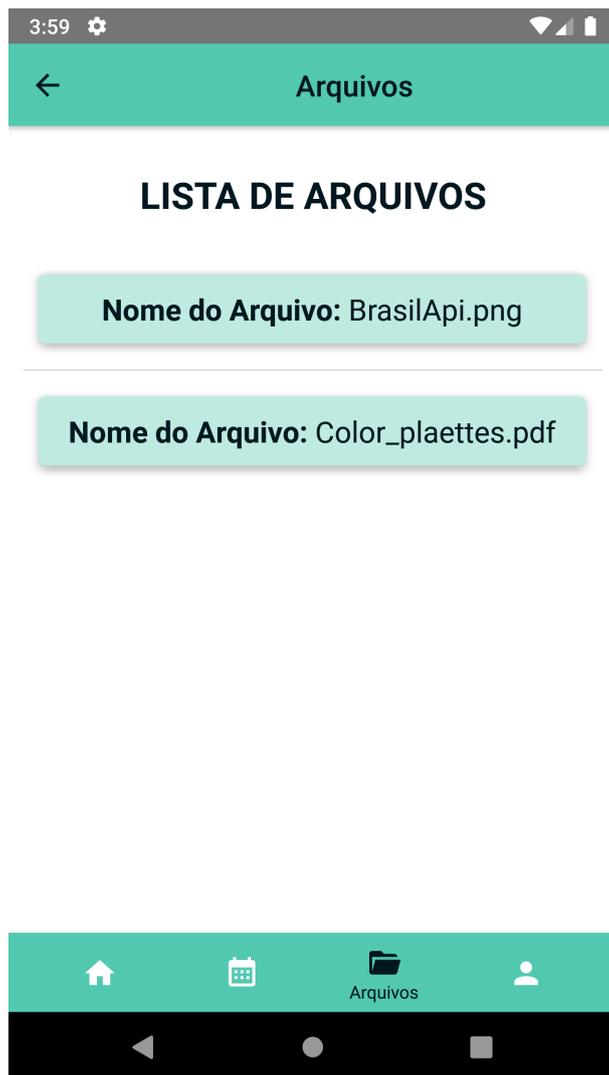


Figura 3.18: Tela com lista de arquivos salvos pelo usuário.

3.17 Tela de perfil

Por fim, a última tela do aplicativo consiste em visualizar o perfil do usuário, onde o mesmo pode consultar dados como: Foto do perfil, nome completo, email, telefone e senha. Além de poder verificar esses dados, o usuário pode realizar o “Logout” do aplicativo.

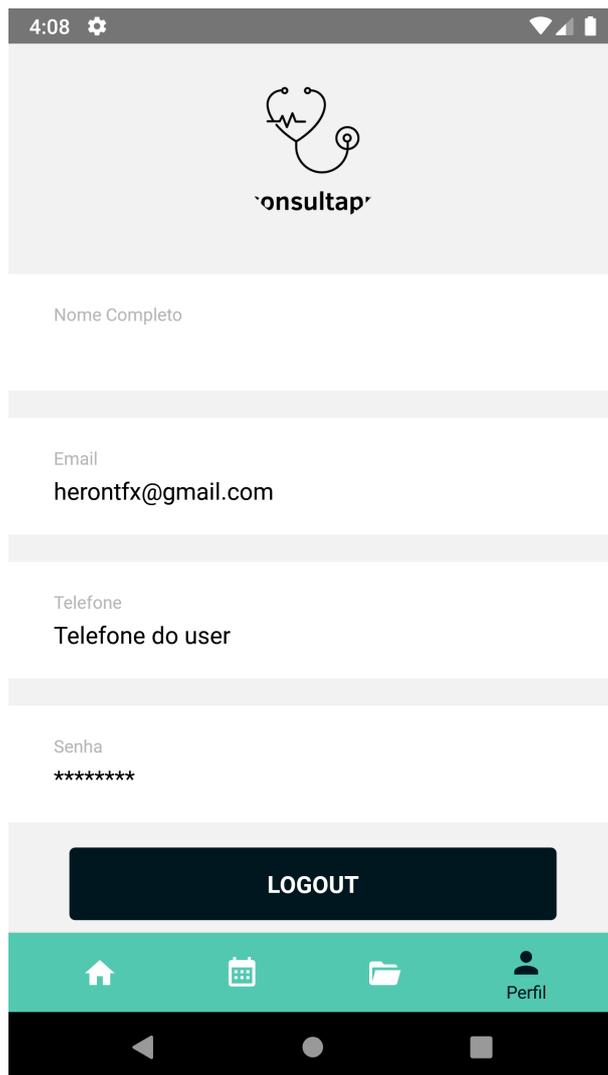


Figura 3.19: Tela de perfil do usuário.

