



UFOP

Universidade Federal
de Ouro Preto

**Universidade Federal de Ouro Preto
Instituto de Ciências Exatas e Aplicadas
Departamento de Computação e Sistemas**

Desenvolvimento de um aplicativo multiplataforma para gestão de vacinação

Guilherme Ferreira Faioli Lima

TRABALHO DE CONCLUSÃO DE CURSO

**ORIENTAÇÃO:
Euler Horta Marinho**

**Abril, 2021
João Monlevade–MG**

Guilherme Ferreira Faioli Lima

**Desenvolvimento de um aplicativo
multiplataforma para gestão de vacinação**

Orientador: Euler Horta Marinho

Monografia apresentada ao curso de Sistemas de Informação do Instituto de Ciências Exatas e Aplicadas, da Universidade Federal de Ouro Preto, como requisito parcial para aprovação na Disciplina “Trabalho de Conclusão de Curso II”.

Universidade Federal de Ouro Preto

João Monlevade

Abril de 2021

SISBIN - SISTEMA DE BIBLIOTECAS E INFORMAÇÃO

L732d Lima, Guilherme Ferreira Faioli .
Desenvolvimento de um aplicativo multiplataforma para gestão de
vacinação. [manuscrito] / Guilherme Ferreira Faioli Lima. - 2021.
68 f.: il.: color., tab..

Orientador: Prof. Me. Euler Horta Marinho.
Monografia (Bacharelado). Universidade Federal de Ouro Preto.
Instituto de Ciências Exatas e Aplicadas. Graduação em Sistemas de
Informação .

1. Aplicativos móveis. 2. Engenharia de software. 3. Sistemas de
recuperação da informação - Saúde pública. 4. Software de aplicação -
Desenvolvimento. 5. Vacinação - administração. I. Marinho, Euler Horta.
II. Universidade Federal de Ouro Preto. III. Título.

CDU 004.41

Bibliotecário(a) Responsável: Flavia Reis - CRB6-2431



FOLHA DE APROVAÇÃO

Guilherme Ferreira Faioli Lima

Desenvolvimento de um aplicativo multiplataforma para gestão de vacinação

Monografia apresentada ao Curso de Sistemas de Informação da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação

Aprovada em 12 de abril de 2021

Membros da banca

Mestre - Euler Horta Marinho - Orientador (Universidade Federal de Ouro Preto)
Doutor - Diego Zuquim Guimarães Garcia - (Universidade Federal de Ouro Preto)
Mestre - Daniela Rodrigues Dias - (Doutoranda em Educação - Universidade Federal de Ouro Preto)

Euler Horta Marinho, orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 26/04/2021



Documento assinado eletronicamente por **Euler Horta Marinho, PROFESSOR DE MAGISTERIO SUPERIOR**, em 26/04/2021, às 14:11, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **0163499** e o código CRC **2EBFB2A4**.

Este trabalho é dedicado a minha família e amigos, que me ajudaram e apoiaram até aqui.

Agradecimentos

Agradeço aos meus pais, Juliana e Marcio pela confiança no meu progresso e apoio.

Agradeço ao meu orientador Euler por sua dedicação e auxílio, além da disponibilidade de sanar todas as minhas dúvidas.

Agradeço também a universidade Federal de Ouro Preto e todo seu corpo docente.

“Que os vossos esforços desafiem as impossibilidades, lembrai-vos de que as grandes coisas do homem foram conquistadas do que parecia impossível.”

— Charles Chaplin

Resumo

Este trabalho tem como objetivo o desenvolvimento de um aplicativo multiplataforma para a gestão de vacinação. Ele apresenta como objetivos específicos, o levantamento de requisitos com base em pesquisa e análise de aplicativos similares, a elaboração do diagrama de caso de uso, a modelagem do banco de dados, o projeto de *Mockups* e a realização de testes. O trabalho apresenta em sua justificativa sua motivação, os ganhos ao se utilizar um sistema de informação na área da saúde e as vantagens do desenvolvimento multiplataforma. Ele apresenta como contribuição a promoção da saúde e conscientização sobre a importância da vacinação. O aplicativo será responsável por armazenar informações de vacinas e informações essenciais dos usuários, com o objetivo de obter uma gestão das vacinas que o usuário já se vacinou ou ainda precisa ser vacinado. O aplicativo também informará quando esse usuário deve tomar novas doses, com o objetivo de auxiliar essas pessoas a estarem sempre imunizadas em relação às doenças com cobertura de vacinas pelo Sistema Único de Saúde (SUS). Como resultado, foi implementado um aplicativo multiplataforma para Android e iOS, capaz de auxiliar os usuários na gestão de vacinação.

Palavras-chave: gestão de vacinação. desenvolvimento multiplataforma. desenvolvimento móvel. Engenharia de Software.

Abstract

This work consists in the development of a multiplatform application for the vaccination management. As specific objectives, the discovery of requirements based on research and analysis of similar applications, the design of the use case diagram, the modeling of the database, the designing of Mockups, and the test planning and execution. The work presents as justificative, its motivation, the gains when using an information system in the area of health, and the advantages of the multiplatform development. It contributes to health promotion and awareness of the importance of vaccination. The application will be responsible for storing vaccine information and essential user information, to get management of the vaccines that the user has already vaccinated or still needs to be vaccinated. The application will also inform when the user should take new doses, to assist people to always be immunized about diseases covered by vaccines in the Unified Health System (SUS). As a result, a multiplatform application for Android and iOS was implemented. It is able of assisting its users in the management of the vaccination.

Keywords: vaccination management. multiplatform development. mobile development. Software Engineering.

Lista de ilustrações

Figura 1 – Calendário Nacional de vacinação.	20
Figura 2 – Diagrama UML.	26
Figura 3 – Mockup.	32
Figura 4 – Diagrama entidade-relacionamento.	33
Figura 5 – Regiões e servidores de armazenamento em nuvem	35
Figura 6 – MongoDB Atlas	36
Figura 7 – Database	36
Figura 8 – Diagrama da View no Android e iOS	38
Figura 9 – Inicialização do Expo no terminal	39
Figura 10 – Inicialização do Expo através do Devtools no navegador	41
Figura 11 – Últimas <i>builds</i> criadas	42
Figura 12 – Detalhes da build	43
Figura 13 – Fila para que o aplicativo seja criado	43
Figura 14 – Tela de Login	45
Figura 15 – Tela de Cadastro de novo usuário	46
Figura 16 – Página inicial	47
Figura 17 – Registro de vacina	48
Figura 18 – Tela de vacina	49
Figura 19 – Tela de próximas doses	50
Figura 20 – Principal diferença entre a versão de Android e iOS	51
Figura 21 – Comparação do login	52
Figura 22 – Comparação do registro de novos usuários	52
Figura 23 – Comparação da página principal	53
Figura 24 – Comparação da tela de vacina	53
Figura 25 – Comparação da tela próximas doses	54
Figura 26 – Teste Robo Lenovo k520, API 26	57
Figura 27 – Teste Robo Pixel 2, API 28	57
Figura 28 – Gráfico de rastreamento	58
Figura 29 – Desempenho ao longo do tempo	59
Figura 30 – Desempenho do processador	59
Figura 31 – Desempenho gráfico	59
Figura 32 – Desempenho de uso de memória	60
Figura 33 – Uso de banda	60
Figura 34 – Problemas com o teste	61
Figura 35 – Gráfico de rastreamento, Parte 1	67
Figura 36 – Gráfico de rastreamento, Parte 2	68

Lista de tabelas

Tabela 1 – Componentes principais	38
Tabela 2 – Comparativo com aplicativos similares	55

Lista de abreviaturas e siglas

API	Interface de programação de aplicações
APK	<i>Android Package</i>
iOS	<i>iPhone Operating System</i>
SO	Sistema Operacional
CSS	<i>Cascading Style Sheets</i>
HTML	<i>Hypertext Markup Language</i>
XML	<i>eXtensible Markup Language</i>
JSX	<i>JavaScript XML</i>
GPS	<i>Global Positioning System</i>
SMS	<i>Short Message Service</i>
UI	<i>User Interface</i>
IDE	Ambiente de desenvolvimento integrado
SI	Sistema de informação
UML	<i>Unified Modeling Language</i>
BD	Banco de Dados
JSON	<i>JavaScript Object Notation</i>
CPU	Unidade Central de Processamento
SDK	<i>Software Development Kit</i>

