



**UNIVERSIDADE ESTADUAL DA PARAÍBA  
CAMPUS I  
CENTRO DE CIÊNCIAS E TECNOLOGIA  
DEPARTAMENTO DE COMPUTAÇÃO  
CURSO DE CIÊNCIA DA COMPUTAÇÃO**

**KAIO GIOVANNI BEZERRA DE LUCENA**

**SISTEMA DE INFORMAÇÃO PARA O GERENCIAMENTO E INTERAÇÃO  
PROFISSIONAL- PACIENTE NO ÂMBITO DAS UNIDADES PRIVADAS DE  
SAÚDE NO BRASIL**

**CAMPINA GRANDE  
2021**

**KAIO GIOVANNI BEZERRA DE LUCENA**

**Sistema de Informação para o Gerenciamento e Interação  
Profissional-Paciente no Âmbito das Unidades Privadas de Saúde  
no Brasil**

Trabalho de Conclusão de Curso de  
Graduação em Ciência da Computação  
da Universidade Estadual da Paraíba,  
como requisito à obtenção do título de  
bacharel em Ciência da Computação.

**Área de concentração:** Engenharia de  
Software.

**Orientador:** Prof. Dr. Vladimir Costa de Alencar.

**CAMPINA GRANDE  
2021**

É expressamente proibido a comercialização deste documento, tanto na forma impressa como eletrônica. Sua reprodução total ou parcial é permitida exclusivamente para fins acadêmicos e científicos, desde que na reprodução figure a identificação do autor, título, instituição e ano do trabalho.

L935s Lucena, Kaio Giovanni Bezerra de.  
Sistema de informação para o gerenciamento e interação profissional-paciente no âmbito das Unidades Privadas de Saúde no Brasil [manuscrito] / Kaio Giovanni Bezerra de Lucena. - 2021.  
59 p.  
Digitado.  
Trabalho de Conclusão de Curso (Graduação em Computação) - Universidade Estadual da Paraíba, Centro de Ciências e Tecnologia, 2021.  
"Orientação : Prof. Dr. Vladimir Costa de Alencar ,  
Coordenação do Curso de Computação - CCT."  
1. Saúde privada. 2. Sistema de informação. 3. Engenharia de software. I. Título  
21. ed. CDD 005.1

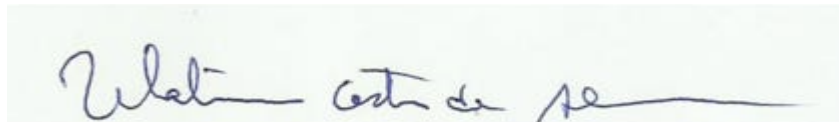
**KAIO GIOVANNI BEZERRA DE LUCENA**

**Sistema de Informação para o Gerenciamento e Interação  
Profissional-Paciente no Âmbito das Unidades Privadas de Saúde  
no Brasil**

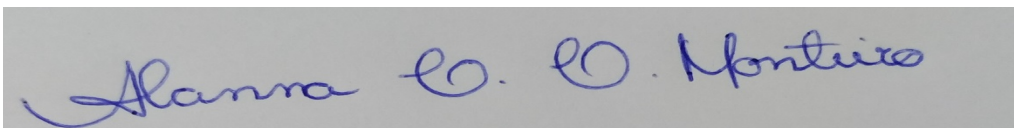
Trabalho de Conclusão de Curso de Graduação  
em Ciência da Computação da Universidade  
Estadual da Paraíba, como requisito à  
obtenção do título de Bacharel em Ciência da  
Computação.

Aprovada em 02 de Junho de 2021.

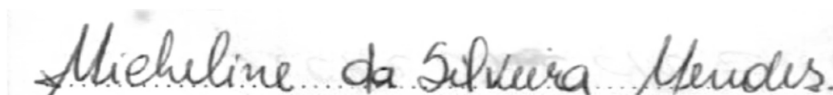
**BANCA EXAMINADORA**



Prof. Dr. Vladimir Costa de Alencar (DC - UEPB)  
Orientador(a)



Profa. Me. Alanna Camylla Coelho Monteiro (UEPB)  
Examinador(a)



Me. Micheline da Silveira Mendes (IAM/FIOCRUZ-PE e UEPB)  
Examinador(a)

À Deus, meus familiares e amigos, pela  
dedicação, apoio, companheirismo e  
amizade, DEDICO.

## **AGRADECIMENTOS**

À Deus por ter me dado saúde, força, perseverança e sabedoria para concluir este trabalho.

À Universidade Estadual da Paraíba por me conceder a oportunidade e privilégio de cursar e concluir o curso de Bacharelado em Ciência da Computação.

Ao orientador e Prof. Dr. Vladimir Costa de Alencar por ter me proporcionado oportunidades únicas na caminhada acadêmica, como também pela sua confiança, companheirismo e disponibilização das ferramentas necessárias para realização deste trabalho.

Por fim, mas não menos importante, agradeço a toda minha família, minha amiga e namorada Deborah Lima Vieira por todo apoio e ajuda e ao meu amigo Joanes Miranda por todo companheirismo e auxílio.

## RESUMO

O atendimento primário de saúde é a principal forma de tratamento da população, porém esse setor ainda enfrenta vários obstáculos quanto a qualidade de acesso aos postos de atendimento. Diante disso, levando-se em consideração o crescente investimento nesse setor, o uso da tecnologia abrange o crescente desenvolvimento de sistemas de informação que venham a sanar os principais problemas enfrentados pela população nesse meio. Advindo disso, o presente trabalho tem como objetivo apresentar o *Health Technology*, um sistema de informação de gerenciamento de consultas das unidades de saúde, que busca facilitar e aproximar o profissional de saúde ao paciente, utilizando de funcionalidades como o chat entre profissional e paciente, cadastro de usuários, agendamento de consultas, cadastro de associados à empresa, entre outros. O sistema é composto por um aplicativo destinado ao uso dos pacientes, um aplicativo destinado ao uso dos profissionais do sistema, um site administrativo para gerenciar as informações referentes à empresa e uma API REST para a persistência e gerenciamento dos dados. Os aplicativos foram desenvolvidos para a plataforma Android, utilizando a linguagem de programação Java, ReactJS para a construção do site do administrador, e o NodeJS para criar a API REST, o Git para o controle de versão e para disponibilizar o site e a API online foi utilizado o Heroku, uma plataforma de hospedagem de aplicações com um ambiente altamente escalável e com suporte a várias tecnologias. O sistema permite, de maneira geral, cadastrar pacientes, profissionais, associados à empresa, marcação de consultas, espaço com chat entre profissionais e pacientes promovendo assim um relacionamento maior entre os envolvidos.

**Palavras-Chave:** Saúde privada. Sistema de informação. Paciente.

## ABSTRACT

Primary health care is the main form of treatment for the population, but this sector still faces several obstacles regarding the quality of access to service points. In Brazil, this service is best performed in private health units because the plans and agreements always seek to deliver the maximum quality in their services to their customers, with better service, infrastructure, attention and care in front of their professionals and various types of resources that involve prevention, treatment and health promotion to their patients. In view of this, taking into account the growing investment in this sector, the use of technology encompasses the growing development of information systems that will solve the main problems faced by the population in this environment. As a result, the present work aims to present Health Technology, an information management system for consultations in private health units, which seeks to facilitate and bring health professionals closer to the patient, using features such as patient registration, scheduling consultation, chat between professional and patient, registration of associates to the company, among others. The system consists of an application for the use of patients, an application for the use of professionals in the private system, an administrative website to manage information related to the company and a REST API for persistence and data management. The applications were developed for the Android platform, using the Java programming language, ReactJS to build the administrator's website, and NodeJS to create the REST API, Git for version control and to make the website and API available online. Heroku was used, an application hosting platform with a highly scalable environment and with support for various technologies. The system allows, in general, to register patients, professionals, associated with the company, book appointments, a chat space between professionals and patients, thus promoting a greater relationship between those involved.

**Keywords:** Private health. Information system. Patient.



## LISTA DE ILUSTRAÇÕES

Figura 1 –	Arquitetura do sistema operacional Android .....	20
Figura 2 –	Ordem de operações do event-loop .....	23
Figura 3 –	O protocolo Web Socket .....	27
Figura 4 –	Diagrama de casos de uso do administrador do sistema .....	29
Figura 5 –	Diagrama de casos de uso do profissional de saúde .....	29
Figura 6 –	Diagrama de casos de uso do paciente .....	30
Figura 7 –	Requisição HTTP com Express .....	31
Figura 8 –	Enviando mensagens com o Socket-io .....	32
Figura 9 –	Exemplo de código JSX .....	33
Figura 10 –	Código de rotas da API .....	34
Figura 11 –	Verificação de usuário autenticado .....	35
Figura 12 –	Enviando dados através do web site .....	36
Figura 13 –	Tela de acesso do administrador .....	37
Figura 14 –	Tela de cadastro do administrador.....	38
Figura 15	Tela de informações dos usuários .....	39
Figura 16 –	Tela de cadastro de profissionais de saúde .....	40
Figura 17 –	Tela de cadastro de associados .....	41
Figura 18 –	Tela de cadastro de profissões .....	42
Figura 19 –	Tela de login dos usuários .....	42
Figura 20 –	Tela de cadastro do paciente .....	43
Figura 21 –	Tela do menu de navegação .....	44
Figura 22 –	Tela de conversas .....	45
Figura 23 –	Tela de consultas .....	46
Figura 24 –	Tela de associados .....	47
Figura 25 –	Tela de estatísticas do profissional .....	48
Figura 26 –	Tela de configurações .....	49

## LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
CSS	Cascade Style Sheet
HTML	Hyper Text Marcation Language
IDE	Integrated Development Environment
I/O	Entrada e saída
JSX	Javascript XML
SDK	Software Development Kit
SPA	Single Page Application

## SUMÁRIO

1	INTRODUÇÃO .....	11
2	JUSTIFICATIVA .....	16
3	OBJETIVOS .....	17
3.1	Objetivos gerais .....	17
3.2	Objetivos específicos .....	17
4	FUNDAMENTAÇÃO TEÓRICA.....	18
4.1	O sistema operacional Android .....	18
4.1.1	Licença .....	19
4.1.2	Arquitetura .....	20
4.2	O NodeJS .....	22
4.2.1	O event-loop .....	23
4.1	API REST .....	24
4.4	O ReactJS .....	25
4.4.1	SPA .....	25
4.4.2	JSX .....	25
4.5	WEB SOCKETS .....	26
4.5.1	Diferenças entre o protocolo HTTP e o Web Socket .....	26
4.6	O Heroku .....	27
5	METODOLOGIA .....	28
5.1	Estudo do diagrama de casos de uso .....	28
5.2	Estudo do ambiente de desenvolvimento Android .....	30
5.3	Estudo da criação de uma API REST .....	31
5.3.1	O framework Express .....	31
5.3.2	O framework Socket-IO .....	32
5.4	Estudo da criação de um web site .....	32
6	RESULTADOS .....	34
6.1	Ferramentas utilizadas e trechos de código .....	34
6.1.1	Tratando requisições HTTP .....	34
6.1.2	Verificação de usuário autenticado .....	35
6.1.3	Envio de dados através do web site .....	35
6.2	Funcionalidades do sistema .....	36
7	CONCLUSÃO .....	49

REFERÊNCIAS .....	51
Apêndice A .....	57
Apêndice B .....	59

## 1 INTRODUÇÃO

A saúde no Brasil é uma das questões mais discutidas e complexas em um país onde já foi palco de inúmeras ocorrências patológicas endêmicas e por todo o tipo de demandas social, política e econômica. Devido a isso foi-se registrado em nossa Constituição que “A saúde é direito de todos e dever do Estado, garantido mediante políticas sociais e econômicas que visem à redução do risco de doença e de outros agravos e assegurem o acesso universal e igualitário às ações e serviços para sua promoção, proteção e recuperação da saúde” (BRASIL, 1988). Porém, essas como tantas outras obrigações do Estado brasileiro diante dos seus cidadãos, tem sido historicamente difícil de ser atendida. No caso da saúde, ainda há de enfrentar adversidades, seja sob o aspecto de avaliação da capacidade do Estado de financiar sua obrigação constitucional, ou pela busca de soluções alternativas na administração do sistema suplementar (FILHO, 1999).

Historicamente, é de veras o descaso na saúde pública fitando desde o Império português, até passagem pela Independência do nosso País, com a construção da nossa República. Até hoje, onde, mesmo após avanços com ações implementadas para o desenvolvimento do mesmo, há negligência na forma de atuação das mesmas no setor público, podendo-se avaliar desde o ano de 1828 onde a fragilidade das medidas sanitárias levava a população a lutar por conta própria contra as doenças e a morte e além disso, os doentes tinham muito medo de serem internados nos raros hospitais públicos e nas Santas Casas, o que leva à comparação do estado atual, onde mesmo com a implantação e criação de novos recursos para o desenvolvimento do Sistema Único de Saúde (SUS), sua atuação ainda encontra-se não consolidada (FILHO, 2010 ; PAIM, 2013).

Da mesma forma que há notas da existência da performance da saúde pública no Brasil, também se tem em conta as medidas tomadas na formação do setor privado de saúde, onde, desde seus primórdios na década de 60, surgiram as primeiras empresas de medicina de grupo direcionadas, principalmente, aos trabalhadores do ABC paulista. Pois, devido as deficiências no setor público, os donos das indústrias multinacionais, que acabavam de se instalar no país, preocuparam-se em buscar outros meios para propiciar atendimento médico de qualidade a seus empregados, para que assim tivesse uma efetiva resposta financeira. A grande procura levou a formação de empresas de medicina de grupo,

com diferentes planos de saúde. No ano de 1997, cerca de 17 milhões de brasileiros faziam parte desses planos de saúde (FILHO, 1999).

Sendo criado os planos de saúde, se obteve uma vasta categorização de diferentes formas de associação e vinculação, incluindo assim diversas classes sociais, dando acesso à população a experimentar um atendimento à saúde de melhor qualidade e cuidado, podendo evidenciar-se múltiplas modalidades como a das cooperativas médicas, objetivadas em prestar assistência aos beneficiários por meio de contratos coletivos, familiares e individuais, ou seja, os lucros das operações do sócio majoritário são divididos entre os cooperativos (médicos e outros profissionais da área da saúde), de acordo com a sua contribuição e esforço comum. Além dessa opção, houve também a modalidade da autogestão, onde os empregadores que gerenciavam planos próprios de saúde para seus funcionários diante a contratação ou credenciamento de médicos e serviços, e de convênios com hospitais, ou seja, a própria empresa que estabelece e define o formato do plano, o credenciamento dos médicos e dos hospitais e estabelece as carências e coberturas necessárias (FILHO,1999).