3.18 Firebase

Para que todas as telas acima apresentadas pudessem realizar e exibir as ações e informações nelas existentes, foi necessário realizar a configuração e conexão dos dados do *front-end* com o *back-end* do Firebase. Assim sendo, serão demonstradas as telas do Firebase onde são armazenados os dados dos usuários, assim como os dados do *storage* e do *firestore* (dados em tempo real).

3.18.1 Firebase Authentication

Neste serviço do Firebase, é onde ocorre a autenticação de cada usuário. Alguns dados podem ser visualizados nessa seção como: Identificador, onde consta o email do usuário

cadastrado; Provedor, onde exibe o método de autenticação que foi realizado neste usuário; Data de criação do usuário; Último login e o UID do usuário, que é o ID individual de cada cadastro, onde se pode verificar quem é este usuário.



The screenshot shows the Firebase user management interface. At the top, there is a search bar with the text "Pesquise por endereço de e-mail, número de telefone ou UID do usuário" and a blue button labeled "Adicionar usuário". Below the search bar is a table with the following columns: "Identificador", "Provedores", "Data de criação", "Último login", and "UID do usuário". The table contains six rows of user data. At the bottom right of the table, there is a pagination control showing "Linhas por página: 50" and "1 - 6 of 6".

Identificador	Provedores	Data de criação	Último login	UID do usuário ↑
medico2@gmail.com	✉	29 de abr. d...	11 de mai. d...	3KhLMihyD6PzW419G6wHRxPK2...
medico10@gmail.com	✉	29 de abr. d...	29 de abr. d...	6NebfZ30OXfYVYIFNA3w1RUPrd2
herontf@gmail.com	✉	12 de abr. d...	4 de mai. de...	CEdWZtKcTngHcWDeMzTSs8ZAV...
herontfx@gmail.com	✉	12 de abr. d...	11 de mai. d...	awhFtveB3dQHo4UOILkOMK8SRu...
teste@gmail.com	✉	11 de mai. d...	11 de mai. d...	jMUzyck8dMbf7olZhExy5RjG8RC2
medico1@gmail.com	✉	29 de abr. d...	29 de abr. d...	ptgnTZcQ5JQfOeotY5qQcDN2pzi2

Figura 3.20: Dados de autenticação dos usuários no Firebase.

3.18.2 Firebase Firestore

No Firestore é onde alguns dados que precisam de verificação em tempo real são armazenados. Logo, existe uma constante verificação e atualização dessas informações ali armazenadas. Neste projeto foram criadas três coleções de dados a serem armazenados e verificados, são eles:

Coleções:

- rooms: Coleção responsável por criar salas para que as chamadas de vídeo ocorram de maneira segura, vide Figura 3.21;
- categories: Coleção responsável por criar e armazenar as especialidades na lista da tela “home” do aplicativo. Neste item, os documentos da coleção constam apenas um campo: title, vide Figura 3.22;
- appointments: Coleção responsável por criar e armazenar as consultas marcadas pelos clientes. Nessa coleção, os documentos possuem campos como: date, doctor, specialty e UID. É importante observar que este ultimo campo foi inserido com a necessidade de identificar a qual usuário esta consulta pertence, individualizando, assim, cada documento da coleção; vide Figura 3.23;