Sumário

1	INTRODUÇÃO	14
1.1	Justificativa	15
1.2	Objetivos	16
1.2.1	Objetivo Geral	16
1.2.2	Objetivos Específicos	17
1.3	Organização do trabalho	17
2	REVISÃO BIBLIOGRÁFICA	18
2.1	Sistemas de Informação	18
2.2	Programa Nacional de Imunizações	19
2.3	Desenvolvimento de Aplicativos Multiplataforma	21
3	DESENVOLVIMENTO	23
3.1	Aplicativos Semelhantes	23
3.1.1	Conecte SUS	23
3.1.2	Calendário de vacinação	23
3.1.3	Carteirinha de Criança	24
3.2	Descoberta de requisitos	24
3.2.1	Requisitos funcionais	24
3.2.2	Requisitos não funcionais	25
3.3	Diagrama de caso de uso	25
3.3.1	Casos de Uso relacionado ao ator Usuário	26
3.3.1.1	Caso de Uso 1	26
3.3.1.2	Caso de Uso 2	27
3.3.1.3	Caso de Uso 3	27
3.3.1.4	Caso de Uso 4	28
3.3.1.5	Caso de Uso 5	28
3.3.1.6	Caso de Uso 6	29
3.3.1.7	Caso de Uso 7	29
3.3.2	Casos de Uso relacionado ao ator Administrador	30
3.3.2.1	Caso de Uso 1	30
3.3.2.2	Caso de Uso 2	30
3.3.2.3	Caso de Uso 3	31
3.4	Protótipos de telas	31
3.5	Banco de dados	32
3.5.1	MongoDB	33

3.6	Framework para o desenvolvimento multiplataforma	36
3.6.1	React Native	37
3.7	Expo	39
4	RESULTADOS	44
4.1	O aplicativo	44
4.1.1	Login	44
4.1.2	Cadastro de usuário	45
4.1.3	Página inicial	46
4.1.4	Registro de vacina	47
4.1.5	Tela de vacina	48
4.1.6	Tela de próximas doses	49
4.2	Comparação entre as versões de Android e iOS	50
4.3	Comparação entre o aplicativo desenvolvido e os aplicativos similares	54
4.4	Testes	56
5	CONSIDERAÇÕES FINAIS	62
5.1	Objetivos alcançados	63
5.2	Trabalhos Futuros	63
	REFERÊNCIAS	64
	APÊNDICES	66
	APÊNDICE A – MATERIAIS ELABORADOS PELO AUTOR	67
A.1	Gráfico de rastreamento	67

1 Introdução

Esse trabalho utiliza na sua concepção o desenvolvimento multiplataforma para elaborar um aplicativo para gestão de vacinação. Desenvolvimento multiplataforma envolve a estratégia em que um único código poder ser compilado para mais de um sistema operacional. Segundo El-Kassas e outros (EL-KASSAS et al., 2017), o conceito principal de uma solução multiplataforma é desenvolver um aplicativo uma vez e executá-lo em qualquer lugar.

O desenvolvimento multiplataforma traz uma grande vantagem na economia de tempo e recursos, uma vez que não é preciso desenvolver mais de uma vez um mesmo aplicativo. “[...] toda vez que um desenvolvedor deseja oferecer um aplicativo em mais de uma plataforma, é necessário realizar novamente uma parte significativa do processo de desenvolvimento de software” (CORRAL; JANES; REMENCIUS, 2012).

Um dos grandes motivos da popularização do desenvolvimento multiplataforma é a necessidade de se ter uma mesma aplicação sendo executada em diversos sistemas operacionais. De acordo com El-Kassas e outros (EL-KASSAS et al., 2017), o número de dispositivos móveis aumenta todo ano, e com isso há uma demanda maior por aplicativos que atendam as necessidades dos usuários.

A ideia principal do sistema que será desenvolvido é que ele auxilie o usuário a manter suas vacinas em dia, através do cadastro das vacinas, datas e de suas outras informações. Além disso, o objetivo do aplicativo é lembrar o usuário das datas de vacinas próximas, para que ele mantenha suas doses e vacinas em dia. Isso será realizado através de notificações e avisos que serão exibidos no aplicativo. Também, o usuário poderá verificar todas as vacinas que já foram aplicadas, do mesmo modo que ocorre quando se utiliza um cartão de vacinação. De acordo com *The Community Preventive Services Task Force* (FORCE et al., 2015), a utilização de um sistema de lembretes e *feedbacks* aos usuários sobre imunização tem uma forte ligação com o número de vacinações. Sobre o funcionamento das vacinas e a capacidade do Brasil em produzir vacinas:

A vacina serve para defender dos vírus e bactérias que provocam doenças. Hoje, o Brasil conta com técnicas modernas para produzi-las em sete laboratórios públicos, atendendo a todo o processo de qualidade de produção exigido pela Agência Nacional de Vigilância Sanitária (Anvisa). As vacinas podem ser produzidas a partir de organismos enfraquecidos, mortos ou alguns derivados, podendo ser aplicadas por meio de injeção ou por via oral (BRASIL, 2017).

O que se espera na conclusão do trabalho é ter um aplicativo funcional tanto para o Android quanto para o iOS, que permite a seus usuários o registro de suas vacinas de forma similar a um cartão de vacinas virtual, e com informações sobre quais as datas das próximas vacinas ou doses. Também, serão exibidas notificações sobre as datas em que

as próximas vacinas devem ser aplicadas. O código fonte do aplicativo desenvolvido está disponível em repositório do Github ¹.

1.1 Justificativa

A ideia da realização desse trabalho veio de um problema que o autor desta monografia teve pelo fato de não manter uma organização adequada do cartão de vacinação, onde ele se deparou com o problema de ter mais de um cartão de vacina e acabar perdendo alguns desses cartões, e com isso tendo a impossibilidade de fazer a gestão de suas vacinações. Então ele teve a ideia de criar um aplicativo que tivesse a capacidade de funcionar como um cartão de vacinação virtual, e que pudesse auxiliar outras pessoas com um problema similar.

Um dos principais focos deste trabalho é a promoção da saúde, com o desenvolvimento de uma ferramenta para auxiliar seus usuários a fazerem a gestão de suas vacinas e manter esses registros no aplicativo, e com isso se manter imunizado das doenças que possuem vacinas fornecidas pelo SUS. Segundo Groom ([GROOM et al., 2015](#)), sistemas de informação voltados a imunização podem alavancar o número de vacinações através de notificações e lembretes. Ainda segundo Groom, nos Estados Unidos, de 2001 a 2003, 8 sistemas de imunização de oito regiões diferentes (Arizona, Distrito de Columbia, Michigan, Cidade de Nova York, Oklahoma, Oregon, San Antonio e Utah) foram estudados e os dados obtidos foram de que 85% de jovens menores que 19 anos tinham ao menos dois registros de vacinas no sistema.

Além de manter um cartão de vacinas em formato de papel, ter também um cartão de vacinas virtual seria muito útil. Pelo fato deste documento estar disponível em ambientes virtuais, os usuários não iriam enfrentar problemas como a perda do cartão ou o fato de ter vários cartões, e com isso ter uma dificuldade em manter uma organização de suas vacinações. De acordo com ([ARELLANO, 2004](#)), a utilização do armazenamento virtual possibilita uma maior preservação de documentos e uma maior disponibilidade dos mesmos, uma vez que esses documentos ficam disponíveis para acesso através da Internet.

É importante destacar que o trabalho também irá abordar sobre a conscientização da importância da vacinação, uma vez que vem ocorrendo uma queda em algumas taxas de vacinação. Segundo a OMS e a Unicef ([ONU, 2020](#)), em 2020 houve uma queda de até metade da taxa de vacinação de crianças, em vacinas contra poliomielite e sarampo.

Ao se pensar em implementar sistemas de informação, primeiro é preciso saber como seria o acesso a essas tecnologias no Brasil, e a pesquisa abaixo informa a capacidade de se usar sistemas de informação voltados à saúde no Brasil:

A disponibilização de computadores e acesso à Internet em todos os

¹ Código fonte disponível em: <<https://github.com/GuilhermeFaioli/TCC>>

estabelecimentos de saúde brasileiros é condição necessária para uma efetiva ampliação da saúde digital no país. A pesquisa TIC Saúde vem apontando, desde 2013, um aumento da presença de computadores e Internet nos estabelecimentos de saúde brasileiros, sendo que entre estabelecimentos com internação e os que prestam serviços de apoio à diagnose e terapia essa presença atingiu a universalização em 2019. Já entre os estabelecimentos sem internação, 95% tinham computadores e 91% tinham acesso à Internet. Esse resultado indica um déficit: cerca de 4,4 mil estabelecimentos sem computadores e 8,2 mil sem acesso à Internet (CETIC, 2020).

Como dito na pesquisa do parágrafo anterior, existe um número muito grande de estabelecimentos de saúde que possuem computadores e acesso a internet, o que torna possível um maior uso de sistemas de informações voltados à área da saúde.