Por fim, temos o modelo do seguro de saúde, sendo ambíguo ao conceito dos planos de saúde. Esses modelos se conceituam em empresas seguradoras que atuam na área da saúde suplementar, da mesma forma que em seus outros segmentos. Tendo seu surgimento a partir de planos de custeios, que garantem aos segurados a livre escolha de médicos e hospitais por meio do reembolso de despesas. Esse modelo de seguro evoluiu e atualmente, além do sistema de reembolso, também trabalham com hospitais, médicos e laboratórios referenciados, sem prejuízo da livre escolha, ou seja, o segurado é visto sem a necessidade de dispêndio prévio (FILHO, 1999).

A partir do sistema privado de saúde, também chamado de sistema de saúde suplementar, deu-se a regulação da participação da iniciativa privada no sistema de saúde brasileiro, que é formada pelas Leis de nº 9.656/98 e nº 9.961/2000 e a Medida Provisória MP nº 2.177-44. A Lei 9.961 de 28 de janeiro de 2000, deu início a Agência Nacional de Saúde Suplementar (ANS), que tem como objetivo, promover a defesa do interesse público na assistência suplementar à saúde, respondendo pela regulação econômica, financeira e assistencial do conjunto de empresas do sistema privado de saúde suplementar. Pelo lado econômico-financeiro, a ANS

monitora as operadoras quanto à obrigação de ter garantias financeiras para o cumprimento de seus compromissos com seus consumidores (ANS, 2000).

Com a Lei de Planos e Seguros de Saúde (9.656/98), os planos de saúde passaram a ser segmentados como: Planos antigos, cujos contratos foram celebrados antes da vigência da Lei nº 9.656/98, valendo, portanto, o que está estabelecido em contrato. A lei define que esses planos devem ser cadastrados na ANS para informar as condições gerais de operação estabelecidas em contrato; e Planos novos, cujos contratos foram celebrados na vigência da Lei nº 9.656/98, ou seja, a partir de 1º de janeiro de 1999. Estes planos de saúde estão totalmente sujeitos à nova legislação e devem ter registro na ANS para que possam ser comercializados (ANS, 2009).

Após a concretização efetiva e positiva da atuação do sistema privado de saúde, trazendo o benefício da escolha, do conforto, da segurança e da maleabilidade diante dificuldades, devido sua ampla gama de opções e caminhos que o segurado pode escolher seguir de acordo com sua compatibilidade, fazendo com que, hoje, o setor de planos e seguros de saúde seja um dos maiores sistemas privados do mundo (ANS, 2019).

Tendo em vista que a meta desses planos e seguros é oferecer o melhor serviço de saúde, a cada ano a busca pelo desenvolvimento nesse setor é ávido, pois a somatória de serviços quali-quantitativos implantados faz com que os pacientes segurados possam ter maior contato com os profissionais de saúde vinculados, recebendo assim uma maior atenção dos mesmos, compreendendo a uma resposta de melhora mais eficaz do paciente. Dessa forma, são obedecidas as Diretrizes Nacionais/Brasileiras que tratam de documentos norteadores das melhores práticas a serem seguidas por profissionais de saúde e gestores, sejam eles do setor público ou privado da saúde (MINISTÉRIO DA SAUDE, 2020).

Hoje em dia, ao se falar em desenvolvimento, liga-se automaticamente ao plano tecnológico e inovações tecnológicas que estão em alta nos meios de comunicação e nas agendas de governos, empresas, agências de fomento a pesquisas e diversas organizações sociais, com forte influência no setor saúde. Com isso, é notório a atuação da inovação tecnológica, seja em termos de disponibilidade de equipamentos ou em novas técnicas assistenciais, sobre diferentes campos ou especialidades do setor saúde, trazendo para esse setor avanços que se expressaram com a introdução da informática e o aparecimento de aparelhos

modernos e sofisticados, levando ao benefício e a rapidez no diagnóstico e tratamento das doenças (LORENZETTI et al., 2012).

A preocupação pela modernidade vai além de um desafio político, econômico ou administrativo, cabe também ao setor tecnológico arranjar, criar e recriar ações de modo a produzir, de modo universal e equitativo, não apenas tratamento, prevenção ou recuperação da saúde, mas, por meio de todos e cada um destes recursos, produzir cuidado. Com isso, é dito que, as tecnologias leves (tecnologias de relações) é a dimensão em que operam as interações humanas no trabalho em ato na saúde, onde devem ser permeáveis à mudança, ao novo, à reconstrução (AYRES, 2009).

A partir da tecnologia, a forma de interação social modificou, tornando mais fácil o acesso a comunicação, sendo assim, mais hábil a relação entre profissional de saúde e paciente. Neste contexto, os trabalhadores da saúde, devem acolher os usuários de modo a conhecê-los pelo nome, procurando saber os motivos de sua vinda à unidade, ou seja, utilizar de recursos para que o mesmo relate com confiança suas reais necessidades de saúde (COELHO et al., 2009).

Pode-se ter em mente que o atendimento e acolhimento deveria ser executado como uma técnica de conversa passível de ser operada por qualquer profissional, em qualquer momento de atendimento, isto é, em qualquer dos encontros, que são, enfim, os "nós" dessa imensa rede de conversações que são os serviços (TEIXEIRA, 2005). Pois, no setor de recepção do paciente, é onde o profissional ouve e discute com o usuário, em busca de identificar suas necessidades e saber o melhor modo de resolvê-las, para que, dessa forma, sejam resolvidas de imediato as dificuldades enfrentadas pelos pacientes como: facilitar sua consulta, receber atendimento adequado, ter o direito a cancelar, antecipar ou adiar suas consultas de forma mais rápida, entre outros recursos que podem ser facilmente proporcionados pelo auxílio de ferramentas tecnológicas (COELHO et al., 2009).

O meio de comunicação mais utilizado para agendamento de consultas nos estabelecimentos que oferecem o sistema de saúde privado é a telefonia. Porém, existem outras opções como agendar o procedimento pessoalmente e/ou, recentemente, através da internet (LIMA, 2002). Mas, como o objetivo dessas unidades de saúde é sempre oferecer o melhor serviço, desde o acolhimento, até o atendimento e recuperação total do paciente diante a sua dificuldade apresentada,



se observar maior conduta nos protocolos de uso que são documentos normativos de escopo mais estrito, que estabelecem critérios, parâmetros e padrões para a utilização de uma tecnologia específica em determinada doença ou condição (MINISTÉRIO DA SAÚDE, 2020).

O que este trabalho propõe é atuar na perspectiva das inovações tecnológicas no ramo do sistema de saúde (Hospitais, Clínicas, etc), provendo mais praticidade no gerenciamento dos atendimentos, evitando o estresse e desgaste físico provocado aos pacientes e profissionais e promovendo uma interação entre as partes envolvidas por meio de chats virtuais, evitando o gasto de tempo gerado pelos atrasos e atendimentos presenciais.

## 2 JUSTIFICATIVA

O presente trabalho foi elaborado visando as dificuldades na área da saúde, bem como suas dificuldades para atender a demanda contínua de pacientes que buscam atendimento especializado, e a gestão dos próprios profissionais no que diz respeito ao gerenciamento do atendimento em sua totalidade. O foco deste trabalho é voltado à duas vertentes: **Melhorar o acesso da população ao serviço de saúde**, diminuindo a necessidade de filas, marcação de consultas de forma presencial, demoras no atendimento e **melhorar o relacionamento paciente-profissional**, promovendo um espaço exclusivo para que os usuários possam interagir de forma menos formal através de um chat.

O longo período de espera de um paciente em uma lista de espera pode repercutir no paciente e sua família, no profissional de saúde, hospital, sistema de saúde e toda a sociedade dependente deste sistema. Para o paciente, essa espera, além de angustiante por não ter seu problema tratado de imediato, pode causar complicações como o agravamento do seu estado inicial e até mesmo a morte. Para o profissional de saúde e para o hospital, essa espera acarreta maior complexidade dos procedimentos necessários, devido principalmente à demora na intervenção inicial (SENNA, 2020).

Desde o agendamento, por exemplo, a preocupação com o bem-estar do paciente deve ser claramente manifestada pela comunicação verbal, fazendo com que ele crie a confiança de que será bem atendido. Uma boa comunicação com o paciente é indispensável para o atendimento humanizado e de excelência, seja por parte das secretárias, médicos, enfermeiros, auxiliares, etc. (HIDOCTOR, 2020).

Diante disso, fica evidente a necessidade de criar melhorias na área da saúde, principalmente, na área privada, que visem facilitar o acesso ao sistema de saúde, aproximando o paciente ao profissional de saúde e proporcionando a população brasileira um maior conforto, tanto físico quanto psicológico, bem como a satisfação de ser usuário de um sistema que funcione de forma eficaz, mas sem esquecer dos fatores sócio individuais de seus segurados, assim, trazendo mais benefícios para os mesmos.

### **3 OBJETIVOS**

#### **3.1 Objetivo Geral**

Construir uma ferramenta de apoio que auxilie no atendimento do sistema privado de saúde, de forma fácil e dinâmica, melhorando a forma de interação entre profissional de saúde e paciente, e que ajude na forma de gerenciamento das unidades de atendimento, tornando o sistema padronizado e funcional, possibilitando ao usuário conseguir atendimento médico mais rápido.

#### **3.2 Objetivos Específicos**

- 1** Fornecer um sistema dinâmico para a rede privada de saúde na sua forma de atendimento;
- 2** Proporcionar, a partir desse sistema, uma maior interação entre paciente e profissional de saúde.
- 3** Atuar de forma positiva na realização do atendimento nas unidades privadas de saúde.

## **4 FUNDAMENTAÇÃO TEÓRICA**

Cientificamente, já foi possível consolidar que a utilização de tecnologias altera positivamente o cotidiano das pessoas, tanto no ambiente profissional, como fora dele, pois, as relações interpessoais estão em contínua reconfiguração, com sucessivas modificações no ritmo de vida pessoal profissional (SILVA; FERREIRA, 2009).

A inclusão de novas tecnologias na saúde vem imprimindo mudanças gradativas no processo de cuidado, e por possuir complexidade e demandas diferenciadas, cria-se um incentivo à busca de conhecimento técnico-científico, enfatizando a necessidade da capacitação, o que pode gerar uma maior proximidade ou afastamento do usuário do serviço de saúde. Também trazem à tona discussões, reflexões e troca de experiências entre os profissionais sobre o cuidado em um ambiente tecnológico (SÁ NETO; RODRIGUES, 2010; SILVA; FERREIRA, 2009).

Nesta seção será abordada uma breve introdução às tecnologias de desenvolvimento e ferramentas utilizadas no projeto como forma de simplificar o entendimento do contexto de como o sistema foi desenvolvido. Na seção 4.1 será apresentado conceitos sobre o sistema operacional Android, usado para o desenvolvimento dos aplicativos móveis, que serão a interface entre o sistema e os usuários. A seção 4.2 abordará os conceitos básicos do NodeJS, utilizado para o desenvolvimento da API REST, responsável por gerenciar todas as regras de negócio do sistema. A seção 4.3 aborda o ReactJS, utilizado para desenvolver o site de administrador do sistema, responsável por gerenciar as informações dos usuários do sistema. A seção 4.4 irá introduzir o conceito de Web Socket, um protocolo da web que foi utilizado para criar um chat de troca de mensagens em tempo real entre os usuários. Na seção 4.5 será abordada a plataforma Heroku, utilizada para disponibilizar a API e o site do administrador do sistema online.

### **4.1 O SISTEMA OPERACIONAL ANDROID**

Podemos definir um sistema operacional como sendo o esquema primordial de todo computador, pois é ele que traduz a linguagem digital para uma linguagem que podemos compreender, controla o teclado, o mouse e os demais dispositivos

além de gerenciar os processos dos programas. Ele se torna essencial, pelo fato de permitir que você interaja e dê “ordens” ao computador (FAUSTINO, 2017).

O Android é uma plataforma abrangente de código aberto projetada para dispositivos móveis sensíveis ao toque como Smartphones e Tablets, e que, também, contém versões para TVs (Android TV), carros (Android Auto) e Smartwatches (Android Wear). Segundo Gargenta (2011), o Android revolucionou os dispositivos móveis, uma vez que foi construído como uma plataforma aberta que separa o hardware do software executando nele, fazendo com que um número bem maior de dispositivos executem os mesmos aplicativos e cria um ecossistema mais rico para desenvolvedores e consumidores.

A plataforma possui o Android SDK, um kit de desenvolvimento de software que fornece as ferramentas necessárias para o desenvolvimento de aplicativos para o sistema Android, sendo o Android Studio o principal ambiente para o desenvolvimento por conter todas as ferramentas necessárias em um só lugar. As linguagens de programação oficiais para o desenvolvimento nativo são **Java** e **Kotlin**, mas atualmente é possível desenvolver aplicativos para Android utilizando ferramentas multiplataforma, como, por exemplo, com o **React Native**, que utiliza JavaScript para desenvolver aplicativos tanto para Android como para IOS (GARGENTA, 2011).