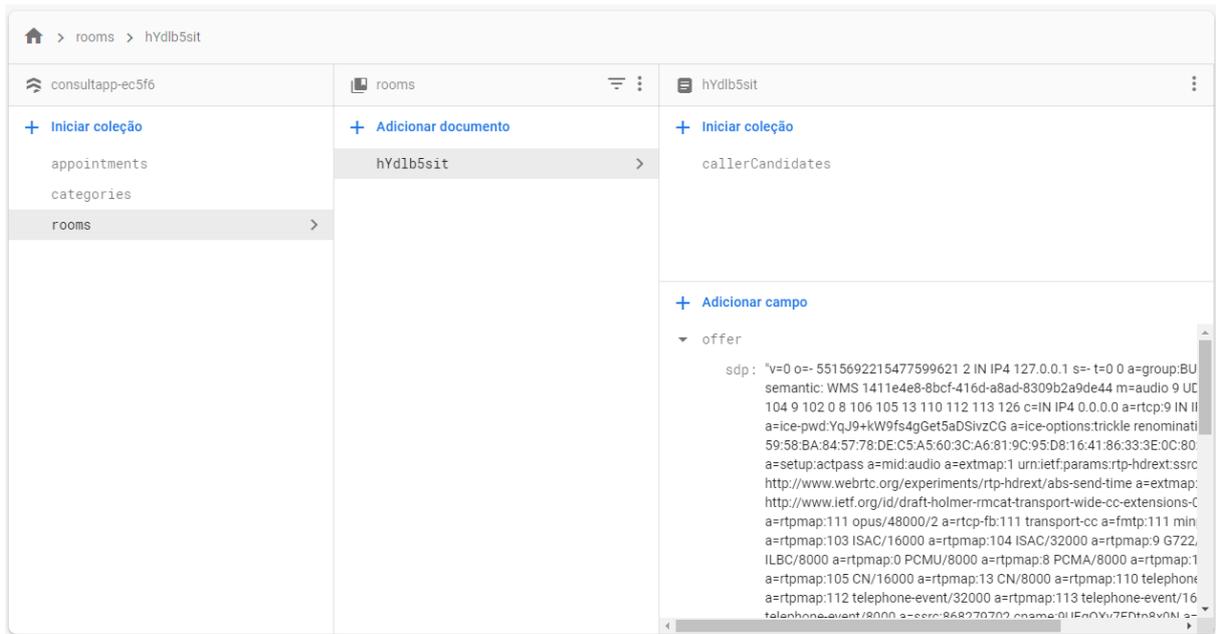


Figura 3.21: Coleção de salas para as vídeo chamadas.

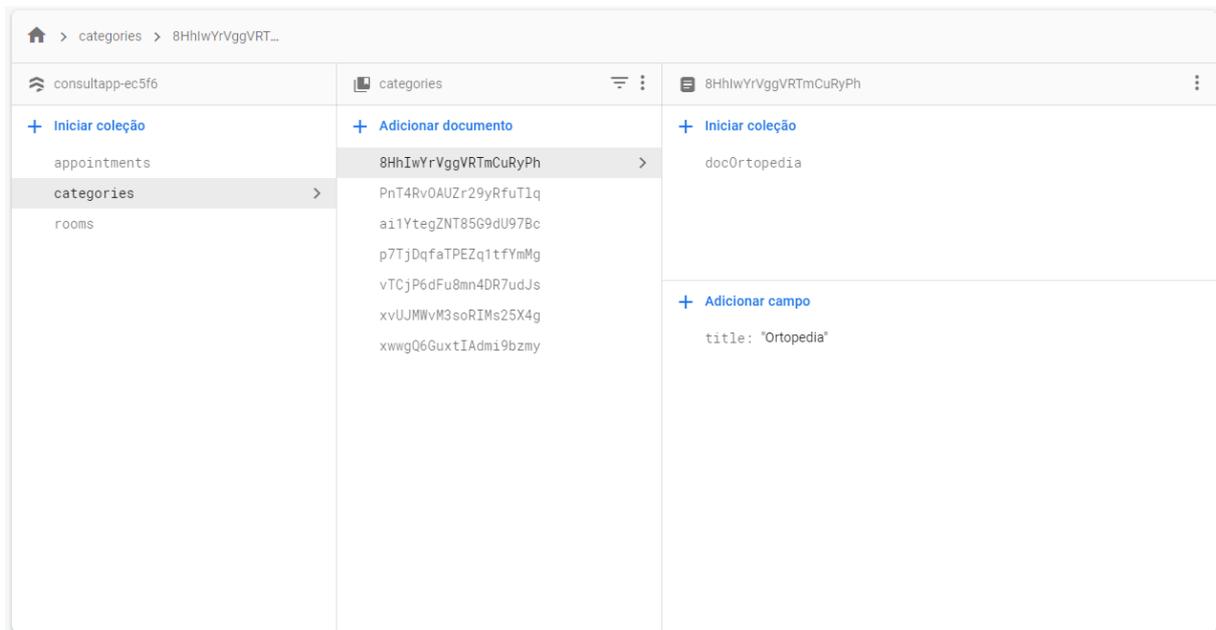


Figura 3.22: Coleção de especialidades médicas.