Outro ponto abordado neste trabalho é a existência de diversos sistemas operacionais, dando maior foco a sistemas móveis. Então sua concepção é feita através do desenvolvimento multiplataforma, que garante a seu desenvolvedor uma significativa economia de tempo e recursos sem que o aplicativo tenha uma perda significativa de desempenho. Assim, durante esse desenvolvimento não ocorrerá retrabalho, uma vez que um único código pode ser compilado para diversos sistemas operacionais, também possibilitando que mais pessoas tenham acesso a esse aplicativo. Para termos uma ideia do número de usuários de smartphones e os sistemas operacionais utilizados por eles podemos ver os dados disponíveis abaixo²:

De acordo com um questionário feito nos Estados Unidos em Outubro, 2011, 56% dos usuários de *smartphones* têm sistema operacional Android, enquanto 28% usam iOS; no resto do mundo, a participação de mercado também é distribuída de maneira uniforme. Por outro lado, em janeiro de 2012, os mercados de distribuição de aplicativos abrangeram uma gama de mais de 500.000 aplicativos na iOS *App Store* e mais de 400.000 no Android *Play Store*, envolvendo 18.000.000.000 de downloads da iOS *App Store* e 10.000.000.000 da *Play Store* Android. Esse panorama mostra que desenvolver para apenas um sistema impediria que uma aplicação atingisse um número considerável de clientes (CORRAL; JANES; REMENCIUS, 2012).

1.2 Objetivos

1.2.1 Objetivo Geral

O objetivo desse trabalho é desenvolver uma ferramenta para auxílio na gestão e manutenção das vacinas de cada usuário, com capacidade de atender usuários de múltiplas plataformas móveis e reduzir o esforço de desenvolvimento.

² De acordo com a GSMA (GSMA, 2019) em 2018, o número de pessoas que usava algum tipo de aparelho celular era 5.1 bilhões de pessoas, sendo que dos usuários de *smartphones* aproximadamente 85% dos usuários usam Android, enquanto 15% usam iOS

1.2.2 Objetivos Específicos

- Levantar requisitos com base em pesquisa e análise de aplicativos similares.
- Projetar diagrama de caso de uso utilizando UML.
- Modelar o banco de dados.
- Projetar *Mockups*.
- Realizar testes e validações.

1.3 Organização do trabalho

Esse trabalho está estruturado de forma a possuir 5 capítulos. O Capítulo 1 diz respeito a introdução do trabalho. O Capítulo 2 entra em detalhes sobre a revisão bibliográfica. O Capítulo 3 fala a respeito do desenvolvimento, onde são abordados os tópicos sobre aplicativos semelhantes, levantamento e Análise de Requisitos, casos de uso, protótipos de telas, banco de dados, *frameworks* e *React Native*. O Capítulo 4 apresenta os resultados e testes e, por fim, o Capítulo 5 contém as considerações finais e trabalhos a serem realizados no futuro.

2 Revisão bibliográfica

2.1 Sistemas de Informação

De acordo com Kenneth e Laudon ([KENNETH; LAUDON, 2011](#)), um sistema de informação é um conjunto de componentes interligados que processam dados e os utilizam na tomada de decisão. Ainda segundo Kenneth e Laudon, a importância deste sistema para uma organização é seu auxílio na tomada de decisões, de modo que os dados de um sistema sejam processados e se tornem totalmente úteis a uma organização. Kenneth e Laudon ([KENNETH; LAUDON, 2011](#)) também classificam os sistemas de informação em seis tipos, sendo eles: Sistemas de Apoio ao Executivo (SAE), Sistemas de Apoio à Decisão (SAD), Sistemas de Informação Gerenciais (SIG), Sistemas de Trabalhadores do Conhecimento (STC), Sistemas de Automação de Escritório (SAEs) e Sistemas de Processamento de Transações (SPT). Esses sistemas são ligados aos níveis organizacionais.

Um sistema de informação tem como objetivo armazenar e processar dados através de um sistema computacional, de forma que esses dados possam ser usados de forma benéfica aos indivíduos que utilizam desse sistema. De acordo com ([SILVA et al., 2018](#)), um sistema de informação voltado a apoiar o Programa Nacional de Imunização (PNI) com o objetivo de realizar coleta de dados das atividades de vacinação, teria seu uso voltado para gerenciar a qualidade das informações que fossem coletadas de cada usuário ao longo do tempo. Como forma de auxílio a imunização, esse sistema deve possuir notificações de vacinas agendadas e em atraso, além de eventos posteriores à vacinação. Ainda de acordo com ([SILVA et al., 2018](#)), a utilização desse tipo de tecnologia propicia um acompanhamento e identificação de bolsões de baixa cobertura vacinal, controle de doenças imunopreveníveis além de dar apoio a estudos sobre impacto das vacinas. A respeito do Programa Nacional de Imunização (PNI) e de seus sistemas de informação:

No Brasil, para gerenciar os dados de vacinações, é utilizado o SIPNI, O SIPNI foi desenvolvido pelo PNI em parceria com o Departamento de Informática do SUS (DATASUS), com a finalidade de agregar os sistemas existentes em um único software e de modo interligado. Ele substituiu os sete módulos ou subsistemas: Sistema de Informação de Estoque e Distribuição de Imunobiológicos (SI-EDI); Sistema de Informação de Apuração dos Imunobiológicos Utilizados (SI-AIU); Sistema de Informação de Eventos Adversos Pós Vacinais (SI-EAPV); Sistema de Informação do Programa de Avaliação de Instrumento de Supervisão (SI-PAIS); Sistema de Informação do Programa de Avaliação do Instrumento de Supervisão em Sala de Vacinação (SI-PAISSV); Sistema de Informação do Centro de Referência de Imunobiológicos Especiais (SI-CRIE); e o Sistema de Informação de Avaliação do Programa de Imunização (SI-API) ([SILVA et al., 2018](#)).

De acordo com Marin ([MARIN, 2010](#)) sobre a importância de um sistema de informação para a área de saúde, a informação é essencial aos profissionais da saúde, sendo

muito importante em processos como cuidado, gerenciamento e avaliação, sendo assim o uso de um sistema de informação traria uma maior qualidade na tomada de decisões.

Além de ter a capacidade de gerenciar informações sobre vacinações, de acordo com (FORCE et al., 2015) um SI voltado a imunização tem a capacidade de aumentar os números de vacinações e reduzir o número de doenças que podem ser prevenidas por vacinas. Segundo (GROOM et al., 2015), nos Estados Unidos, de 2001 a 2003, 8 sistemas de imunização de oito regiões diferentes foram estudados e os dados obtidos foram de que 85% de jovens menores de 19 anos tinham ao menos dois registros de vacinas no sistema. O estudo também cita um aumento na taxa de vacinação que ocorreu na Austrália entre 1997 e 2007, onde houve um aumento de 64% para 92,7% de taxas de imunização completa entre crianças de 24 meses associadas à implementação do sistema de informação.

2.2 Programa Nacional de Imunizações

O Brasil possui o PNI que é o Programa Nacional de imunizações. De acordo com (BRASIL, 2015), ele foi iniciado em 1973. Seu objetivo é conter as doenças para as quais haviam vacinas, através da vacinação de toda a população brasileira, onde dispõe de 28 vacinas, 13 soros heterólogos e 4 homólogos, sendo 18 dessas vacinas presentes no cartão de vacinação. Desde sua implementação, o PNI se mostrou um sucesso em manter um controle sobre as doenças imunopreveníveis, tendo erradicado doenças de alcance mundial como a varíola e a poliomielite.

Segundo o Ministério da Saúde (BRASIL, 2015), o PNI trouxe grandes avanços sendo responsável pela erradicação do sarampo, poliomielite e o tétano neonatal, além de ter controlado doenças como difteria, coqueluche e tétano acidental, hepatite B, meningites, febre amarela, formas graves da tuberculose, rubéola e caxumba. Sobre o desempenho do PNI no Brasil:

No Brasil, ainda que as coberturas vacinais tenham se mantido abaixo da meta e as taxas de abandono elevadas, estudo relativo à internação por diarreias no Brasil, antes (2005) e depois (2011) da implantação da vacina, mostrou queda nas taxas de internação e óbitos, reduzindo em 36,5% as internações de crianças na idade de 1 a 4 anos, declinando os óbitos de 54,4 por 10 mil (2005) para 44 por 10 mil (2011), e provocando a queda na taxa de mortalidade por 100 mil menores de 5 anos, que declinou de 3,1 (2005) para 2,1 (2011) (DOMINGUES; TEIXEIRA, 2013).

A imagem abaixo mostra o calendário de vacinação estabelecido pelo PNI e o Ministério da Saúde do ano de 2020 que apresenta todas as vacinas que todo cidadão brasileiro deve tomar desde o nascimento até a fase adulta.

Figura 1 – Calendário Nacional de vacinação.