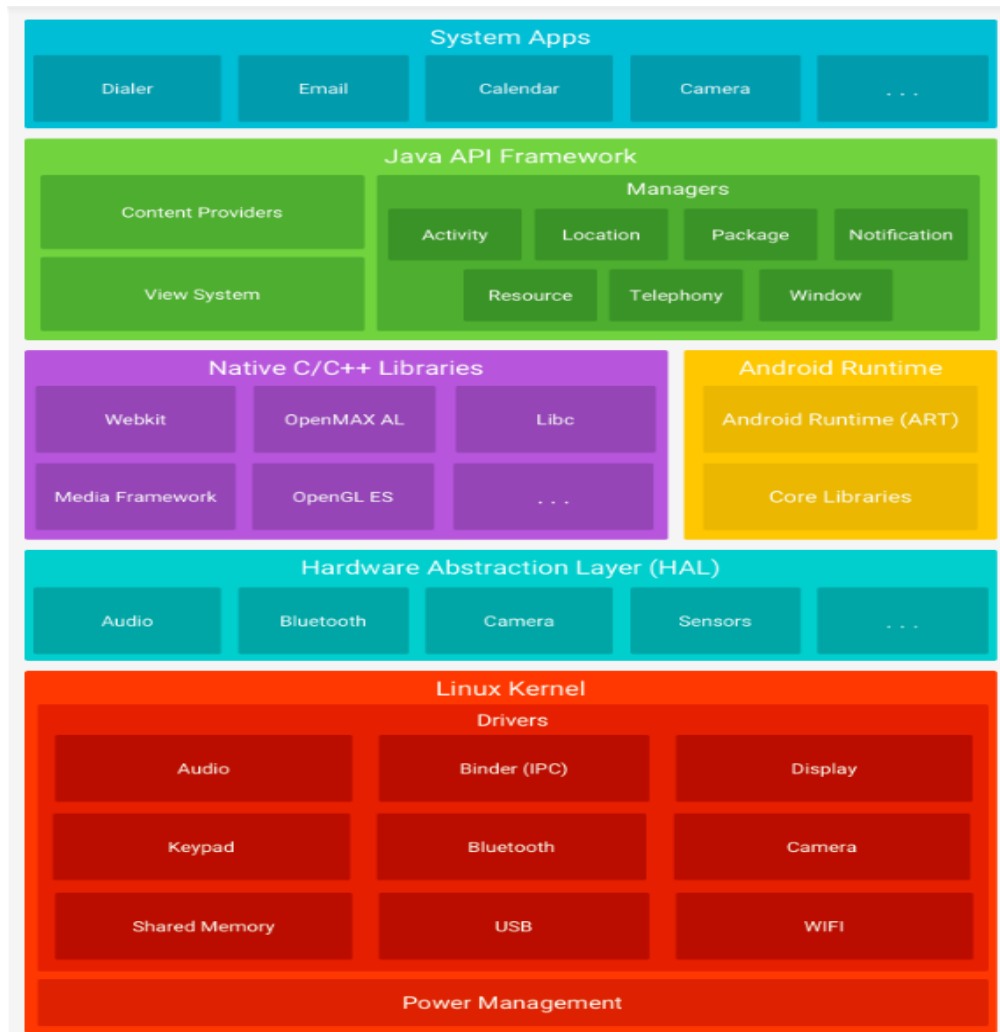
#### 4.1.1 LICENÇA

O projeto de código do Android é de código aberto, e licenciado sob as licenças Apache/Mit. Isso permite que os desenvolvedores e fabricantes de aparelhos tenham acesso a todo código-fonte da plataforma, permitindo que eles vejam o funcionamento do sistema, façam modificações e as utilizem em seus produtos, sem a necessidade de compartilhar as modificações com a comunidade, permitindo que os mesmos tenham versões personalizadas do Android. Todavia, algumas partes do projeto não são de código livre. De acordo com Gargenta (2011), alguns fragmentos de código são proprietários, tais como as pilhas de código para a rede de telefone, Bluetooth, WI-FI, gráficos 3D, GPS câmera, entre outros. Ou seja, mesmo fazendo parte do projeto, esses fragmentos não são de código livre, e, portanto, a modificação desses não é legalizada.

#### 4.1.2 ARQUITETURA

O Android é uma pilha de software com base no Sistema Operacional Linux, que pode ser subdividida em vários componentes. A Figura 1 ilustra como esses componentes se organizam (DEVELOPER, 2020)

**Figura 1** – Arquitetura do sistema operacional Android



**Fonte:** (DEVELOPER, 2020)

De acordo com DEVELOPER (2020), a arquitetura do sistema operacional Android é composta por:

- **Kernel do Linux:** Camada responsável pelo gerenciamento do drivers, memória, câmera, Bluetooth, entre outros. A vantagem de utilizar o Linux como Kernel é aproveitar os vários recursos do próprio Linux, tais como

recursos de segurança e permitir que os fabricantes dos dispositivos desenvolvam drivers de hardware para um Kernel conhecido.

- **Camada de abstração de hardware (HAL):** Fornece interfaces padrão que expõem as capacidades de hardware do dispositivo para a estrutura Java API de maior nível. A HAL consiste em módulos de biblioteca, que implementam uma interface para um tipo específico de componente de hardware, como o módulo de câmera. Quando um Framework API faz uma chamada para acessar o hardware do dispositivo, o sistema Android carrega o módulo da biblioteca para este componente de hardware.
- **Android Runtime (ART):** Projetado para executar várias máquinas virtuais em dispositivos de baixa memória, executando arquivos DEX, um formato de bytecode projetado especialmente para Android, otimizado para oferecer consumo mínimo de memória. Alguns dos recursos principais do ART são:
  - Compilação “Ahead of Time” (AOT) e “Just in time” (JIT).
    - Ahead of Time: Processo de compilação ocorre antes da execução.
    - Just in time: Processo de compilação ocorre durante a execução
  - Coleta de lixo (GC) otimizada.
  - Melhor compatibilidade de depuração, inclusive um criador de perfil de exemplo, exceções de diagnóstico detalhadas e geração de relatórios de erros, além da capacidade de definir pontos de controle para monitorar campos específicos.
- Bibliotecas C/C++ nativas: Vários componentes e serviços do Android, como ART e HAL, são implementados por código nativo que exige bibliotecas nativas programadas em C e C++. A plataforma Android fornece as Java Framework APIs para expor a funcionalidade de algumas dessas bibliotecas nativas aos aplicativos.
- Estrutura da Java API: O conjunto completo de recursos do sistema operacional Android está disponível pelas APIs programadas na linguagem Java. São essas APIs que formam os blocos de programação que são necessários para a criação de aplicativos Android, **simplificando a reutilização de componentes** e serviços de sistema modulares e principais,

inclusive: Um sistema de visualização rico e extensivo (útil para programar a UI de um aplicativo, com listas, grades, caixas de texto, botões e até mesmo um navegador da web incorporado); Um gerenciador de recursos, fornecendo acesso a recursos sem código como strings localizadas, gráficos e arquivos de layout; Um gerenciador de notificação que permite que todos os aplicativos exibam alertas personalizados na barra de status; Um gerenciador de atividade que gerencia o ciclo de vida dos aplicativos e fornece uma pilha de navegação inversa; Provedores de conteúdo que permite que aplicativos acessem dados de outros aplicativos, como o aplicativo de Contatos, o compartilhe os próprios dados.

- **Aplicativos do sistema:** O Android por padrão vem com um conjunto de aplicativos principais para e-mail, SMS, calendários, navegador de internet, contatos, etc. Os aplicativos inclusos na plataforma não têm status especial entre os aplicativos que o usuário opta por instalar. Por esse motivo, um aplicativo terceirizado pode se tornar o navegador da Web padrão, por exemplo. Os aplicativos do sistema funcionam como aplicativos para usuários e fornecem capacidades principais que os desenvolvedores podem acessar pelos próprios aplicativos. Por exemplo, para enviar um SMS, não é necessário programar essa funcionalidade, é possível invocar o aplicativo de SMS que já está instalado para enviar uma mensagem ao destinatário especificado (DEVELOPER, A., 2020).

## 4.2 O NODEJS

O NodeJS é um ambiente de execução JavaScript assíncrono orientado a eventos que foi projetado para o desenvolvimento de aplicações escaláveis de rede, como por exemplo servidores web (NODEJS, 2020). Além disso, ele usa uma arquitetura voltada a eventos não bloqueáveis. As requisições são interpretadas dentro de um loop de eventos (Event-loop) em uma única *thread* e, isso é feito de forma assíncrona não bloqueável, o que torna a aplicação mais rápida e eficiente, principalmente quando se tem um número extenso de requisições (SANTOS, G., 2016).

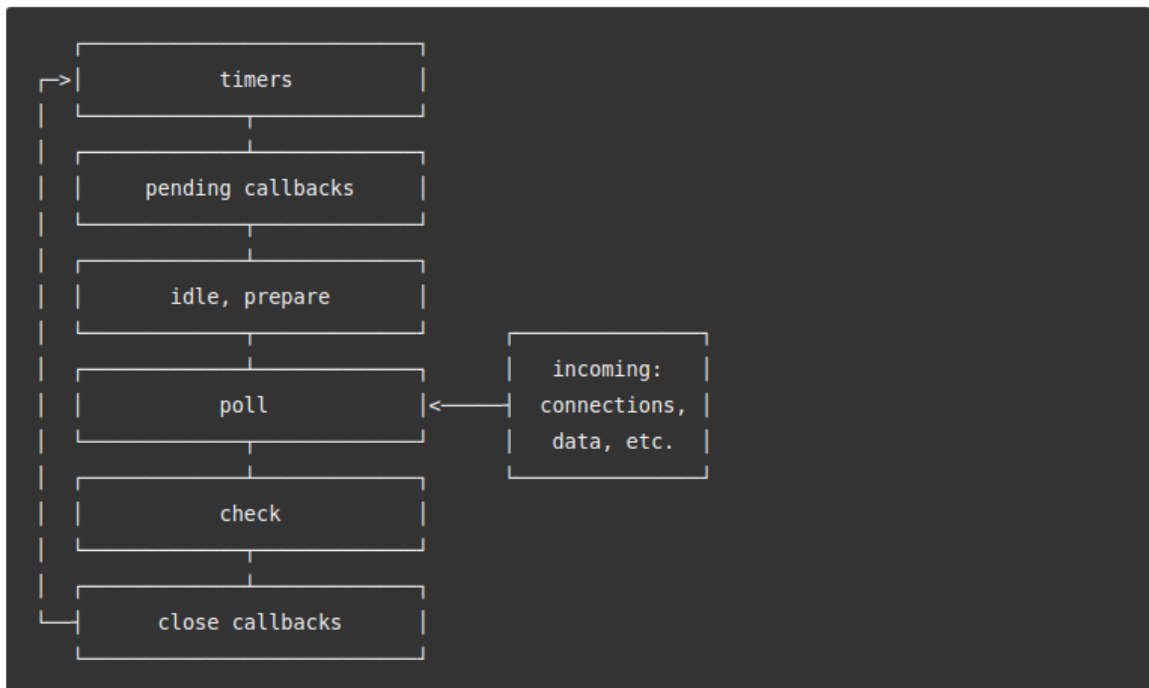
O *Event-loop* é o que permite que o NodeJS execute operações de I/O sem bloqueio apesar do fato de o JavaScript ser *single-thread* (*thread* única). Quando o



NodeJS é iniciado, ele inicializa o *Event-loop*, processa o script de entrada fornecido e, sem seguida, começa a processar o ciclo do evento (NODEJS, 2020).

O Event-loop recebe várias requisições, como buscar alguma informação no banco de dados, ler um arquivo no servidor, etc. Depois de executar uma das ações, em vez de esperar a resposta (como esperar o banco responder), ele passa a processar a próxima requisição. Quando a resposta é retornada, é disparado um evento e a função que deve ser executada como resultado da resposta, é posta na fila para ser executada assim que possível (SANTOS, G., 2016). A figura 2 mostra uma visão geral da ordem de operações do Event-loop.

**Figura 2** – Ordem de operações do *Event-loop*



**Fonte:** (NodeJS, 2020)

Visão geral das fases:

- **Temporizadores:** Esta fase executa as callbacks agendadas por `setTimeout()` e `setInterval()`.
- **Callbacks pendentes:** Executa callbacks de I/O adiados para a próxima iteração do loop.
- **Idle, prepare:** Usado apenas internamente.

- **Votação:** Recupera novos eventos de I/O; Executa callbacks relacionados a I/O (quase todos com exceção de callbacks de fechamento, aqueles programados por temporizadores e `setImmediate()`).
- **Check:** Callbacks `setImmediate()` são invocados aqui.
- **Close callbacks:** Executa as callbacks de fechamento.

Um aplicativo NodeJS é *single-thread*, ou seja, é executado em um único processo, sem criar uma nova Thread para cada solicitação. Quando o NodeJS executa uma operação bloqueante, como acessar um banco de dados ou sistema de arquivos, em vez de bloquear o encadeamento e desperdiçar a espera dos ciclos de CPU, o NodeJS retornará as operações quando a resposta voltar. Isso permite que o NodeJS lide com milhares de conexões simultâneas sem introduzir o fardo de gerenciar a simultaneidade de threads (NODEJS, 2020).

### 4.3 API REST

O conceito de API, que do inglês significa *Application Programming Interface* (interface de programação de aplicações), trata de um conjunto de métodos e padrões estabelecidos em uma aplicação de forma que ela consiga entregar funcionalidades, recursos e serviços de forma simplificada a outras aplicações, que não precisam conhecer a implementação interna dessas funcionalidades, assim escondendo a complexidade por trás dos métodos da API, que existem com o intuito de fornecer acesso amigável as funcionalidades e proporcionam interoperabilidade entre aplicações diferentes (ALVES, R., 2019).

O conceito de REST é simplesmente um modelo que é utilizado para projetar arquitetura de softwares distribuídos que fazem comunicação de dados pela rede (COSTA, H., 2019).

Segundo (GONÇALES, R. L. G., 2019), o REST se baseia em princípios, tais como:

- Estilo arquitetural cliente-servidor, permitindo portabilidade da interface do usuário em múltiplas plataformas e auxilia na escalabilidade.
- Ausência de estado, ou seja, cada requisição do cliente ao servidor deve conter todas as informações necessárias para que o servidor entenda a requisição.

- Interface uniforme de acesso por parte do servidor para que diferentes clientes interajam com o servidor.

#### 4.4 O REACTJS

De acordo com Pandit (2018), o React é uma biblioteca JavaScript de código aberto utilizado para criar interfaces, especificamente Single Page Application (SPA), sendo uma biblioteca capaz de criar grandes aplicações Web, com a capacidade de alterar seu estado e seus dados sem a necessidade de recarregar a página, além do fato de ser rápida, escalável e simples.

O React foi desenvolvido no Facebook e lançado no ano de 2013, sendo utilizado em alguns dos aplicativos mais utilizados do mundo, como o próprio Facebook e o Instagram (COPES, 2019).

##### 4.4.1 SPA

Uma aplicação SPA (*Single Page Application*, do inglês), é uma aplicação que carrega uma única página e todos os arquivos necessários (como JavaScript e CSS) para a aplicação ser executada. Quaisquer interações com a página ou páginas subsequentes não necessitam de requisições para o servidor, o que significa que a página não é recarregada (REACT, 2020).

De acordo com SKÓLSKI (2016), aplicações SPA tem várias vantagens, como:

- Rapidez, pois a maioria dos recursos são carregados apenas uma vez durante a vida útil do aplicativo.
- Desenvolvimento simplificado e agilizado, pois não há a necessidade de escrever código para renderizar páginas no servidor.
- Facilidade de depuração de código pelo Chrome, por exemplo, pois você pode monitorar as operações da rede, investigar os elementos da página e os dados associados a ela.
- SPAs podem armazenar em cache com eficácia. Depois de fazer uma solicitação, o aplicativo armazena todos os dados e então pode usar esses dados até de forma offline.

#### 4.4.2 JSX

No React, em vez de usarmos JavaScript para modelagem, utilizamos o JSX (JavaScript XML). O JSX é uma mistura de JavaScript simples com a linguagem de marcação HTML e usa essa sintaxe de tag para renderizar subcomponentes (PANDIT, 2019).

### 4.5 WEB SOCKETS

Web Socket é um protocolo de comunicação entre navegadores e servidores. No protocolo Web Socket a conexão é constante, em contraste com solicitações HTTP tradicionais, que são sempre iniciadas pelo cliente, o que significa que o servidor notificará o cliente sempre que houver atualizações. Ao manter uma conexão aberta entre o cliente e o servidor, as atualizações serão atualizadas em tempo real, sem que os clientes precisem fazer pesquisas em determinados intervalos, atualizar a requisição, etc. Este é o principal fator que torna o Web Socket ideal para aplicativos em tempo real para os quais os dados precisam ser disponibilizados imediatamente (MARDAN, A., 2018).