3.18.3 Firebase storage

Finalmente, o serviço de storage do Firebase permite que os usuários armazenem dados estáticos de uma maneira segura, restringindo apenas ao usuário a opção de visualizar o documento que ele guardou. Outra função que pode ser encontrada nesse sistema é a opção de o usuário realizar o download do arquivo guardado por ele, o que permite ao

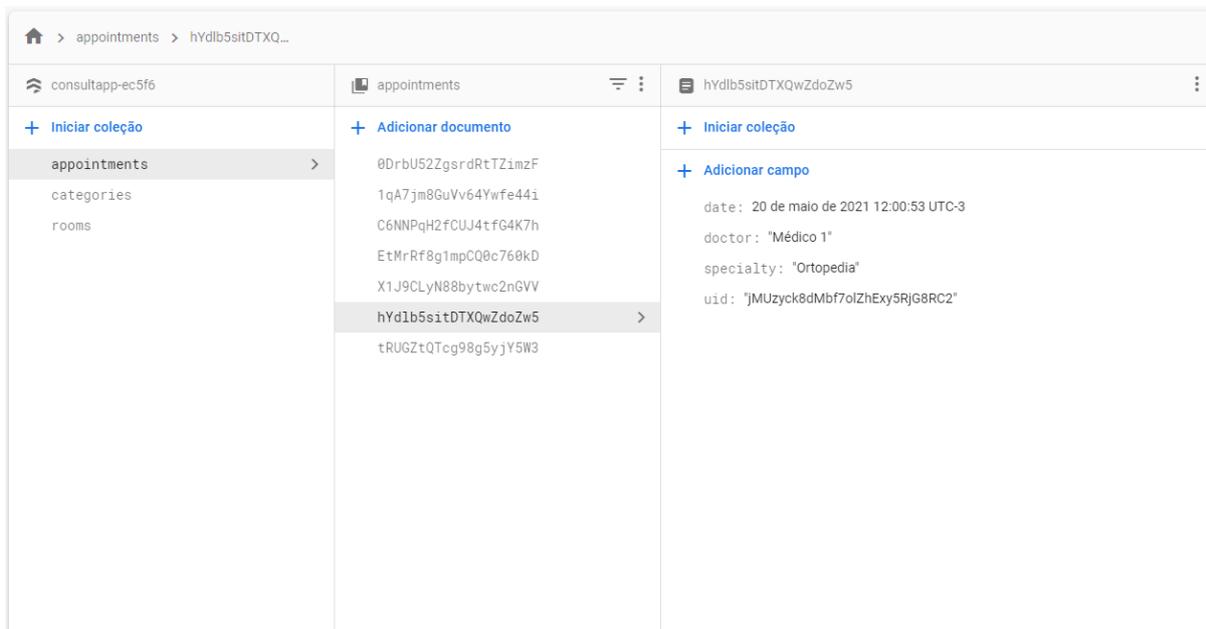


Figura 3.23: Coleção de consultas.

utilizador do aplicativo a possibilidade de baixar o documento em qualquer dispositivo ao qual esteja conectado. Na Figura 3.24 podemos observar a estrutura de armazenamento dos arquivos no Firestore.



Figura 3.24: Banco de dados do Firebase.

3.19 Repositório do projeto

O projeto exposto acima pode ser visualizado e clonado no repositório do GitHub representado no seguinte link:

<https://github.com/HeronFonseca/Consultapp>

Capítulo 4

Conclusões

Com o exposto, percebe-se o quanto a telemedicina pode ajudar tanto os profissionais da área da saúde quanto os pacientes que necessitam de interagir com eles. A interação entre os dois personagens da utilização deste projeto se faz ainda mais necessária levando em conta o contexto da sociedade atual que está passando por uma pandemia de SARS-CoV-2 (coronavírus). Com isso, a tecnologia pode inovar e ajudar a minimizar as barreiras existentes na aproximação entre médicos e pacientes.

O aplicativo desenvolvido visa facilitar a interação acima descrita por meio da telemedicina e de ferramentas de desenvolvimento de *software* como o *React Native* e o *Firebase*. Essas ferramentas, por sua vez, têm crescido, cada vez mais, e auxiliado desenvolvedores do mundo inteiro a criar e implementar ferramentas que ajudem no dia-a-dia da humanidade.

Graças à facilitação das linguagens e *frameworks* utilizados na elaboração do projeto, pôde-se implementar um projeto que promove a interação segura entre profissionais da saúde e pacientes, de maneira que eles possam interagir, com as vídeo chamadas; compartilhar conteúdos, realizando upload e download de documentos para compartilhar quando necessário; e realizar a marcação de consultas, removendo barreiras que dificultam este tipo de serviço e promovendo a segurança dos envolvidos por não precisarem sair de suas casas.