CALENDÁRIO NACIONAL DE VACINAÇÃO/2020/PNI/NIMS																			
Vacinas	BCG	Hepatite B	VORH Rotavírus	Pentavalente (DTP+Hib+Hep B)	DTP	VIP e VOP	Pneumocócica 10	Meningocócica C	Febre Amarela	Triplice Viral	Tetra Viral	Vacina monovalente	Hepatite A	HPV	Meningocócica ACWY	Dupla Adulto	dTpa (adulto)		
Protege contra	Formas graves da tuberculose	Hepatite B	Rotavírus	Difteria, Tétano, Hepatite B e meningite por Haemophilus influenzae tipo b	Difteria, Tétano e Coqueluche	Poliomielite	Pneumonia, otite, meningite e outras doenças causadas pela pneumococo	Doença invasiva causada pela Meningococo	Febre Amarela	Sarampo, Caxumba e Rubéola	Sarampo, Caxumba, Rubéola e Varicela	Varicela	Hepatite A	HPV	Doença invasiva causada pela meningococo	Difteria e Tétano	dTpa (adulto)		
																		Grupo Alvo	Idade
																		Ao nascer	Dose Única (1)
Criança		Dose ao nascer (2)																	
			1ª dose			1ª dose VIP (1)													
							2ª dose VIP (1)	1ª dose											
							3ª dose VIP (1)	2ª dose											
									Dose Inicial										
										1ª dose									
											Dose Única (1)								
Adolescente		3 doses: a partir de 7 anos de idade (5)																	
	10 a 19 anos																		
Adulto		3 doses (5)																	
	20 a 59 anos																		
Idoso		3 doses (5)																	
	60 anos ou mais																		
Gestante		3 doses (5)																	

Fonte: (BRASIL, 2020)

2.3 Desenvolvimento de Aplicativos Multiplataforma

Uma estratégia utilizada para o desenvolvimento multiplataforma consiste na criação de um único código que pode ser compilado para mais de um sistema operacional. Segundo El-Kassas (EL-KASSAS et al., 2017), o conceito principal de uma solução multiplataforma é desenvolver um aplicativo uma vez e executá-lo em qualquer lugar.

A economia de tempo e recurso do desenvolvimento multiplataforma ocorre pois com esse processo de desenvolvimento se pode evitar o retrabalho com os processos de criação de um software. Quando se decide desenvolver nativamente para mais de um sistema operacional (SO), precisa se ter para cada plataforma um time especializado de desenvolvimento para aquela plataforma em uma determinada IDE e com uma linguagem específica, cada componente de software precisa ser pensado e desenvolvido mais de uma vez, precisa ter uma etapa de testes para cada SO. Já com o desenvolvimento multiplataforma as etapas citadas anteriormente só precisam ser desenvolvidas e testadas uma vez. Segundo (EL-KASSAS et al., 2017) quando uma empresa pretende desenvolver para mais de um sistema operacional e não opta pelo desenvolvimento multiplataforma, inevitavelmente o time de desenvolvimento dessa empresa irá perder um grande tempo de desenvolvimento por ter que passar pelo período aprendizado e o tempo de se habituar com o ambiente de desenvolvimento de diferentes plataformas.

Um dos grandes motivos da popularização do desenvolvimento multiplataforma é a necessidade de se ter uma mesma aplicação sendo executada em diversos sistemas operacionais. Além disso, de acordo com El-Kassas (EL-KASSAS et al., 2017), o número de dispositivos móveis aumenta todo ano, e com isso a uma demanda maior por aplicações que atendam as necessidades dos usuários.

De acordo com (EL-KASSAS et al., 2017), os tipos de aplicações móveis são aplicações Web, aplicações nativas e aplicações híbridas. Aplicações Web consistem em aplicações que utilizam tecnologia Web no seu desenvolvimento (*HTML, CSS, Javascript*). Elas não precisam ser instaladas e podem ser acessadas pela *URL*. Aplicações nativas são aplicações desenvolvidas com o uso de linguagens e ferramentas fornecidas pela própria plataforma. Elas também podem ser instaladas por downloads ou pela própria *store*. As aplicações híbridas são as que utilizam tecnologias de desenvolvimento Web mas são renderizadas dentro de uma aplicação nativa utilizando uma *Web View*. Assim como as aplicações nativas, elas podem ser instaladas por downloads ou pela própria *store*.

O *framework* utilizado para o desenvolvimento deste trabalho é o React Native, que é um *framework* de desenvolvimento Javascript criado pelo Facebook. Apesar da definição de aplicativo nativo que diz que aplicativos nativos são desenvolvidos na linguagem e ferramenta fornecida pela plataforma, o aplicativo desenvolvido com React Native também é um aplicativo nativo pois seu código em Javascript é convertido para a linguagem

nativa da plataforma. Ele é voltado para o desenvolvimento multiplataforma móvel, dando suporte ao desenvolvimento para Android e iOS. Segundo (EISENMAN, 2015), o React Native utiliza o JavaScript e JSX (linguagem de marcação baseada em XML). Através do React Native, são chamadas Interfaces de Programação de Aplicações (APIs) nativas de renderização em Objective-C para iOS ou em Java para Android. Diferente de outras linguagens multiplataforma, o React Native utiliza a interface móvel nativa ao invés de carregar no aplicativo uma *Web View* como outros *frameworks* que funcionam como um navegador móvel para a interface. Isso faz com que o React Native tenha uma performance similar a aplicativos nativos e possua acesso a recursos nativos da plataforma como a câmera, localização e sensores.

3 Desenvolvimento

Para a etapa de desenvolvimento, foram definidas as atividades de coleta de requisitos, a criação do diagrama de caso de uso UML, elaboração do banco de dados, concepção dos *mockups* e descrição das ferramentas e tecnologias que foram utilizadas.

Para o levantamento dos requisitos, foram escolhidos e analisados alguns aplicativos da *Play Store*, do sistema operacional *Android*. O critério de escolha dos aplicativos foram aplicações voltadas a saúde e com o foco na gestão de vacinação. De acordo com Sommerville (IAN, 2011), o levantamento de requisitos é fundamental no processo de desenvolvimento, já que com a definição das necessidades de forma abstrata é possível facilitar o entendimento das funcionalidades e utilidades do aplicativo por outras pessoas, além de verificar se essas funcionalidades irão atender as necessidades desejadas. Assim, outros indivíduos ou possíveis clientes podem validar a funcionalidade de um software antes do seu desenvolvimento.

3.1 Aplicativos Semelhantes

3.1.1 Conecte SUS

O aplicativo do Ministério da Saúde, durante a elaboração do TCC 1 se chamava Meu digisus. Porém, ele foi refatorado e agora se chama Conecte SUS. Anteriormente, foi observado que uma das vantagens do aplicativo são as inúmeras funções, além do cadastro de vacinas, como retirada de medicamentos, agendar consulta, doação de sangue e algumas outras. O aplicativo também traz uma descrição para que o usuário entenda a importância de cada vacina. Como ponto negativo, o antigo Meu digisus dependia de uma base de dados do governo, o que não possibilita que o usuário mantenha a gestão de suas próprias vacinas. O sistema também apresenta lentidão e em alguns casos erro no carregamento dos dados. Após o aplicativo ser refeito ele basicamente manteve as mesmas funcionalidades e adicionou melhorias na interface tendo um foco em deixar o aplicativo mais amigável ao usuário.

3.1.2 Calendário de vacinação

O aplicativo Calendário de vacinação separa as vacinas em determinadas faixas etárias, deixando de forma organizada e fácil para o usuário achar qual vacina ele deve tomar ao ir em um posto de saúde. O aplicativo também informa o intervalo de tempo entre as doses e traz uma descrição detalhada de cada vacina, informando suas indicações, contraindicações, esquema, conservação e validade. Como ponto negativo, o sistema não

possibilita um cadastro de vacinas e usuário em um banco de dados, o que não permite que o usuário mantenha a gestão e informações de suas próprias vacinas.

Sendo assim, o aplicativo se caracteriza como um aplicativo informativo com seu foco totalmente voltado para conscientizar e informar o usuário, dispondo de informações bem detalhadas e específicas sobre cada vacina. De acordo com as informações presentes no aplicativo, seu objetivo é atender as necessidades dos profissionais e estudantes das áreas de saúde sobre atualizações no Programa Nacional de Imunização.

3.1.3 Carteirinha de Criança

O aplicativo Carteirinha de Criança tem capacidade de efetuar vários cadastros de carteirinhas de vacinação virtuais para diversas pessoas de uma mesma família. O aplicativo também irá informar por meio de um cadastro quais vacinas o usuário possui ou não possui, além de informar a data que o usuário deveria ter tomado cada vacina com base na sua data de nascimento. Como pontos negativos, o aplicativo apresenta uma falta de descrição e informações detalhadas sobre cada vacina, além de só possuir para cadastro vacinas de crianças de até 4 anos. Como diferencial, o aplicativo informa as próximas vacinas e as que estão pendentes para serem administradas.