#### 4.5.1 Diferenças entre o protocolo HTTP e o Web Socket

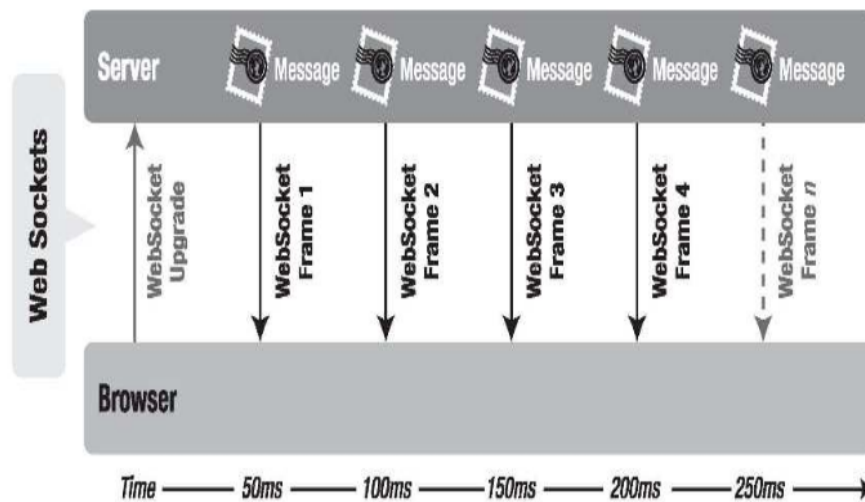
A comunicação entre cliente e servidor é feita geralmente com base em conexões HTTP que requerem cabeçalhos de informações anexados à solicitação do cliente e resposta do servidor. De acordo com a definição do protocolo HTTP, esses cabeçalhos contêm algumas informações de controle de transmissão, como tipo de protocolo, versão do protocolo, tipo de navegador, entre outras. (LIU, Q; SUN, X., 2012).

O *Hypertext Transfer Protocol* (HTTP) é a base de todas as comunicações na rede mundial de computadores. Ele é combinado ao *Transmission Control Protocol* (TCP), que segue o princípio de solicitação-resposta, ou seja, o navegador (cliente) solicita uma requisição ao servidor, e este retorna uma resposta com as informações solicitadas (SRINIVASAN L; SCHARNAGL, J; SCHILLING, K., 2013).

No protocolo Web Socket, uma vez que a conexão é estabelecida com sucesso, o cliente pode se comunicar com o servidor livremente e apenas dois bits de controle de informações são anexadas aos dados necessários do usuário final que é codificado por UTF-8. Esta definição de Web Socket diminui consideravelmente largura de banda e tempo consumido pelo processamento de

cabeçalhos em conexões, então levam a menos tráfego de rede e menor latência. Estas são as razões exatas pelas quais o WebSocket é mais adequado para a comunicação web em tempo real (LIU, Q; Sun, X., 2012). A figura 3 ilustra o funcionamento do protocolo Web Socket.

**Figura 3** – O protocolo Web Socket



**Fonte:** (ZHAO; MAN, Shu; AL, Et).

#### 4.6 O HEROKU

O Heroku é uma plataforma de desenvolvimento e entrega de aplicativos baseada na nuvem que ajuda os desenvolvedores na criação, gerenciamento, configuração e manutenção de servidores, removendo a responsabilidade de administração do desenvolvedor (KEMP, C.; GYGER, B. 2013).

A plataforma Heroku oferece recursos para a implantação quase imediata para o desenvolvimento de aplicativos de produção, utiliza um sistema integrado para o controle de versão de dependências do projeto utilizando o GIT, dispõe de opções para desenvolvimento de aplicativos por meio de linha de comando, online e em dispositivos móveis, por meio de uma API.

Os aplicativos criados no Heroku se beneficiam da arquitetura de autocorreção, garantindo que eles não exijam que os administradores de sistemas gerenciem a plataforma subjacente para mantê-los seguros e confiáveis (KEMP, C.; GYGER, B. 2013).

## 5 METODOLOGIA

Nesta seção serão apresentadas as principais etapas para a realização deste trabalho. A pesquisa pode ser classificada como exploratória, pois no processo de desenvolvimento do sistema, foi necessário um estudo sobre o panorama da saúde público-privada no Brasil, o estudo de casos de uso para uma melhor compreensão do comportamento do sistema, o estudo do ambiente de desenvolvimento Android, usado para a criação de aplicativos móveis, do framework NodeJS, usado para a criação de uma API REST e o framework ReactJS, para o desenvolvimento de um web site.

### 5.1 ESTUDO DO DIAGRAMA DE CASOS DE USO

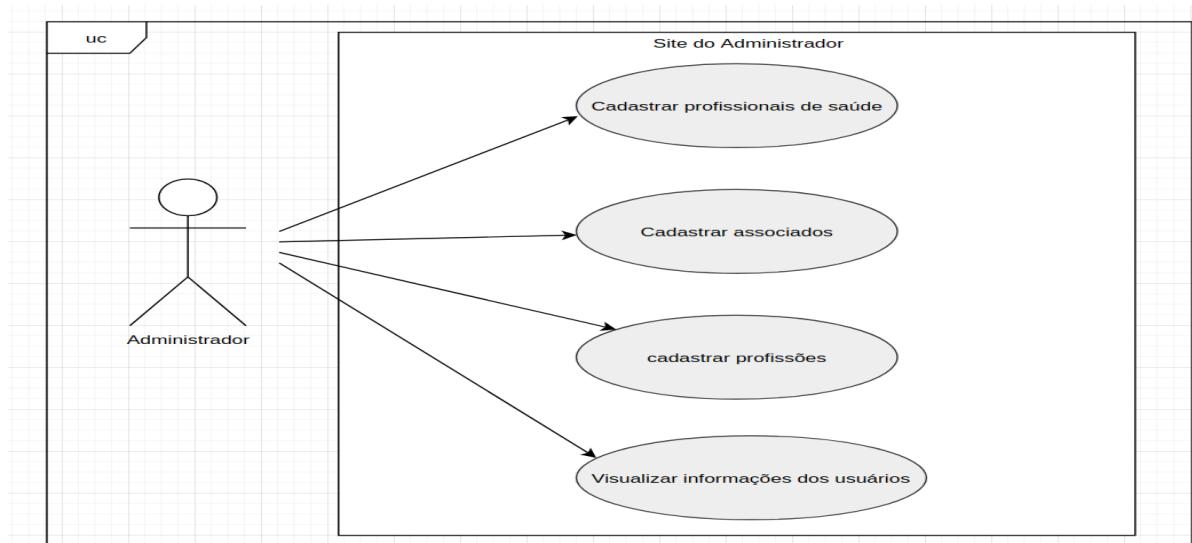
Nesta etapa foi realizado o estudo de casos de uso do sistema, através da elaboração de diagramas de casos de uso, que possibilitaram um melhor entendimento de como a aplicação deveria se comportar e quais ações os usuários poderiam executar, acelerando assim o processo de desenvolvimento.

O diagrama de casos de uso possibilita a compreensão do comportamento do sistema por pessoas sem conhecimento algum sobre o problema, apresentando o sistema da perspectiva dos usuários. O diagrama tem por objetivo apresentar a visão externa geral das funcionalidades que o sistema deverá oferecer aos usuários, sem se preocupar em como as funcionalidades serão implementadas. Este diagrama é de grande auxílio para a identificação e compreensão dos requisitos funcionais do sistema, ajudando a especificar, visualizar e documentar as funções e serviços do software (GUEDES, 2018).

O diagrama de casos de uso do sistema é composto de três atores que ilustram ações de cada usuário do sistema:

O **primeiro ator** é o usuário administrador, responsável por administrar o sistema. É ele quem cadastra um novo profissional de saúde no sistema, associados da empresa, profissões e fiscaliza as informações de todos os usuários do sistema.

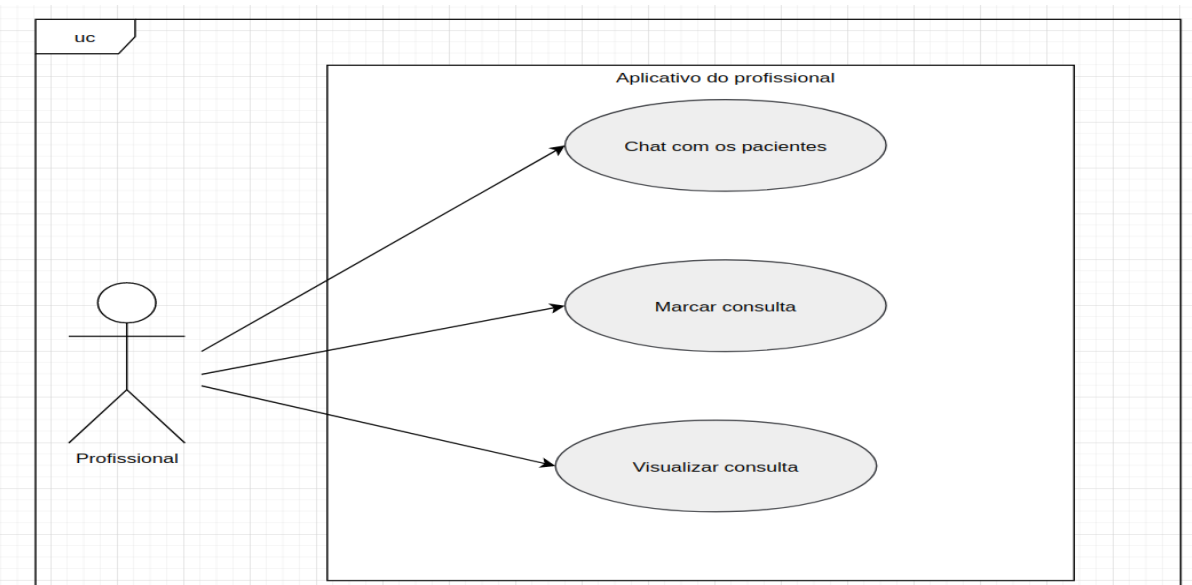
**Figura 4** – Diagrama de casos de uso do administrador do sistema



**Fonte:** Elaborada pelo próprio autor, 2020.

O **segundo ator** é o profissional de saúde, que pode ser um médico, fisioterapeuta, psicólogo, entre outras profissões permitidas pelo sistema. O profissional de saúde pode conversar com os pacientes através de um chat virtual, marcar consultas diretamente pelo aplicativo e visualizar consultas marcadas e concluídas.

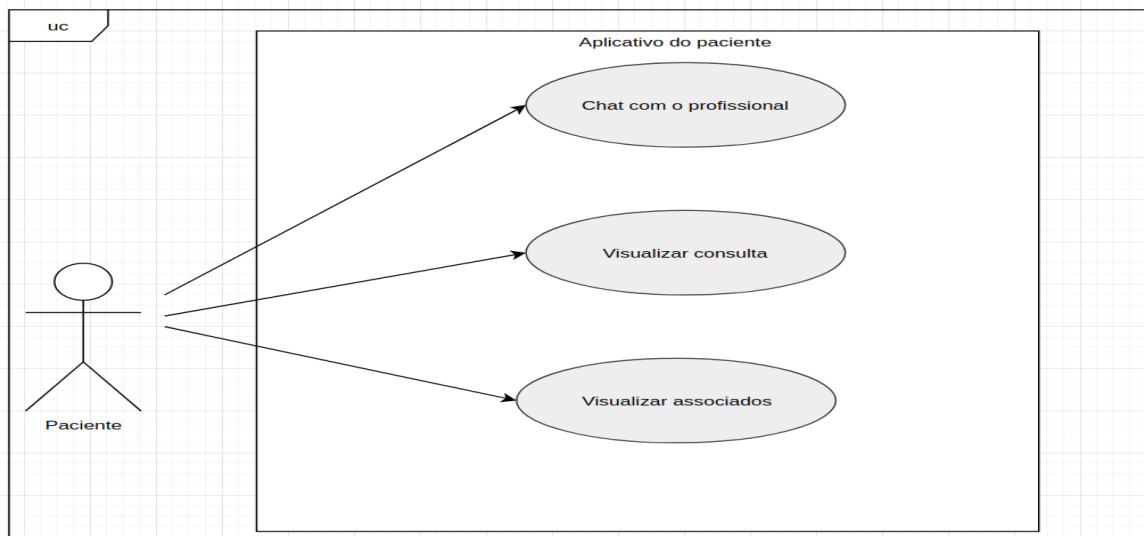
**Figura 5** – Diagrama de casos de uso do profissional de saúde



**Fonte:** Elaborada pelo próprio autor, 2020.

O **terceiro ator** do sistema é o paciente. O paciente pode ser qualquer pessoa que deseja ser atendido pelos profissionais do sistema. Este usuário terá a sua disposição um chat com todos os profissionais, onde poderá tirar dúvidas, solicitar consultas presenciais, visualizar as consultas marcadas e obter informações sobre os associados da empresa, como Farmácias.

**Figura 6** – Diagrama de casos de uso do paciente



**Fonte:** Elaborada pelo próprio autor, 2020.

## 5.2 ESTUDO DO AMBIENTE DE DESENVOLVIMENTO ANDROID

Nesta etapa foi realizado um estudo detalhado sobre o sistema Android (Seção 4.1), que foi a plataforma escolhida para o desenvolvimento dos aplicativos móveis (aplicativo do paciente e do profissional de saúde) e o passo-a-passo da utilização do Android Studio.

O Android Studio é a IDE (Ambiente de desenvolvimento integrado) oficial para o desenvolvimento de apps para o sistema Android. Além da edição de código, o Android Studio possibilita a compilação flexível do código baseado em Gradle, um emulador rápido com inúmeros recursos, a aplicação de mudanças de código e recursos ao aplicativo em execução sem reiniciar o app, frameworks e ferramentas de testes, ferramentas para detectar problemas de desempenho, usabilidade, compatibilidade com versões, entre outros (DEVELOPER, A., 2020).



### 5.3 ESTUDO DA CRIAÇÃO DE UMA API REST

Nesta etapa foi feito um estudo detalhado sobre a criação de uma API REST (Seção 4.3). No contexto do sistema, a API REST é responsável por fornecer e gerenciar as informações dos usuários do sistema através de requisições HTTP. A pesquisa foi centralizada em ferramentas que pudessem oferecer um desenvolvimento fácil, rápido, escalável e que atendesse demandas mais específicas, como a utilização de Web Sockets para a criação de **chats virtuais** em tempo real sem a necessidade de configurações muito complexas. A ferramenta selecionada e utilizada no desenvolvimento foi o ambiente de execução NodeJS, descrito na seção 4.2, por conter várias bibliotecas de fácil utilização e configuração e por ser mais rápido e eficiente para um conjunto extenso de requisições. A seguir serão descritas as principais bibliotecas utilizadas para o desenvolvimento da API REST.