O aplicativo desenvolvido neste projeto é um MPV (*Minimum Viable Product*)[16], portanto, ainda há muito o que agregar neste aplicativo. Para implementações futuras é indicado realizar melhorias na estrutura UI e UX de maneira a melhorar aspectos de como as cores vão harmonizar com o aplicativo e onde cada componente deve se localizar nas telas, a fim de melhorar como um usuário vê cada função de interação dentro do *software*. Além do mais, é necessário, também, melhorar a experiência do usuário com o aplicativo realizando pesquisas e estudos de maneira a tornar a experiência de uso do aplicativo o mais intuitivo e simples possível.

Algumas funções, como o de recuperação de senha e gravação da chamada de vídeo, ainda não foram implementadas nessa primeira versão do aplicativo, por conseguinte é importante desenvolver e aprimorar essas e outras funções, que poderão agregar ainda mais valor ao *software*.

Contudo, o aplicativo desenvolvido, também, apresenta algumas melhorias em relação à outros já existentes, como o fato de possuir uma usabilidade bem simples e fluida, de maneira que é necessário pouca interação com o sistema para poder entendê-lo. Além disso, o *software* foca nas suas três funções principais: agendamento de consulta, chamada de vídeo e *upload/download* de arquivos; de forma que não existam outras informações que possam poluir as atividades principais, como informativos ou listas de possíveis exames, por exemplo.

Referências

- [1] Brasil, 2020. [https://www.planalto.gov.br/ccivil_03/_Ato2019-2022/2020/Lei/L13989.htm#:~:text=Disp%C3%B5e%20sobre%20o%20uso%20da,SARS%2DCoV%2D2\).&text=0%20PRESIDENTE%20DA%20REP%C3%9ABLICA%20Fa%C3%A7o,SARS%2DCoV%2D2\).](https://www.planalto.gov.br/ccivil_03/_Ato2019-2022/2020/Lei/L13989.htm#:~:text=Disp%C3%B5e%20sobre%20o%20uso%20da,SARS%2DCoV%2D2).&text=0%20PRESIDENTE%20DA%20REP%C3%9ABLICA%20Fa%C3%A7o,SARS%2DCoV%2D2).), acesso em 20/04/2021. 1
- [2] Cunha Pereira França Magalhães, Carolina da: *Guia prático para atendimentos por telemedicina em tempos de covid-19*, 2020. <https://onlinedoctor.com.br/cartilha-telemedicina.pdf>, acesso em 20/04/2021. 1
- [3] Erno, Harzheim, Katz Natan, Ferri Cleusa, Gomes Fernandes Jefferson e Ingrid Barbosa: *Guia de avaliação, implantação e monitoramento de programas e serviços em telemedicina e telessaúde*. https://rebrats.saude.gov.br/images/MenuPrincipal/Guia_Avaliacao_telessaude_telemedicina.pdf, acesso em 20/04/2021. 1
- [4] *Doutor hoje*. <https://drhoje.com/>, acesso em 10/05/2021. 2
- [5] *Doutor consulta*. <https://www.drconsulta.com/>, acesso em 10/05/2021. 2
- [6] *Boa consulta*. <https://www.boaconsulta.com/>, acesso em 10/05/2021. 3
- [7] Mozilla: *Javascript tutorials*. <https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>, acesso em 20/04/2021. 4
- [8] TIOBE: *Tiobe index for may 2021*, 2021. <https://www.tiobe.com/tiobe-index/>, acesso em 28/04/2021. 4
- [9] Facebook: *Facebook/react-native*. <https://github.com/facebook/react-native>, acesso em 28/04/2021. 5
- [10] Franchet, Marvin: *Understanding the react native bridge concept*, 2017. <https://hackernoon.com/understanding-react-native-bridge-concept-e9526066ddb8>, acesso em 04/05/2021. 5
- [11] Priya: *15+ react native component libraries you should know in 2021*, 2020. <https://www.codeinwp.com/blog/react-native-component-libraries/>, acesso em 04/05/2021. 6
- [12] Facebook: *Redux*. <https://redux.js.org/introduction/installation>, acesso em 04/05/2021. 6

- [13] Rocketseat: *Unform*. <https://unform.dev/>, acesso em 04/05/2021. 6
- [14] Gasperin, Carlos Alberto: *Firestore: o que é e como funciona*. <https://micreiros.com/firebase-o-que-e-e-como-funciona/>, acesso em 05/05/2021. 6
- [15] Firebase: *Firestore documentation*. <https://firebase.google.com/docs/database/web/structure-data?hl=pt-br>, acesso em 05/05/2021. 7
- [16] Lenarduzzi, Valentina e Davide Taibi: *MVP Explained: A Systematic Mapping Study on the Definitions of Minimal Viable Product*. Addison-Wesley Reading, MA, USA, 2016. 32