3.2 Descoberta de requisitos

O levantamento dos requisitos ocorreu após a análise dos aplicativos com conceito similar a ideia do projeto. Além disso, também foi observado o cartão de vacina físico utilizado pelo SUS, para que fosse possível levantar os requisitos de forma que o aplicativo irá se basear em um cartão de vacina virtual. Sobre a descoberta de requisitos:

A descoberta de requisitos (às vezes, chamada elicitación de requisitos) é o processo de reunir informações sobre o sistema requerido e os sistemas existentes e separar dessas informações os requisitos de usuário e de sistema. Fontes de informação durante a fase de descoberta de requisitos incluem documentação, *stakeholders* do sistema e especificações de sistemas similares (IAN, 2011).

3.2.1 Requisitos funcionais

- O usuário deve efetuar o login ou o cadastro para a utilização do aplicativo.
- O usuário poderá filtrar suas vacinas pelo ano selecionado.
- O usuário poderá selecionar uma das vacinas para obter mais detalhes.
- O usuário poderá deletar uma vacina ao arrastar ela para o lado ou ele terá acesso a essa opção ao selecionar uma vacina.

- O usuário poderá acessar outra tela que o irá informar quais as próximas vacinas que devem ser tomadas e sua data.
- O usuário poderá exportar suas vacinas em pdf.

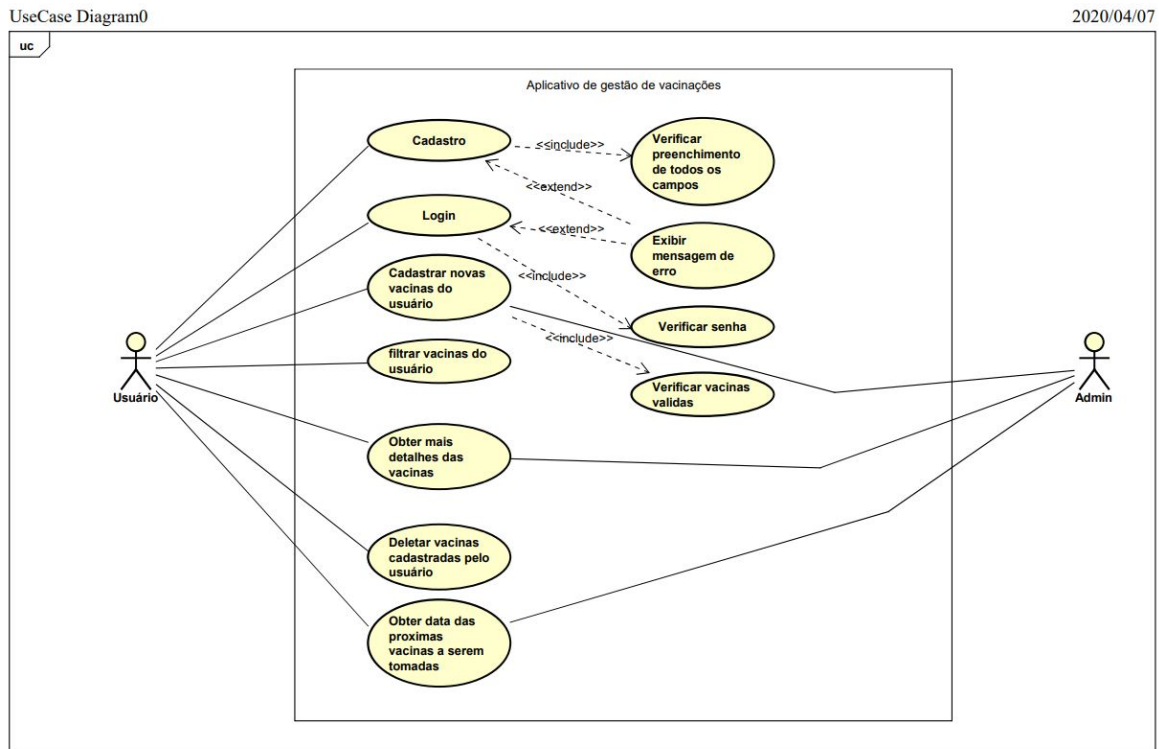
3.2.2 Requisitos não funcionais

- O sistema deve ter o mesmo funcionamento em diferentes sistemas operacionais suportados.
- O sistema deve ter uma interface amigável a novos usuários.
- Apenas usuários administradores devem possuir acesso à camada de administração do sistema.
- Os dados dos usuários devem ser armazenados de forma segura e usados apenas para a funcionalidade do aplicativo.

3.3 Diagrama de caso de uso

Como uma das etapas do desenvolvimento, foi elaborado o diagrama de caso de uso da aplicação, utilizando a ferramenta Astah ([ASTAH, 2020](#)), focada em elaboração de diagramas da UML. “Em sua forma mais simples, um caso de uso identifica os atores envolvidos em uma interação e dá nome ao tipo de interação. Essa é, então, suplementada por informações adicionais que descrevem a interação com o sistema. A informação adicional pode ser uma descrição textual ou um ou mais modelos gráficos, como diagrama de sequência ou de estados da UML”. ([IAN, 2011](#)).

Figura 2 – Diagrama UML.



Fonte: Autor do trabalho (2020)

Na figura 1, podemos observar o diagrama de caso de uso, onde os agentes são o administrador do software que ficará responsável pela manutenção do sistema e atualização de quais vacinas podem ser cadastradas pelos usuários, e os usuários que serão serão os utilizadores do aplicativo e através dele irão cadastrar e consultar suas vacinas e informações.

3.3.1 Casos de Uso relacionado ao ator Usuário

3.3.1.1 Caso de Uso 1

- Caso de uso 1: Cadastrar.
- Contexto de uso: Quando o usuário não possuir uma conta cadastrada para utilizar o aplicativo.
- Precondições: Nenhuma.
- Condições e sucesso: Email já não ter sido registrado, todos os dados serem preenchidos de forma correta e o usuário é redirecionado para tela de login.
- Falha e proteção: A conta não é registrada e uma mensagem de erro é exibida ao usuário.

- Ator principal: Usuário.
- Condição para começar: Clicar na opção “Não sou cadastrado”.
- Dados Coletados:
 - Nome
 - Email
 - Senha
 - Endereço
 - Data de nascimento
 - Foto de perfil (Opcional)

3.3.1.2 Caso de Uso 2

- Caso de uso 2: *Login*.
- Contexto de uso: Utilizado para fazer autenticação do usuário para que ele possa acessar seus dados.
- Precondições: Possuir cadastro.
- Condições e sucesso: Inserir email e senha corretos e válidos, usuário será redirecionado para a página principal estando logado em sua conta.
- Falha e proteção: Inserir email ou senha incorretos ou inválidos, uma mensagem aparecerá informando ao usuário sobre o erro.
- Ator principal: Usuário.
- Condição para começar: Iniciar o aplicativo ou fazer logout.

3.3.1.3 Caso de Uso 3

- Caso de uso 3: Cadastrar novas vacinas.
- Contexto de uso: O usuário poderá fazer o cadastro de novas vacinas que foram aplicadas nele.
- Precondições: A vacina precisa estar pré cadastrada no sistema pelo administrador e o usuário precisa estar logado para fazer o cadastro.
- Condições e sucesso: Preencher todos os campos obrigatórios, uma mensagem será exibida falando que cadastro foi feito.

- Falha e proteção: Não preencher todos os campos obrigatórios, uma mensagem informando erro no cadastro aparecerá para o usuário.
- Ator principal: Usuário.
- Condição para começar: Acessar a página de cadastro de vacinas através da página principal.
- Dados coletados:
 - Qual a vacina (Entre as pre-cadastradas pelo administrador)
 - lote (Opcional)
 - Data de aplicação
 - Data da segunda dose (Opcional)
 - Data da terceira dose (Opcional)
 - Data da quarta dose (Opcional)

3.3.1.4 Caso de Uso 4

- Caso de uso 4: Filtrar vacinas.
- Contexto de uso: O usuário poderá filtrar suas vacinas na tela principal através do ano de aplicação da vacina.
- Precondições: O usuário precisa ter ao menos uma vacina cadastrada e estar logado.
- Condições e sucesso: Escolher o ano, Somente vacinas do ano escolhido serão exibidas.
- Falha e proteção: Anos que não tem vacinas cadastradas não serão exibidos na hora da escolha do ano para realizar a filtragem.
- Ator principal: Usuário.
- Condição para começar: Acessar a página principal.

3.3.1.5 Caso de Uso 5

- Caso de uso 5: Obter detalhes de vacinas.
- Contexto de uso: O usuário poderá visualizar informações de vacinas específicas que foram cadastradas pelo administrador do sistemas.
- Precondições: O usuário precisa ter ao menos uma vacina cadastrada e estar logado.
- Condições e sucesso: Ter ao menos uma vacina cadastrada.

- Ator principal: Usuário.
- Condição para começar: Acessar a página principal e através dela selecionar a vacina que se deseja ver informações.