#### 5.3.1 O Framework Express

O Express é uma das bibliotecas para NodeJS mais populares para lidar com solicitações HTTP diferentes em uma URL específica. Além disso, é minimalista, de código aberto e flexível, o que ajuda o desenvolvedor a investir menos esforço e tempo no desenvolvimento de sites e aplicativos ainda melhores. A figura 7 ilustra o código responsável por atender uma requisição HTTP e retornar uma resposta ao usuário (ARIF, 2020).

**Figura 7** – Requisição HTTP com Express

```
1  const express = require('express')
2  const app = express()
3
4  app.get('/', (req,res)=> { //get method
5    res.send('Hello World') //send response
6  })
7  app.listen(3000)
```

**Fonte:** (ARIF, 2020)

### 5.3.2 O Framework Socket-io

O Socket-io é uma biblioteca de comunicação bidirecional baseada em eventos para aplicativos da web em tempo real. Ela abstrai muitos transportes, incluindo o AJAX long-polling e Web Sockets (Seção 4.5), em uma única API e permite que os desenvolvedores enviem e recebam dados em tempo real sem se preocupar com a compatibilidade entre navegadores. O Socket-io foi utilizado para a criação do chat virtual do sistema. A figura 8 ilustra o código utilizado para criar um chat em tempo real (KHACHATRYAN, 2016).

**Figura 8** – Enviando mensagens com o Socket-io

```

1  const io = socket(server);
2
3  io.on("connection", (client) => {
4    console.log(`User ${client.id} connected`);
5
6    client.on("sendMessageAll", (args) => {
7      client.broadcast.emit("receiveMessageAll", {
8        "id": client.id,
9        "author": args.author,
10       "message": args.message
11     });
12   });
13
14   client.on("disconnect", () => {
15     io.emit("disconnected", "User disconnected");
16   });
17 });

```

**Fonte:** Elaborada pelo próprio autor, 2021.

## 5.4 ESTUDO DA CRIAÇÃO DE UM WEB SITE

Nesta etapa foi feito um estudo sobre as ferramentas de desenvolvimento de web sites para a criação de uma interface de sistema que fosse responsável por gerenciar as informações dos usuários do sistema, como cadastrar profissionais de saúde, associados do sistema e administrar o acesso dos usuários ao sistema.

A ferramenta selecionada para o desenvolvimento da interface web foi o ReactJS (Seção 4.4) por ser um framework que otimiza o tráfego com o servidor, evitando requisições HTTP repetidas, que utiliza as SPAs (Seção 4.4.1) e facilita o desenvolvimento da aplicação por utilizar JSX (Seção 4.4.2), que usa JavaScript e HTML para criar componentes que podem ser reutilizados na aplicação, evitando repetição de códigos. A figura 9 exemplifica o código JSX usado pelo ReactJS.

**Figura 9** – Exemplo de código JSX

```
1  import React from 'react';
2  import { BrowserRouter } from 'react-router-dom';
3  import Router from './routes/router';
4  import AlertInfo from './components/Alert/index';
5  import 'bootstrap/dist/css/bootstrap.min.css';
6  import './global.css';
7
8  function App() {
9    return (
10     <>
11       <BrowserRouter>
12         <Router />
13       </BrowserRouter>
14       <AlertInfo />
15     </>
16   );
17 }
18
19 export default App;
```

**Fonte:** Elaborada pelo próprio autor, 2021.

## 6 RESULTADOS E DISCUSSÕES

O desenvolvimento do sistema foi dividido em três etapas principais: **O desenvolvimento do back-end (API REST), o desenvolvimento dos aplicativos dos usuários e o desenvolvimento de um site de administrador do sistema.** Essas etapas serão apresentadas nesta seção, incluindo os resultados obtidos em cada uma delas.

### 6.1 FERRAMENTAS UTILIZADAS E TRECHOS DE CÓDIGO

Nesta seção serão apresentados trechos de códigos utilizados no desenvolvimento do sistema, como o código responsável por tratar as requisições HTTP e enviar uma resposta através da API REST, verificação dos usuários nos aplicativos móveis e o envio de informações para o servidor a partir do web site.

#### 6.1.1 Tratando requisições HTTP

A API REST do sistema fornece e gerencia os dados da aplicação através de requisições HTTP, enviadas pelos usuários ao executar uma ação. As requisições são feitas através de caminhos do servidor, conhecidos como rotas que, por sua vez, estão associadas a uma funcionalidade.

A figura 10 ilustra o código responsável por tratar as requisições HTTP.

**Figura 10** – Código de rotas da API

```

25
26 routes.post('/registrar/paciente/', multer(multerConfigs).single('photo') ,pacienteController.store);
27
28 routes.post('/login/paciente/', pacienteController.login);
29
30 routes.get('/paciente/:id/', [authentication], pacienteController.getById);
31
32 routes.patch('/paciente/:id/editar/', [authentication, multer(multerConfigs).single('photo')] , pacienteController.update);
33
34 routes.get('/paciente/:id/consulta/', [authentication], pacienteController.getConsultas);
35
36 routes.get('/paciente/:id/profissionais/', [authentication], pacienteController.getAllProfissionais);
37
38 routes.get('/paciente/:id/associados/', [authentication], pacienteController.getAllAssociados);
39

```

**Fonte:** Elaborada pelo próprio autor, 2020.

#### 6.1.2 Verificação de usuário autenticado

Ao iniciar o aplicativo pela primeira vez, o usuário precisará se autenticar para ter acesso as funcionalidades, mas após a primeira autenticação o usuário não precisará se autenticar novamente. Isso acontece graças a uma funcionalidade do

sistema que permite salvar as informações do usuário internamente, sem que seja necessário se autenticar sempre que acessar o aplicativo.

Na Figura 11 é mostrado o trecho de código Java que analisa se o usuário está logado no aplicativo, e o redireciona.

**Figura 11** – Verificação de usuário autenticado

```

44     @Override
45     protected Boolean doInBackground(Void... voids) {
46         boolean result = false;
47         try {
48             Thread.currentThread();
49             Thread.sleep(SPLASH_SCREEN_TIME);
50
51             if (mUserSaved.contains("userEmail") && mUserSaved.contains("userPassword")) {
52                 result = true;
53             }else {
54                 result = false;
55             }
56         } catch (InterruptedException e) {
57             e.printStackTrace();
58         }
59         return result;
60     }
61
62     @Override
63     protected void onPostExecute(Boolean result) {
64         super.onPostExecute(result);
65
66         Intent intent;
67
68         if(result){
69             intent = new Intent(getBaseContext(), MainActivity.class);
70             intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
71             startActivity(intent);
72             finish();
73         }else{
74             intent = new Intent(getBaseContext(), LoginActivity.class);
75             intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
76             startActivity(intent);
77             finish();
78         }

```

**Fonte:** Elaborada pelo próprio autor, 2020

### 6.1.3 Envio de dados através do web site

O web site do sistema engloba todas as ações destinadas para o administrador do sistema, e para cada ação são trocadas informações com o servidor do sistema.

A Figura 12 mostra o trecho de código Javascript responsável por enviar os dados de cadastro de um novo associado da empresa para a API REST.

**Figura 12** – Enviando dados através do web site

```

41     const formData = new FormData();
42     formData.append('photo', associado.logo, associado.logo.filename);
43     formData.append('name', associado.name);
44     formData.append('type', associado.type);
45     formData.append('cnpj', associado.cnpj);
46     formData.append('phone_main', associado.phone_main);
47     formData.append('phone_secondary', associado.phone_secondary);
48     formData.append('email', associado.email);
49     formData.append('latitude', associado.latitude);
50     formData.append('longitude', associado.longitude);
51
52     api
53       .post(`admin/${id}/registrar/associado`, formData, config)
54       .then((response) => {
55         // eslint-disable-next-line no-console
56         console.log(response.data);
57       })
58       .catch((error) => {
59         // eslint-disable-next-line no-console
60         console.log({ error });
61       });
62   };

```

Fonte: Elaborada pelo próprio autor, 2020.

## 6.2 FUNCIONALIDADES DO SISTEMA

O sistema foi desenvolvido de acordo com os requisitos funcionais, descritos no APÊNDICE A, e não-funcionais, descritos no APÊNDICE B.

Foram definidos três tipos de usuários (**paciente**, **profissional de saúde** e **administrador**), cada um com suas respectivas ações, e três ambientes distintos foram criados com suas funcionalidades:

- Para o usuário **paciente** é permitido se cadastrar no sistema, fazer login, conversar com os profissionais de saúde, visualizar as consultas marcadas e visualizar os associados da empresa com as informações de localização e contato;

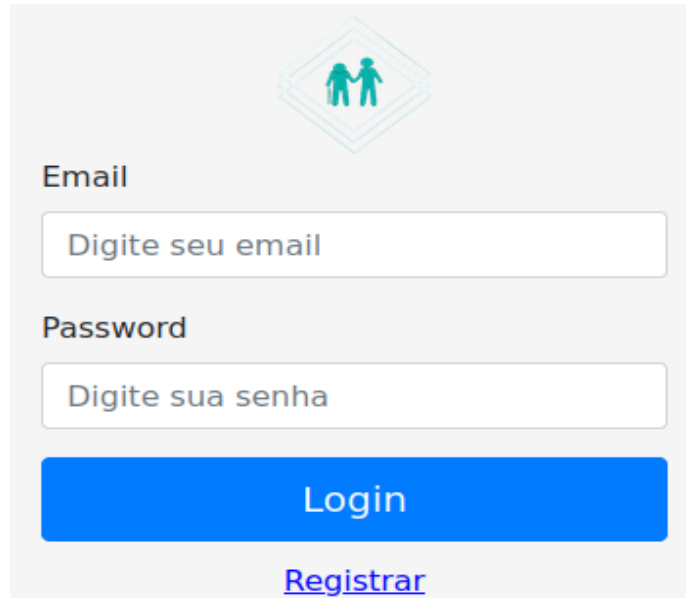
- Para o usuário **profissional** é permitido fazer login no sistema, conversar com os pacientes, marcar consultas, visualizar suas consultas e visualizar gráfico de número de consultas;

- Para o usuário **administrador** é permitido se cadastrar no site, fazer login, cadastrar novos profissionais, cadastrar novas profissões atendidas pelo sistema, cadastrar novos associados da empresa e visualizar informações sobre os pacientes, profissionais e associados. Todas as funcionalidades do sistema serão descritas a seguir.

**Tela de Login do administrador:**

A partir dessa tela é possível ao administrador do sistema se autenticar, digitando suas credenciais de acesso, e se cadastrar no sistema.

**Figura 13** – Tela de acesso do administrador

A imagem mostra a interface de login para um administrador. No topo, há um ícone de duas pessoas em um losango. Abaixo, há um campo de texto rotulado "Email" com o placeholder "Digite seu email". Logo abaixo, há um campo de texto rotulado "Password" com o placeholder "Digite sua senha". Na base, há um botão azul com o texto "Login" e um link azul rotulado "Registrar" abaixo dele.

Email

Digite seu email

Password

Digite sua senha

Login

[Registrar](#)

**Fonte:** Elaborada pelo próprio autor, 2020.

**Tela de cadastro do administrador:**

A partir dessa tela é possível cadastrar um novo administrador do sistema. As informações necessárias para cadastrar um administrador são: Nome, data de nascimento, CPF, e suas credencias de acesso que são seu e-mail e uma senha. A imagem a seguir mostra a tela de cadastro de administradores.

**Figura 14** – Tela de cadastro do administrador.

Cadastrar

---

Nome

Data de nascimento

CPF

Email

Password

Salvar

**Fonte:** Elaborada pelo próprio autor, 2021.

#### **Tela de informações dos usuários:**

Esta é a tela que é aberta logo após a autenticação. A partir dela, o administrador pode verificar e analisar as informações de cada usuário do sistema (pacientes e profissionais), bem como as empresas associadas. Vale ressaltar que o administrador não tem permissão para alterar os dados pessoais dos pacientes do sistema, nem ter acesso a informações sensíveis como senhas de acesso, ele terá acesso apenas as informações de cadastro que são necessárias para identificação do paciente. Mas caso o administrador identifique alguma ação dos usuários que não esteja de acordo com as regras da empresa, ele poderá desativar o usuário em questão, removendo seu acesso ao sistema. A Figura 15 ilustra a tela de informações dos usuários.



**Figura 15** – Tela de informações dos usuários

Admin Cadastrar ▾ <span style="float: right;">admin@sidoso.com</span>						
Pacientes						
Nome	Dt. Nascimento	Tel. I	Tel. II	Sexo	Email	Ativo
Kaio giovanni	1996-03-13	083 99602-4107		M	kaio.teste@gmail.com	<input checked="" type="checkbox"/>
Kaio giovanni bezerra de lucena	2020-02-10	(000)00000-0000		M		<input checked="" type="checkbox"/>
Paciente teste	1899-11-30	(085)23669-7411	(333)69977-0451	M	paciente@teste.com	<input checked="" type="checkbox"/>
Mariana Oliveira	1899-11-30	(041)37945-7975			mari.oliveira@paciente.com	<input type="checkbox"/>
Lucas Silva	1899-11-30	(081)37487-6796			lucas.silva@paciente.com	<input type="checkbox"/>
Profissionais						
Nome	Dt. Nascimento	Tel. I	Sexo	Profissão	Email	Ativo
Carlos Alberto	1990-05-12	083 99999-0000	M	Médico	carlos.medico@gmail.com	<input checked="" type="checkbox"/>
Deborah Lima	2000-05-30	083 99999-4058	F	Fisioterapeuta	deby.fisio@gmail.com	<input checked="" type="checkbox"/>
Kaio Giovanni	2000-10-20	083 99999-4058	M	Fisioterapeuta	kaio.fisio@gmail.com	<input checked="" type="checkbox"/>

**Fonte:** Elaborada pelo próprio autor, 2020.

### Tela de cadastro de profissionais de saúde:

Os profissionais de saúde do sistema são cadastrados pelo administrador, como forma de impedir que um usuário mal-intencionado possa se cadastrar no sistema e se passar por um profissional de saúde. Dessa forma, os pacientes podem ter uma maior segurança sobre os profissionais de saúde que estão lhe atendendo. Para cadastrar um profissional de saúde, o administrador precisa inserir dados pessoais do profissional, como nome, CPF, números de contato, e-mail, profissão, e as suas respectivas credenciais de acesso ao sistema. que insere seus dados pessoais e profissionais. A Figura 16 mostra o formulário de cadastro de profissionais de saúde.

**Figura 16** – Tela de cadastro de profissionais de saúde

Cadastrar profissional da saúde

Foto  
Escolher ficheiro Nenhum ficheiro selecionado

Nome  
Nome completo

Data de nascimento  
dd/mm/aaaa

CPF  
CPF

Sexo  
Masculino

Telefone I  
Telefone principal

Telefone II  
Telefone secundário

Profissão  
Médico

Email  
Email

Password  
Senha de login

Salvar

**Fonte:** Elaborada pelo próprio autor, 2020.

### **Tela de cadastro de associados:**

Os associados do sistema, são empresas que tem interesse em se associar e oferecer serviços externos aos pacientes do sistema. Dessa forma, o sistema disponibiliza aos seus pacientes um sistema que abrange muito mais áreas de atuação, como por exemplo: farmácias, centros de reabilitação, centros esportivos, entre outros, que poderão ser indicados pelos profissionais de saúde em casos especiais. Os associados são cadastrados no sistema pelo administrador, e logo após a inserção dos dados, os associados terão suas informações de contato e localização disponíveis para os pacientes do sistema. A Figura 17 ilustra a tela de cadastro de associados do sistema.

**Figura 17 – Tela de cadastro de associados**

### Cadastrar associado

Logo da empresa

Nenhum ficheiro selecionado

Nome

Nome social

Tipo

Farmácia, etc

CNPJ

CNPJ da empresa

Telefone I

Telefone principal

Telefone II

Telefone II

Email

Email

Click no mapa para selecionar a localização da empresa



**Fonte:** Elaborada pelo próprio autor, 2020.

### **Tela de cadastro de profissões:**

Para que o administrador cadastre um novo profissional de saúde no sistema, ele precisa inserir a profissão do profissional em questão. No sistema, essa funcionalidade existe para impedir que sejam cadastrados profissionais com profissões erradas ou não coniventes com a realidade do sistema de saúde, ou seja, o cadastro de profissões nada mais é que um cadastro para controle das profissões que o sistema atende (Figura 18).

**Figura 18** – Tela de cadastro de profissões

Cadastrar profissão no sistema

Nova Profissão

Profissão

Salvar

**Fonte:** Elaborada pelo próprio autor, 2020.

**Tela de login dos usuários:**

Os aplicativos do paciente e do profissional de saúde dispõem de uma tela para que os usuários possam se autenticar e utilizar o sistema como um usuário da aplicação. As figuras abaixo mostram as telas de login para pacientes e profissionais de saúde.

**Figura 19** – Tela de login dos usuários

7:01

Login

Digite seu email

Digite sua senha

Login

[Não tem uma conta? Crie uma.](#)

**Fonte:** Elaborada pelo próprio autor, 2020.

**Tela de cadastro do paciente:**

Quando o paciente acessa o sistema pela primeira vez, ele precisa realizar o seu cadastro no sistema, para que posteriormente ela possa se autenticar no sistema e assim utilizar as funcionalidades como um usuário da aplicação. A tela de cadastro de pacientes esta ilustrada na Figura 20.

**Figura 20** – Tela de cadastro do paciente



A imagem mostra a tela de cadastro de um paciente em um aplicativo móvel. O layout é o seguinte:

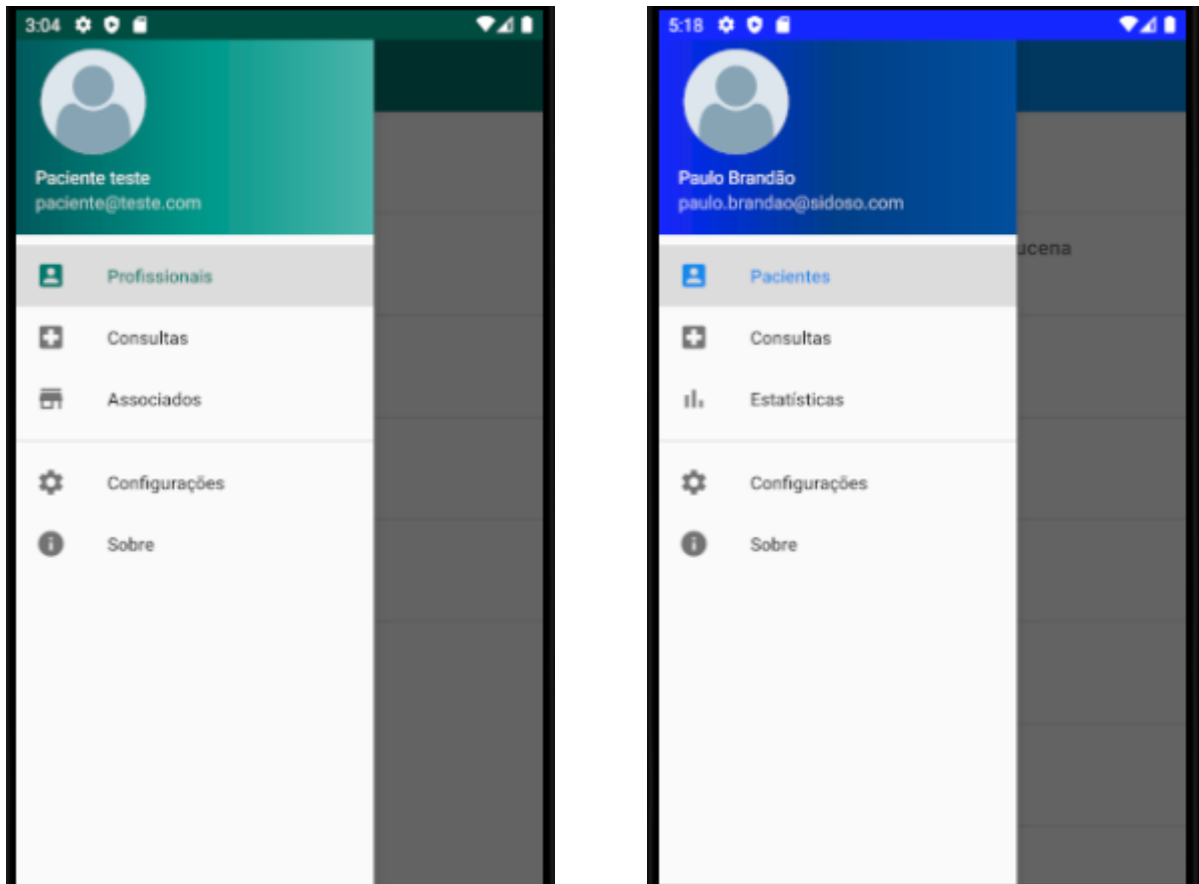
- Barra superior: 7:01, ícones de notificação, Wi-Fi e bateria.
- Formulário de entrada:
  - Campos para "Data de nascimento" e "Digite seu CPF".
  - Opções de gênero: "Masculino" (selecionado) e "Feminino".
  - Campos para "Telefone" e "Telefone II (Opcional)".
  - Campos para "Digite seu email", "Digite sua senha" e "Confirme sua senha".
- Botões de ação: "Cadastrar" e "Cancelar" (ambos em fundo verde).

**Fonte:** Elaborada pelo próprio autor, 2020.

**Tela de navegação do aplicativo:**

Após os usuários se autenticarem no sistema, eles serão direcionados para a tela principal do aplicativo. Nessa tela eles poderão encontrar o menu de navegação, que será o modo pelo qual os usuários poderão navegar em todas as telas do aplicativo e conhecer todas as funcionalidades que o sistema oferece. Cada item do menu corresponde a um módulo, e cada módulo é responsável por uma parte específica do sistema. A Figura 21 apresenta o menu de navegação dos usuários.

**Figura 21** – Tela do menu de navegação

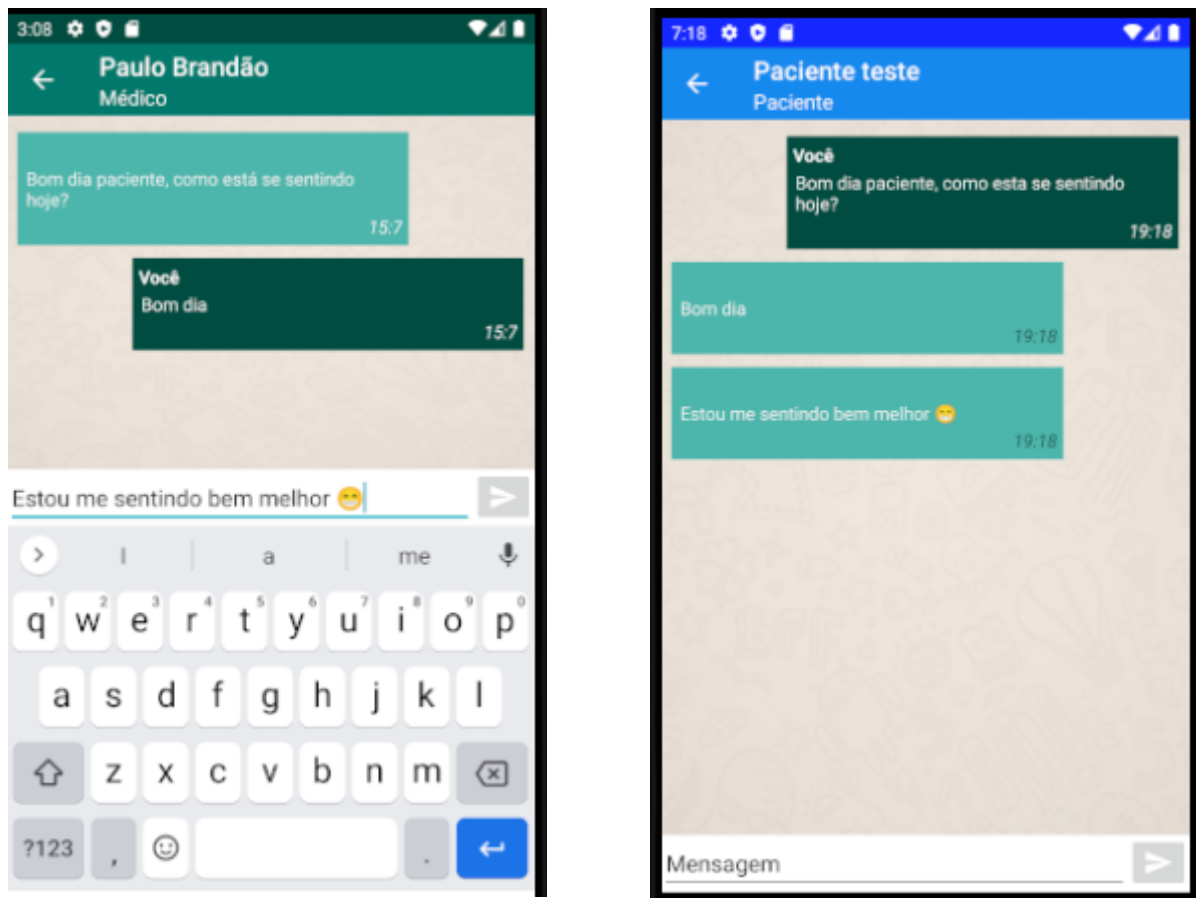


**Fonte:** Elaborada pelo próprio autor, 2020.

#### **Tela de conversa:**

O primeiro item do menu de navegação do sistema é o módulo de conversas. O nome dele vai aparecer de acordo com o tipo de usuário do sistema. Por exemplo, o paciente ao se autenticar no sistema verá uma lista de profissionais de saúde, enquanto que os profissionais verão uma lista de pacientes. Apesar de terem nomes diferentes para cada usuário, a funcionalidade será a mesma para os dois, pois essa é a lista de conversas que o usuário terá acesso no sistema.

Após o usuário selecionar um item na lista de conversas, será aberta a tela de chat propriamente dita, e é por ela que o usuário poderá enviar e receber mensagens de texto em tempo real para outro usuário, ou seja, a partir dela o usuário poderá se comunicar com outros usuários em tempo real. A Figura 22 o ilustra o fluxo de uma conversa para um paciente do sistema.

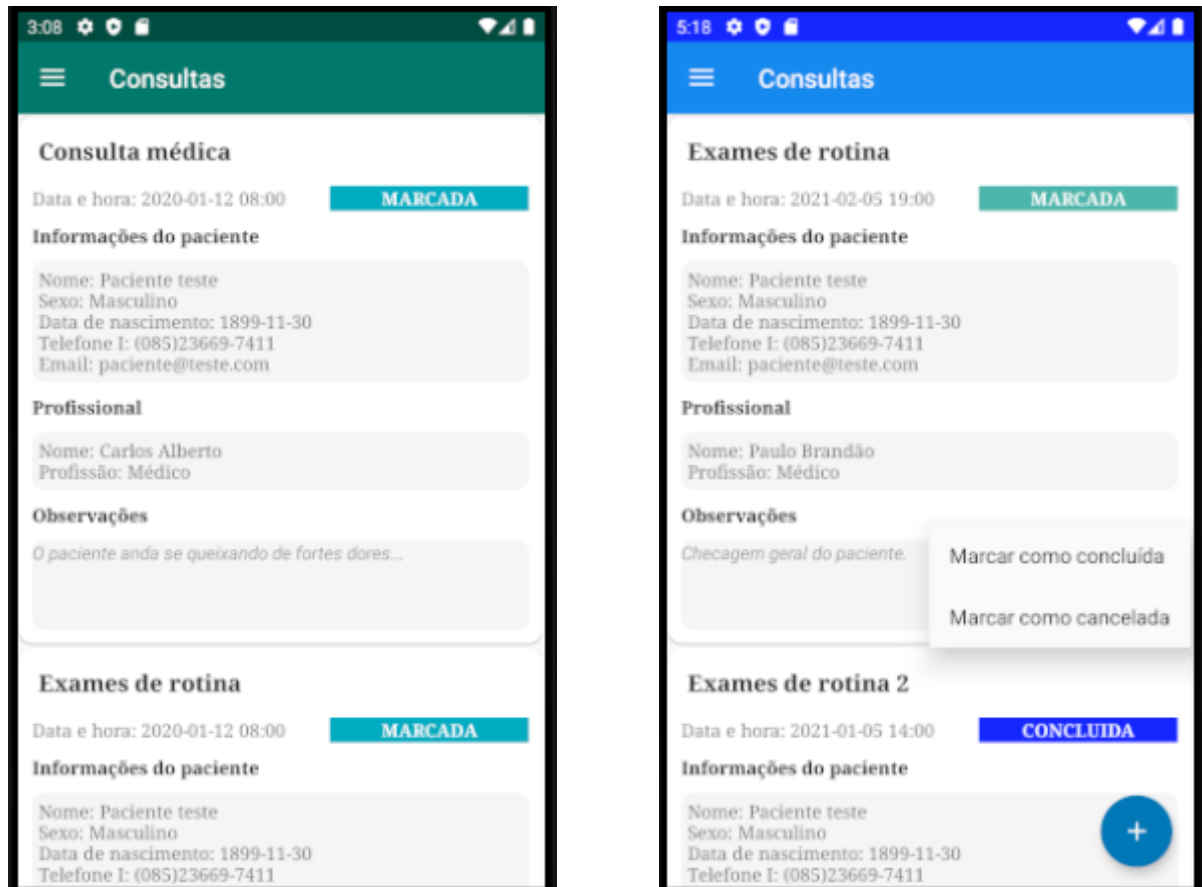
**Figura 22** – Tela de conversa

**Fonte:** Elaborada pelo próprio autor, 2020.

### **Tela de consultas:**

O sistema oferece tanto ao paciente quanto ao profissional de saúde uma visualização de todas as consultas relacionadas a ele, tenham elas sido marcadas, canceladas ou concluídas. A principal diferença para os dois tipos de usuário é que o profissional de saúde tem a possibilidade marcar uma nova consulta para um paciente qualquer através de um botão no canto inferior direito do aplicativo, enquanto que o paciente não pode realizar nenhum tipo de cadastro de consultas. A figura 23 mostra a diferença entre a tela de consultas do paciente e do profissional.