3.3.1.6 Caso de Uso 6

- Caso de uso 6: Deletar vacinas cadastradas.
- Contexto de uso: O usuário poderá deletar as vacinas que ele cadastrou.
- Precondições: O usuário precisa ter ao menos uma vacina cadastrada e estar logado.
- Condições e sucesso: Confirmar que deseja deletar a vacina, a vacina é deletada do banco de dados.
- Falha e proteção: Uma mensagem de confirmação é exibida para o usuário, caso ocorra algum erro uma mensagem de erro será exibida ao usuário.
- Ator principal: Usuário.
- Condição para começar: Acessar a página da vacina e clicar no ícone da lixeira.

3.3.1.7 Caso de Uso 7

- Caso de uso 7: Obter data das próximas doses de vacinas.
- Contexto de uso: O usuário visualiza um calendário com a data das próximas doses de vacinas.
- Precondições: O usuário precisa ter ao menos uma vacina cadastrada e estar logado.
- Condições e sucesso: Cadastrar a data das próximas doses de suas vacinas, um calendário com essas datas será exibido na página de cada vacina.
- Falha e proteção: Caso ocorra algum erro uma mensagem de erro será exibida ao usuário.
- Ator principal: Usuário.
- Condição para começar: Acessar a página da vacina.

3.3.2 Casos de Uso relacionado ao ator Administrador

3.3.2.1 Caso de Uso 1

- Caso de uso 1: Cadastrar novas vacinas.
- Contexto de uso: O administrador poderá cadastrar novos tipos de vacinas para que o usuário possa registrar em seu cartão de vacina.
- Precondições: Possuir acesso de administrador no banco de dados e acesso ao código fonte do aplicativo.
- Condições e sucesso: Inserir dados válidos, a vacina estará disponível no aplicativo.
- Falha e proteção: Inserir dados inválidos, a vacina não estará disponível no aplicativo.
- Ator principal: Administrador.
- Condição para começar: Acessar o banco de dados através do MongoDB Atlas.
- Dado coletado:
 - Nome da vacina
 - Descrição (Opcional, pode ser cadastrada ou editado futuramente)

3.3.2.2 Caso de Uso 2

- Caso de uso 2: Cadastrar detalhes de vacinas.
- Contexto de uso: O administrador poderá cadastrar novas informações sobre uma vacina específica ou inserir links em vacinas já cadastradas.
- Precondições: Possuir acesso de administrador no banco de dados e acesso ao código fonte do aplicativo.
- Condições e sucesso: Inserir dados válidos, os detalhes ficarão disponíveis no aplicativo.
- Falha e proteção: Inserir dados inválidos, os detalhes não ficarão disponíveis no aplicativo.
- Ator principal: Administrador.
- Condição para começar: Acessar o banco de dados através do MongoDB Atlas.
- Dado coletado:
 - Links com informações relacionadas
 - Descrição

3.3.2.3 Caso de Uso 3

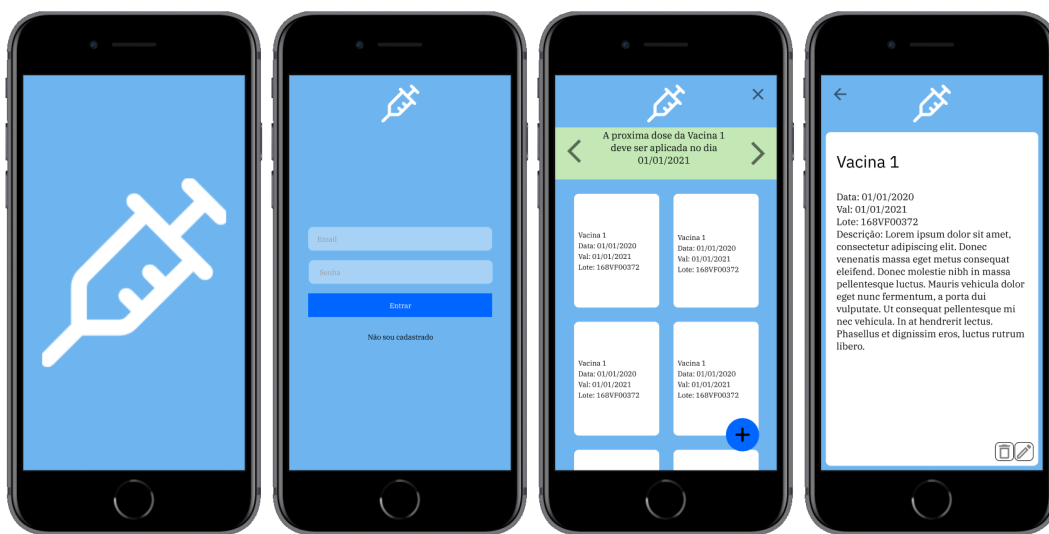
- Caso de uso 3: Administrar sistema de notificações e data de próximas vacinas.
- Contexto de uso: O administrador poderá gerenciar o sistema de notificação de próximas doses de vacina.
- Precondições: Possuir acesso de administrador no banco de dados e acesso ao código fonte do aplicativo.
- Condições e sucesso: Sistema funcionando de notificação disponível.
- Falha e proteção: Sistema de notificação indisponível.
- Ator principal: Administrador.
- Condição para começar: Acessado pelo MongoDB Atlas.

3.4 Protótipos de telas

Com o objetivo de se obter uma representação de média fidelidade do *design* do aplicativo, os *mockups* foram feitos através da ferramenta Figma ([FIGMA, 2020](#)). O objetivo é trazer o conceito de como a aplicação será e quais suas funcionalidades, mas trazendo a representação de forma estática.

“Protótipos do sistema permitem aos usuários ver quão bem o sistema dá suporte a seu trabalho. Eles podem obter novas ideias para requisitos e encontrar pontos fortes e fracos do software; podem, então, propor novos requisitos do sistema. Além disso, o desenvolvimento do protótipo pode revelar erros e omissões nos requisitos propostos. A função descrita em uma especificação pode parecer útil e bem definida” ([IAN, 2011](#)).

Figura 3 – Mockup.



Fonte: Autor do trabalho (2020)

Como visto na [Figura 3](#), as telas estão dispostas da ordem da esquerda para a direita e os usuários irão encontrar as telas nessa ordem ao utilizar o aplicativo. A primeira tela apresenta a tela de carregamento que é vista ao iniciar o aplicativo. A segunda tela é a tela de login onde o usuário pode ser redirecionado para uma tela de cadastro ou realizar o login e seguir para a próxima tela. A terceira tela representa a tela principal do aplicativo onde ele pode ver todas as suas vacinas cadastradas e ser redirecionado para a tela de próximas doses ou de cadastro de novas vacinas e a última tela representa a tela de mais informações da vacina, que é obtida ao clicar em uma determinada vacina na tela principal. A partir dessa tela também é possível ser direcionado para uma tela onde se pode atualizar os dados dessa vacina ou a excluir.

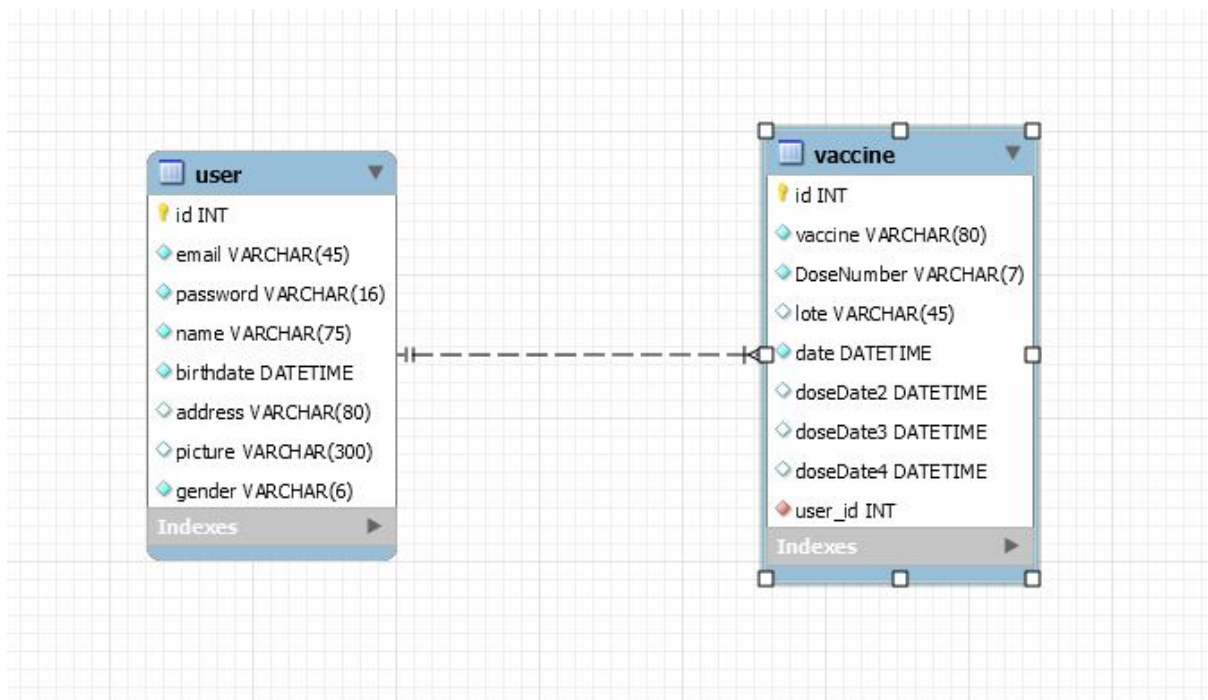
Houveram algumas alterações visuais que se observaram necessárias durante o desenvolvimento, como por exemplo a alteração da cor de fundo da tela principal para dar mais destaque à lista de vacinas.