**Figura 23** – Tela de consultas



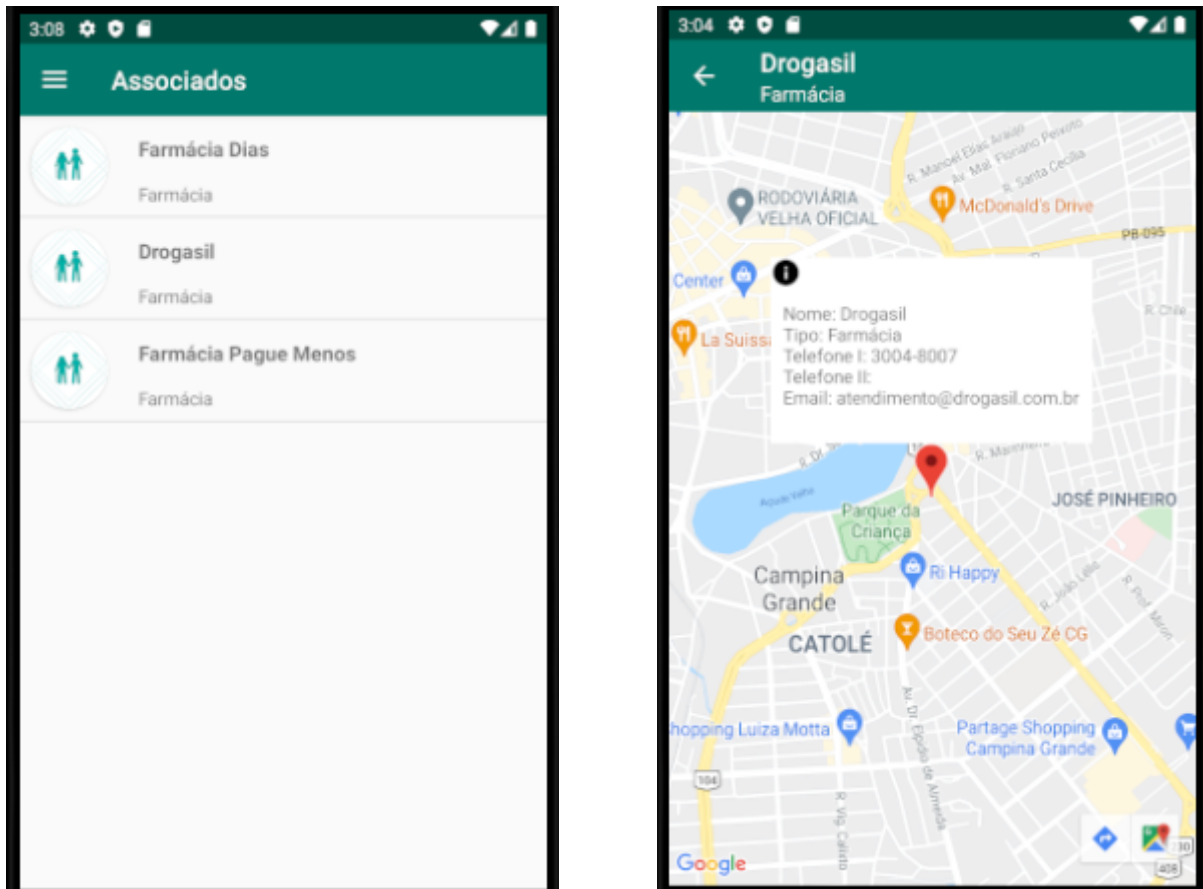
**Fonte:** Elaborada pelo próprio autor, 2020.

### Tela de associados:

Nessa tela são listados os associados da empresa, que foram cadastrados pelo administrador. As informações dos associados ficarão disponíveis para os pacientes como um tipo de serviço externo do sistema.



Figura 24 – Tela de associados

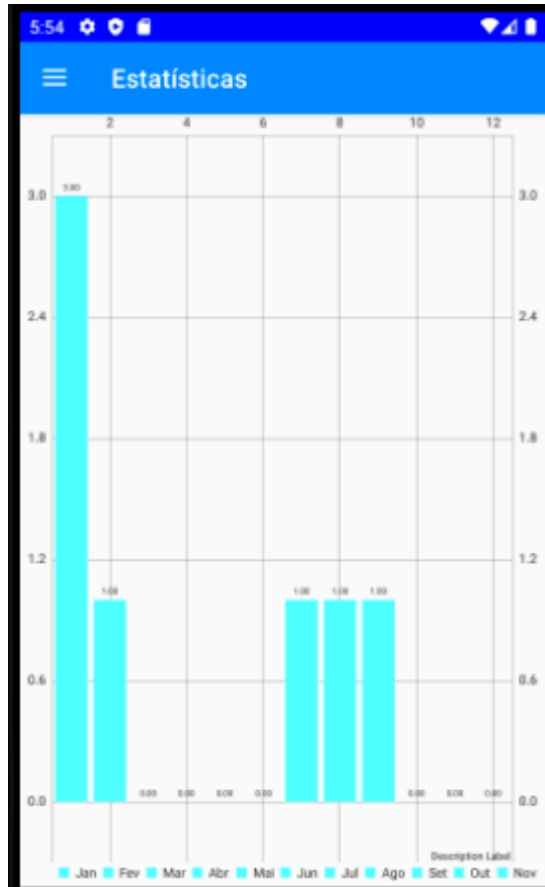


Fonte: Elaborada pelo próprio autor, 2020.

#### Tela de estatísticas do profissional:

Nessa tela o profissional de saúde pode ter um controle gráfico das suas consultas. Com a possível alta demanda de atendimentos, é importante que o profissional possa ter um controle do número de consultas que ele marcou ao longo do seu tempo de trabalho. A imagem abaixo ilustra o gráfico de estatísticas de consultas dos profissionais.

**Figura 25** – Tela de estatísticas do profissional

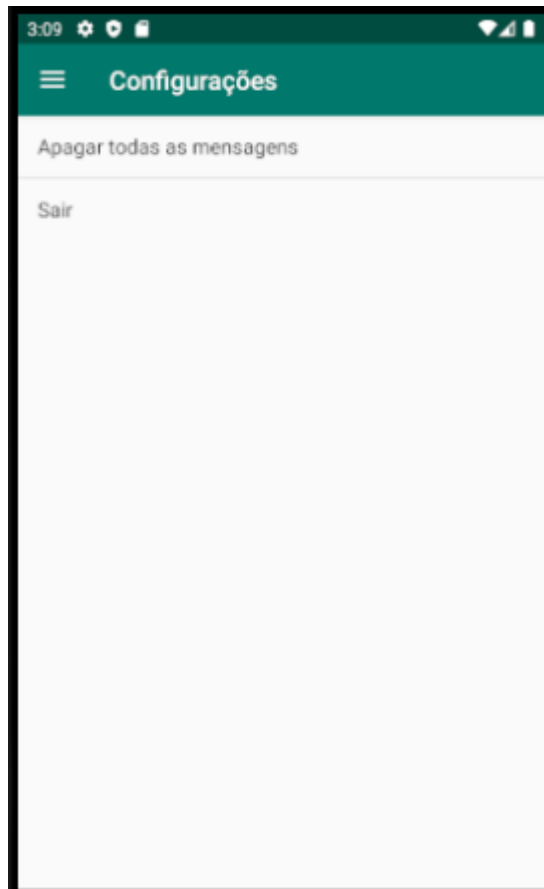


Fonte: Elaborada pelo próprio autor, 2020.

### **Tela de configurações:**

Nessa tela são apresentadas as configurações do aplicativo, tais como: Apagar todas as mensagens e sair do aplicativo. A função de apagar as mensagens é útil quando a memória do dispositivo estiver muito cheia, já que as mensagens são salvas na memória interna do dispositivo. A função de sair apaga as informações de autenticação do usuário da memória interna do dispositivo. Essas configurações estão disponíveis para os dois tipos de usuários, profissionais de saúde e pacientes. A imagem a seguir ilustra a tela de configurações.

**Figura 26** – Tela de configurações.



**Fonte:** Elaborada pelo próprio autor, 2020.

## 7 CONCLUSÃO

Devido ao alto crescimento na demanda de atendimentos na área de saúde, novas tecnologias são constantemente desenvolvidas para disponibilizar à população em geral formas mais eficazes de oferecer atendimento médico de qualidade. A solução proposta neste trabalho vai de encontro a esse problema, visto que mesmo com os avanços obtidos ao longo dos anos nesta área, o sistema de saúde ainda enfrenta vários desafios no âmbito da assistência.

O sistema foi desenvolvido usando tecnologias em constante ascensão no mercado, e buscando sempre seguir as boas práticas de desenvolvimento de software e com alto nível de modularização, de modo a tornar a solução reutilizável e facilitar a inserção de novas funcionalidades. O sistema é composto pelos módulos de paciente, profissional de saúde e administrador, tendo cada módulo uma interface de sistema específica para seu usuário alvo.

O presente trabalho teve por objetivo, mediante as situações observadas no âmbito da saúde no Brasil, oferecer à população um sistema de informação que para auxiliar na interação entre profissionais de saúde e pacientes, proporcionando um acompanhamento virtual dos atendimentos, não se fazendo mais necessário a presença do paciente na unidade de atendimento apenas para marcar uma consulta, somente ficando necessário a presença do paciente no dia marcado para o atendimento eliminando o desgaste físico ocasionado por filas de espera e perda de tempo apenas para marcar horário com o profissional de saúde. Portanto, o objetivo do trabalho, além de enfatizar a importância de se ter mais estudos nessa área para desenvolver outras formas e métodos que melhorem a assistência, facilita a marcação e acompanhamento de consultas com profissionais de saúde.

Para trabalhos futuros, espera-se que sejam feitas melhorias no sistema de acordo com a necessidade dos usuários (armazenamento dos dados em nuvem, chatbot, módulo de recomendação de especialistas, etc) , promovendo cada vez mais um maior conforto e satisfação da população.

## REFERÊNCIAS

A, Cruz; JMSV, Maldonado; CAG, Gadelha. Telemonitoramento e a dinâmica empresarial em saúde: desafios e oportunidades para o SUS. Revista de Saúde Pública, Rio de Janeiro, v. 1, n. 1, p. 1-11, dez./2019. Disponível em: [https://www.scielo.br/pdf/rsp/v54/pt\\_1518-8787-rsp-54-65.pdf](https://www.scielo.br/pdf/rsp/v54/pt_1518-8787-rsp-54-65.pdf). Acesso em: 23 nov. 2020.

ALVES, RULLY. O que são APIs e requisições HTTP ?. Disponível em: <https://medium.com/@rullyalves/o-que-s%C3%A3o-apis-e-requisi%C3%A7%C3%B5es-http-919238f48206>. Acesso em: 25 nov. 2020.

ANS-AGÊNCIA NACIONAL DE SAÚDE SUPLEMENTAR. ANS-Histórico. Disponível em: [www.ans.gov.br/aans/quem-somos/historico](http://www.ans.gov.br/aans/quem-somos/historico) . Acesso em: 20 nov. 2020.

ARIF, H. Use Express.JS as a Web Server. Disponível em: <https://medium.com/easyread/introduction-to-express-js-246191ec05f2>. Acesso em: 09 jan. 2021.

AYRES, J. R. D. C. M. Cuidado: tecnologia ou sabedoria prática?. Comunic, Saúde, Educ, São Paulo, v. 6, n. 1, p. 117-120, fev./2000. Disponível em: <https://www.scielo.br/pdf/icse/v4n6/10.pdf>. Acesso em: 22 nov. 2020.

AYRES, J. R. D. C. M. O cuidado, os modos de ser (do) humano e as práticas de saúde. Saúde e Sociedade, Brasil, v. 13, n. 3, p. 16-29, dez./2004. Disponível em: <https://www.scielo.br/pdf/sausoc/v13n3/03.pdf>. Acesso em: 24 nov. 2020.

AYRES; M., J. R. C. Cuidado e reconstrução das práticas de Saúde. Comunic., Saúde, Educ, Angra dos Reis, v. 8, n. 14, p. 73-92, fev./2004. Disponível em: <https://www.scielo.br/pdf/icse/v8n14/v8n14a04.pdf>. Acesso em: 22 nov. 2020.

BRASIL. CONSELHO NACIONAL DE SECRETÁRIOS DE SAÚDE. A gestão do SUS/Conselho nacional de secretários de saúde. – Brasília. 1a ed. CONASS, 2015.

C-CORNER. O que é por quê React.js. Disponível em: [www.c-sharpcorner.com/article/what-and-why-reactjs](http://www.c-sharpcorner.com/article/what-and-why-reactjs). Acesso em: 24 nov. 2020.

CEBALLOS, A. G. D. C. D. Modelos conceituais de saúde, determinação social do processo saúde e doença, promoção da saúde. 2. ed. Recife: UNA-SUS UFPE, 2015. p. 1-22.

COELHO, M. O.; JORGE, M. S. B. Tecnologia das relações como dispositivo do atendimento humanizado na atenção básica à saúde na perspectiva do acesso, do acolhimento e do vínculo. Revista ciência e saúde coletiva. Ceará, 2009.

COPES, FLAVIO. The React Handbook. Disponível em: <https://www.freecodecamp.org/news/the-react-handbook-b71c27b0a795> . Acesso em: 24 nov. 2020

DEVELOPER ANDROID. Arquitetura da plataforma. Disponível em: <https://developer.android.com/guide/platform> . Acesso em: 23 nov. 2020.

DEVELOPER ANDROID. Conheça o Android Studio. Disponível em: <https://developer.android.com/studio/intro> . Acesso em: 23 nov. 2020.

FAUSTINO, G. K. S; CALAZANS, H. K. N. S; LIMA, W. D. Android e a influência do Sistema Operacional Linux v. 8, n. 1, p. 1-111 jan./2017. Disponível em: <<http://revista.faculdadeprojecao.edu.br/index.php/Projecao4/article/view/829/728>>. Acesso em: 04 jan 2021

FILHO, L. T. P. Iniciativa privada e saúde. Scielo, São Paulo, v. 13, n. 1, p. 109-116, abr./1999

FILHO, Claudio Bertolli. História em movimento: História da saúde pública no Brasil. 4. ed. São Paulo: Ática, 2010. p. 1-71.

FREECODECAMP. Manual do React. Disponível em: [www.freecodecamp.org/news/the-react-handbook-b71c27b0a795](http://www.freecodecamp.org/news/the-react-handbook-b71c27b0a795). Acesso em: 24 nov. 2020.

FREIRE, L. L. R. INFORME INDÚSTRIA E SERVIÇOS ETENE : SISTEMA BRASILEIRO DE SAÚDE . ESCRITÓRIO TÉCNICO DE ESTUDOS ECONÔMICOS DO NORDESTE ETENE, BRASIL, v. 3, n. 1, p. 1-17, dez./2009. Disponível em: [https://www.bnb.gov.br/documents/88765/89729/iis\\_sistema\\_brasileiro\\_saude.pdf/be5b3654-7d1e-464a-9745-c243f5d12b47](https://www.bnb.gov.br/documents/88765/89729/iis_sistema_brasileiro_saude.pdf/be5b3654-7d1e-464a-9745-c243f5d12b47). Acesso em: 23 nov. 2020.

GARGENTA, Marko. Building Applications for the Android Market: Learning Android. 1. ed. Califórnia: O'Reilly, 2011. p. 1-239.

GONÇALES, R. L. G. Arquetetando uma API RESTful. Disponível em: <https://medium.com/@ramonrune/arquetetando-uma-api-restful-8ffcf892586a>. Acesso em: 25 nov. 2020.

GUEDES; T., Gilleanes. UML 2: Uma abordagem pratica. Câmara Brasileira do Livro, São Paulo, v. 2, n. 3, p. 17-44, mai./2009. Disponível em: [https://www.academia.edu/6202541/UML\\_2\\_uma\\_abordagem\\_pr%C3%A1tica](https://www.academia.edu/6202541/UML_2_uma_abordagem_pr%C3%A1tica). Acesso em: 25 nov. 2020.

HIDOCTOR. Disponível em: <https://blog.hidoctor.com.br/p/gestao-do-consultorio/1287528/importancia+da+boa+comunicacao+verbal+nos+consultorios+medicos.htm>. Acesso em: 29 dez. 2020.

JORGE, M. O. C. M. S. B. Tecnologia das relações como dispositivo do atendimento humanizado na atenção básica à saúde na perspectiva do acesso, do acolhimento e do vínculo. Ciência & Saúde Coletiva, Ceará, v. 14, n. 1, p. 1523-1531, mar./2008. Disponível em: <https://www.scielo.br/pdf/csc/v14s1/a26v14s1.pdf>. Acesso em: 25 nov. 2020.

KHACHATRYAN, G. Understanding Socket.io Disponível em : <https://grigorkh.medium.com/understanding-socket-io-23ad6d26d003> Acesso em: 09 jan. 2021.

KEMP, C.; GYGER, B. Professional heroku programming. 1a ed. Editora Wrox, 2013.

LIMA. H. R. C. Agendamento de consultas pela internet: proposta de um aplicativo de conectividade para a saúde. Fundação Getúlio Vargas – Escola de administração de empresas – São Paulo, 2002.

LIU, Qigang; SUN, Xiangyang. Research of Web Real-Time Communication Based on Web Socket . Scientific Research, Shanghai, v. 1, n. 1, p. 797-801, dez./2012. Disponível em: [https://www.scirp.org/pdf/IJCNS20121200001\\_70890007.pdf](https://www.scirp.org/pdf/IJCNS20121200001_70890007.pdf). Acesso em: 24 nov. 2020.

LORENZETTI1, J. et al. TECNOLOGIA, INOVAÇÃO TECNOLÓGICA E SAÚDE: UMA REFLEXÃO NECESSÁRIA. Texto Contexto Enferm, Florianópolis, v. 21, n. 2, p. 432-439, jun./2012. Disponível em: <https://www.scielo.br/pdf/tce/v21n2/a23v21n2.pdf>. Acesso em: 25 nov. 2020.

SANTOS, GUILHERME. O que é, por que usar e primeiros passos. Disponível em: <https://medium.com/thdesenvolvedores/node-js-o-que-%C3%A9-por-que-usar-e-primeiros-passos-1118f771b889>. Acesso em: 23 nov. 2020.

SENNÁ, S. B. B. et. al. GESTÃO DA FILA DE ESPERA PARA CIRURGIAS ELETIVAS EM HOSPITAIS DO SISTEMA ÚNICO DE SAÚDE. v. 30, n. 2, p. 79-82, mar./2020. Disponível em : [https://www.mastereditora.com.br/periodico/20200408\\_122550.pdf](https://www.mastereditora.com.br/periodico/20200408_122550.pdf)

SKÓLSKI, PAWEL. Aplicativo de página única vs. aplicativo de várias páginas. Disponível em: <https://medium.com/@NeotericEU/single-page-application-vs-multiple-page-application-2591588efe58>. Acesso em: 23 nov. 2020.

MARDAN, Azat. Practical Node.js.: Aplicativos em tempo real com WebSocket, Socket.IO e DerbyJS. 1. ed. Califórnia: Apress, 2018. p. 307-308.

Ministério da Saúde. Disponível em: [www.gov.br/saude/pt-br](http://www.gov.br/saude/pt-br) . Acesso em: 22 nov. 2020.

MINISTÉRIO DA SAÚDE. Protocolos Clínicos e Diretrizes Terapêuticas - PCDT. Disponível em: [antigo.saude.gov.br/protocolos-e-diretrizes](http://antigo.saude.gov.br/protocolos-e-diretrizes) . Acesso em: 22 nov. 2020.

NODEJS. Introdução ao Node.Js. Disponível em: <https://nodejs.dev/learn/introduction-to-nodejs> . Acesso em: 23 nov. 2020.

NODEJS. Node.Js - O que é, por que usar e primeiros passos. Disponível em: <https://medium.com/thdesenvolvedores/node-js-o-que-%C3%A9-por-que-usar-e-primeiros-passos-1118f771b889>. Acesso em: 23 nov. 2020.

NODEJS. Sobre Node.js. Disponível em: <https://nodejs.org/pt-br/about/> . Acesso em: 23 nov. 2020.

NODEJS. The Node.js Event Loop, Timers, and process.nextTick. Disponível em: <https://nodejs.org/pt-br/docs/guides/event-loop-timers-and-nexttick/> . Acesso em: 23 nov. 2020.

PAIM, J. S. Sistema único de saúde (SUS) aos 30 anos. Revista Ciência e saúde coletiva. Bahia, 2018.

PANDIT, NITIN. What and Why React.js Disponível em: [www.c-sharpcorner.com/article/what-and-why-reactjs](http://www.c-sharpcorner.com/article/what-and-why-reactjs) . Acesso em: 21 nov. 2020.

PENSESUS. PúblicoXPrivado. Disponível em: <https://pensesus.fiocruz.br/publico-x-privado>. Acesso em: 20 nov. 2020.



PORTAL EDUCAÇÃO. Determinantes sociais da saúde. Disponível em: [siteantigo.portaleducacao.com.br/conteudo/artigos/nutricao/determinantes-sociais-da-saude/38650](http://siteantigo.portaleducacao.com.br/conteudo/artigos/nutricao/determinantes-sociais-da-saude/38650) . Acesso em: 25 nov. 2020.

REACT. Glossário de Termos React. Disponível em: <https://pt-br.reactjs.org/docs/glossary.html> . Acesso em: 23 nov. 2020.

REACT. O que é React?. Disponível em: <https://pt-br.reactjs.org/tutorial/tutorial.html#what-is-react> . Acesso em: 23 nov. 2020.

SÁ NETO, J. A.; RODRIGUES, B. M. R. D. Tecnologia como fundamento do cuidar em neonatologia. Texto e Cont. Enferm. [Internet], Florianópolis, v. 19, n. 2, p. 372-377, 2010. Disponível em: <[http://www.scielo.br/scielo.php?pid=S0104-07072010000200020&script=sci\\_arttext](http://www.scielo.br/scielo.php?pid=S0104-07072010000200020&script=sci_arttext)>. Acesso em: 04 jan. 2021.

SILVA, R. C.; FERREIRA, M. A. A tecnologia em saúde: uma perspectiva psicossociológica. Esc. Anna Nery Rev. Enferm.[Internet], Rio de Janeiro, v. 13, n. 1, p. 169-173, 2009. Disponível em:<<http://www.scielo.br/pdf/ean/v13n1/v13n1a23.pdf>>. Acesso em: 04 jan. 2021.

SOURCE ANDROID. Arquitetura Android. Disponível em: <https://source/android.com/devices/architecture> . Acesso em: 23 nov. 2020.

SRINIVASAN, Lakshminarasimhan; SCHARNAGL, Julian; SCHILLING, Klaus. Analysis of WebSockets as the New Age Protocol for Remote Robot Tele-operation. Ifac, Korea, v. 1, n. 1, p. 83-88, nov./2013. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1474667015343688>. Acesso em: 24 nov. 2020.

SUSEP. Susep Apresentação. Disponível em: [www.susep.gov.br/menu/a-susep/apresentacao](http://www.susep.gov.br/menu/a-susep/apresentacao). Acesso em: 20 nov. 2020.

TANENBAUM, A. S. Sistemas Operacionais Modernos. 3. ed. São Paulo: Pearson Prentice Hall, 2009

TEIXEIRA, RICARDO RODRIGUES. Humanização e atenção primária a saúde: Humanization and Primary Health Care. Scielo, São Paulo, v.10 , n.1, p. 585-597, jul./2005

THEMUDO, Vasco Morais. Implementação de um servidor de negociação em bolsa baseado em WebSocket. U.Porto, Porto, v. 1, n. 1, p. 1-63, jan./2014. Disponível em: [repositorio-aberto.up.pt/bitstream/10216/75373/2/31554.pdf](https://repositorio-aberto.up.pt/bitstream/10216/75373/2/31554.pdf). Acesso em: 24 nov. 2020.

WEGNER, M. F. A. L. W. Eventos adversos no cuidado da criança: concepções de familiar/cuidador na atenção básica: Adverse events in the health care for children: conceptions of family/caregiver in the primary care. Revista Ciência & Saúde, Porto Alegre, v. 6, n. 3, p. 190-196, dez./2013. Disponível em: [https://www.researchgate.net/profile/Wiliam\\_Wegner/publication/307548575\\_Eventos\\_adversos\\_no\\_cuidado\\_da\\_crianca\\_concepcoes\\_de\\_familiarcuidador\\_na\\_atencao\\_basica/links/5ccf3db0299bf14d957adcdf/Eventos-adversos-no-cuidado-da-crianca-concepcoes-de-familiar-cuidador-na-atencao-basica.pdf](https://www.researchgate.net/profile/Wiliam_Wegner/publication/307548575_Eventos_adversos_no_cuidado_da_crianca_concepcoes_de_familiarcuidador_na_atencao_basica/links/5ccf3db0299bf14d957adcdf/Eventos-adversos-no-cuidado-da-crianca-concepcoes-de-familiar-cuidador-na-atencao-basica.pdf). Acesso em: 22 nov. 2020.

ZHAO; MAN, Shu; AL, Et. Uma solução de aplicativo da Web em tempo real baseada em Node.js e WebSocket.: Pesquisa de materiais avançados. Trans Tech Publications, Ltd., Suíça, v. 817, n. 1, p. 1111-1115, set./2013. Disponível em: <https://www.scientific.net/AMR.816-817.1111>. Acesso em: 24 nov. 2020.

## **APÊNDICE A – REQUISITOS FUNCIONAIS DO SISTEMA**

### **RF001 – Tela de acesso do administrador**

Permite a autenticação do usuário administrador no site.

### **RF002 – Tela de cadastro do administrador**

Permite cadastrar um novo usuário administrador do sistema.

### **RF003 – Tela de informações dos usuários**

Permite que o administrador acesse as informações dos usuários do sistema.

### **RF004 – Tela de cadastro de profissionais**

Permite ao usuário administrador cadastrar profissionais de saúde no sistema.

### **RF005 – Tela de cadastro de associados**

Permite ao usuário administrador cadastrar novos associados à empresa.

### **RF006 – Tela de cadastro de profissões**

Permite ao usuário administrador cadastrar novas profissões no sistema.

### **RF007 – Tela de login do paciente**

Permite ao usuário paciente se autenticar no aplicativo.

### **RF008 – Tela de cadastro do paciente**

Permite ao usuário paciente se cadastrar no sistema.

### **RF009 – Tela do menu de navegação do aplicativo do paciente**

Permite ao usuário paciente navegar através das principais funcionalidades do aplicativo.

### **RF010 – Tela de lista de profissionais**

Permite ao usuário paciente obter uma lista com os profissionais de saúde cadastrados no sistema.

### **RF011 – Tela de conversa**

Permite ao usuário paciente conversar em tempo real com os profissionais de saúde.

### **RF012 – Tela de visualização de consultas do paciente**

Permite ao usuário paciente visualizar todas as consultas relacionadas a ele.

### **RF013 – Tela de associados**

Permite ao usuário paciente visualizar uma lista com todos os associados cadastrados na empresa.

### **RF014 – Tela de informações dos associados**

Permite ao usuário paciente visualizar informações de localização e contato dos associados cadastrados na empresa.

**RF015 – Tela de configurações do aplicativo**

Permite aos usuários paciente e profissional acessar configurações do aplicativo.

**RF016 – Tela de login do profissional**

Permite ao usuário profissional se autenticar no aplicativo.

**RF017 – Tela do menu de navegação do aplicativo dos profissionais**

Permite ao usuário profissional navegar através das principais funcionalidades do aplicativo.

**RF018 – Tela de lista dos pacientes**

Permite ao usuário profissional obter uma lista de todos os pacientes cadastrados no sistema.

**RF019 – Tela de visualização de consultas do profissional**

Permite ao usuário profissional visualizar todas as consultas marcadas por ele.

**RF020 – Tela de criação de consultas**

Permite ao usuário profissional cadastrar uma nova consulta no sistema.

**RF021 – Tela de número de consultas do profissional**

Permite ao usuário profissional visualizar um gráfico com estatísticas das consultas marcadas.

## **APÊNDICE B – REQUISITOS NÃO FUNCIONAIS DO SISTEMA**

### **RN001 – Fácil usabilidade**

O sistema deve ser de fácil manuseio, permitindo que qualquer pessoa possa operá-lo sem treinamento.

### **RN002 – Desenvolvimento back-end**

O sistema será desenvolvido utilizando a plataforma NodeJS.

### **RN003 – Desenvolvimento front-end**

O sistema terá dois aplicativos móveis, para paciente e profissional, e um site para gerenciamento das informações por meio de um usuário administrador.