3.5 Banco de dados

Para o planejamento de como e quais dados serão armazenados no aplicativo, foi realizada a elaboração de um diagrama entidade-relacionamento através da ferramenta MySQL workbench ([MYSQL, 2020](#)). De acordo com análise da coleta de requisitos, o banco de dados necessita de duas tabelas, uma para armazenamento de dados das vacinas e outro para informações do usuário.

As informações necessárias para os usuários são o nome, o email, a senha, a data de nascimento, o endereço (preenchimento não obrigatório), a foto (preenchimento não obrigatório) e o gênero. Já os dados das vacinas são o nome da vacina, o número de doses, o lote, a data da aplicação e a data das próximas doses (podendo não ter mais doses ou chegar a mais de 3 doses). O diagrama pode ser observado na [Figura 4](#).

Figura 4 – Diagrama entidade-relacionamento.



Fonte: Autor do trabalho (2020)

3.5.1 MongoDB

Para o armazenamento dos dados, a ferramenta de banco de dados escolhida foi o MongoDB, que tem seu foco total em oferecer soluções de armazenamento em nuvem e ser altamente escalável. Através de um *cluster* de servidores, ele armazena os dados na nuvem no formato JSON (JavaScript Object Notation). A versão do MongoDB utilizada no trabalho foi a versão gratuita que representa certas limitações. Quando comparada a planos pagos oferecidos pela ferramenta, algumas das limitações presentes na versão gratuita é a quantidade limitada de servidores disponíveis a serem escolhidos, a realização de backups automáticos de dados, os *clusters* são compartilhados, a performance é menor e o espaço de armazenamento é limitado a 512 MB.

De acordo com Chodorow ([CHODOROW, 2013](#)), o MongoDB é um banco de dados orientado a documentos. Assim, ele não é relacional. A principal razão para não optar por um modelo relacional é alcançar uma escalabilidade maior, de forma facilitada. Esse modelo substitui o conceito de linhas por um modelo mais flexível, um modelo documental. Ele faz isso deixando que uma hierarquia complexa possa ser representada em um único

registro. Nesse modelo, também não há definição de tamanho ou tipos para o que será armazenado. Portanto, há uma facilidade em adicionar e remover campos.

Inicialmente, a ideia era utilizar um banco de dados relacional, o SQL. Porém, durante o desenvolvimento do trabalho, foi optado por um banco de dados não relacional, que é o MongoDB. Essa escolha foi feita pelo fato do sistema não possuir muitas tabelas no modelo SQL, além de apostar em um desempenho maior ao não se usar o modelo relacional. Além disso, o MongoDB apresenta uma maior facilidade para se fazer a escalabilidade do banco de dados.

O processo de mudança do modelo relacional para o modelo não relacional não foi complexo, já que o modelo relacional ainda não havia sido implementado. Por isso, foi possível começar a implementação do modelo não relacional do zero e, com isso, foi possível utilizar o modelo entidade-relacionamento como auxílio na criação do banco não relacional.

O que é mostrado na [Figura 5](#) é o que se encontra após fazer o login no site do MongoDB e criar um novo projeto. Ao se escolher a opção de cluster gratuito, só é possível escolher um número limitado de regiões e servidores de armazenamento em nuvem. Dentre eles é possível escolher entre os serviços da Amazon (aws), Google (Google Cloud) e Microsoft (Azure), que também oferecem soluções pagas com limite maior de armazenamento, menos limitações no hardware e mais opções de locais a serem escolhidos.

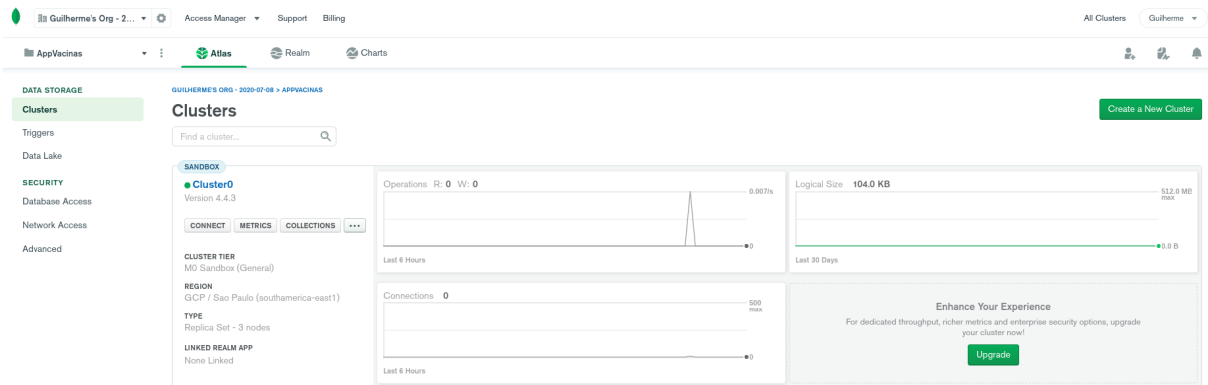
Figura 5 – Regiões e servidores de armazenamento em nuvem

The screenshot displays the MongoDB Atlas configuration interface. At the top, the 'Cloud Provider & Region' section shows three options: AWS, Google Cloud, and Azure. The AWS option is selected and highlighted with a green border. To the right, the selected configuration is 'AWS, N. Virginia (us-east-1)'. Below this, a 'Recommended region' section lists various regions categorized by continent: EUROPE (Ireland, Frankfurt), NORTH AMERICA (N. Virginia, Oregon), ASIA (Mumbai, Singapore), and AUSTRALIA (Sydney). The 'N. Virginia (us-east-1)' region is highlighted with a green border. Below the region selection, the 'Cluster Tier' is set to 'M0 Sandbox (Shared RAM, 512 MB Storage)' with a right-pointing arrow and the word 'Encrypted' below it. The 'Additional Settings' are set to 'MongoDB 4.4, No Backup' with a right-pointing arrow. The 'Cluster Name' is set to 'Cluster0' with a right-pointing arrow.

Fonte: (MONGODB, 2021a)

Na Figura 6, é possível observar o MongoDB Atlas que permite ao usuário o gerenciamento do banco de dados e dos *clusters*, sendo possível, através dessa ferramenta, fazer todo o gerenciamento através do navegador. De acordo com a documentação do MongoDB Atlas (MONGODB, 2021b), o Atlas fica responsável por toda a complexidade de implementação, o gerenciamento, implementações e manutenções no provedor de serviço de nuvem escolhido.

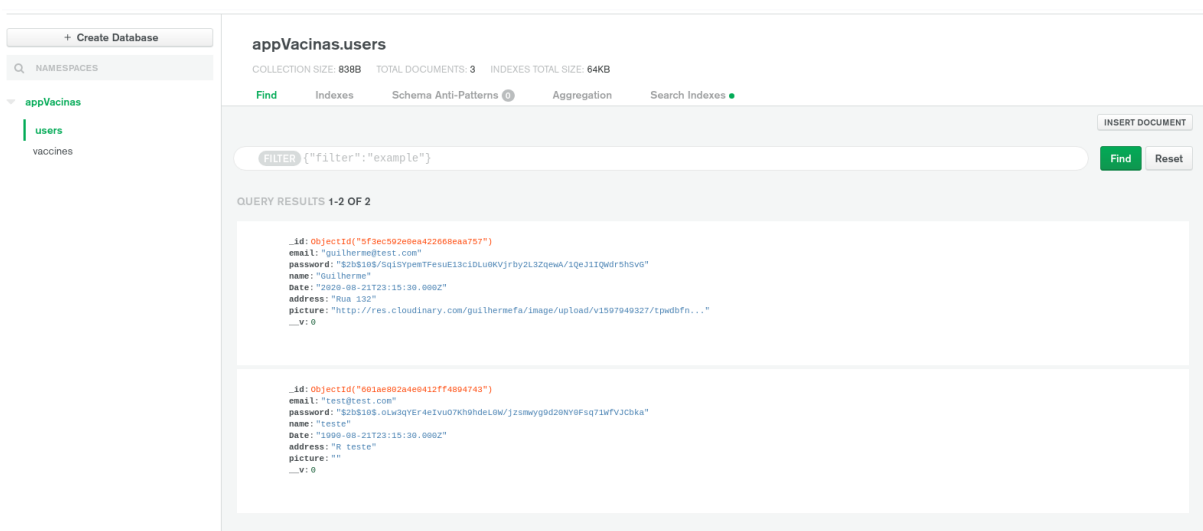
Figura 6 – MongoDB Atlas



Fonte: (MONGODB, 2021a)

Na Figura 7, é mostrado os dados e todos os campos que são armazenados no banco. Essa imagem mostra os dados de usuário, onde é possível observar que os dados de senha estão criptografados com auxílio da biblioteca bcrypt do Javascript para manter um nível de segurança caso o banco de dados seja comprometido. Também é possível ver a estrutura do banco de dados, similar a um arquivo JSON que não possui tipagem nos campos que armazenam os dados.

Figura 7 – Database



Fonte: (MONGODB, 2021a)

3.6 Framework para o desenvolvimento multiplataforma

Para o desenvolvimento multiplataforma, é necessário o uso de um *framework* que possibilite obter um aplicativo para as múltiplas plataformas alvo, possuindo apenas

um código fonte. O que se altera entre cada *framework* é que alguns *frameworks* criam aplicativos nativos que possibilitam a conversão de um código em determinada linguagem para um código nativo do SO alvo. Outros frameworks se baseiam em aplicativos híbridos que funcionam com uma *Web View*. Neste caso, o aplicativo é criado utilizando tecnologias de desenvolvimento Web (HTML, CSS, Javascript).

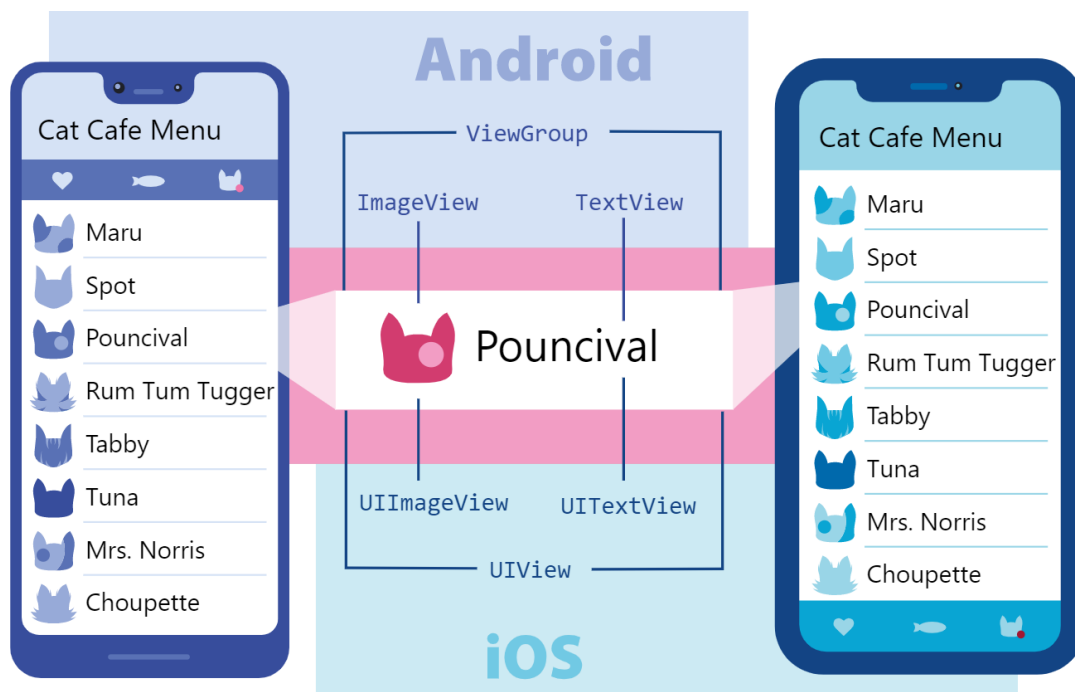
Durante o desenvolvimento deste trabalho foi utilizado um *framework* de desenvolvimento multiplataforma que permite o desenvolvimento de aplicativos nativos. De acordo com (EL-KASSAS et al., 2017), as vantagens do desenvolvimento multiplataforma é que o aplicativo irá alcançar mais usuários que usam plataformas diversas, tem uma facilidade de desenvolvimento maior já que o código é escrito somente uma vez e pode ser compilado para várias plataformas e também reduz o tempo de desenvolvimento e esforço necessários. Como ponto negativo existe a necessidade do *framework* estar em constante desenvolvimento, já que ele precisa ser atualizado toda vez que uma nova versão de um SO alvo é lançada.

3.6.1 React Native

O *framework* de desenvolvimento utilizado neste trabalho é o React Native que é capaz de converter seu código Javascript e JSX para o código nativo da linguagem do SO desejado, seja ele Android ou iOS. Entre as vantagens desse *framework* estão o fato dele ser desenvolvido pela equipe do Facebook e por isso ter uma grande e experiente equipe de desenvolvimento que sempre está atualizando o *framework*, sempre que uma nova versão do Android ou iOS é lançada. Além disso, ele não é dependente de tecnologias *Web* como muitos outros *frameworks* de desenvolvimento multiplataforma, que de acordo com (EL-KASSAS et al., 2017) representa uma perda de desempenho quando comparados a aplicativos que são nativos da plataforma e por ser nativo também pode chamar componentes nativos dos dispositivos que permitem acesso a recursos como câmera, GPS, e-mail e agenda.

De acordo com a documentação do React Native (NATIVE, 2021) o React Native usa o Javascript para acessar a API e criar a aparência e o comportamento da interface de usuário (UI) usando componentes da biblioteca React do Javascript. A *view* do React Native é um bloco da UI que serve para mostrar os elementos necessários do aplicativo na tela, uma *view* pode conter outras *view* e assim formar a UI do aplicativo, como visto na Figura 8. No Android, as *views* são escritas nas linguagens Java ou Kotlin, enquanto no iOS são usadas as linguagens Swift ou Objective-C. O React Native é capaz de chamar essas mesmas *views* utilizando componentes React.

Figura 8 – Diagrama da View no Android e iOS



Fonte: (NATIVE, 2021)

A Tabela 1 mostra os componentes principais do React Native em comparação com outras tecnologias de desenvolvimento que a biblioteca do React pode chamar.

Tabela 1 – Componentes principais.

Componentes da UI do React Native	View do Android	View do iOS	análogo a Web	Descrição
<View>	<ViewGroup>	<UIView>	<div> sem rolagem	Um container que suporta um layout com flexbox, estilo, manuseio ao toque e controle de acessibilidade.
<Text>	<TextView>	<UITextView>	<p>	Exibe, estiliza e encapsula strings de texto além de poder lidar com eventos de toque.
<Image>	<ImageView>	<UIImageView>		Exibe diferentes tipos de imagens
<ScrollView>	<ScrollView>	<UIScrollView>	<div>	Um contêiner de rolagem genérico que pode conter vários componentes e visualizações
<TextInput>	<EditText>	<UITextField>	<input type="text">	Permite ao usuário inserir texto

Fonte: (NATIVE, 2021)

3.7 Expo

Durante o desenvolvimento, foi utilizado o *framework* Expo que é voltado para dar apoio no desenvolvimento com React Native. O Expo fornece um ambiente de desenvolvimento para se trabalhar com o React Native. Com ele é possível ver as modificações feitas no código em tempo real através de um emulador do Android, iOS ou até mesmo utilizando um dispositivo físico através do aplicativo Expo Go.

Como pode ser visto na [Figura 9](#), o Expo é iniciado pelo terminal de comandos do Windows através do comando “expo start”. A partir daí, ele irá indicar algumas opções ao usuário para que ele possa iniciar a compilação do aplicativo em um emulador, enviar o link para o telefone, iniciar o DevTools no navegador ou fornecer um QR Code para que o usuário possa ver o aplicativo compilando no dispositivo físico através do aplicativo Expo Go através rede local.

Figura 9 – Inicialização do Expo no terminal



```
Selecionar Windows PowerShell
C:\Users\guilh\Documents\react\TCC\AppVacinas>npm start
> @ start C:\Users\guilh\Documents\react\TCC\AppVacinas
> expo start

Starting project at C:\Users\guilh\Documents\react\TCC\AppVacinas
Expo DevTools is running at http://localhost:19002
Press d to open DevTools now, or shift-d to always open it automatically.
Starting Metro Bundler
Opening DevTools in the browser...

Press ? to show a list of all available commands.
exp://192.168.31.231:19000



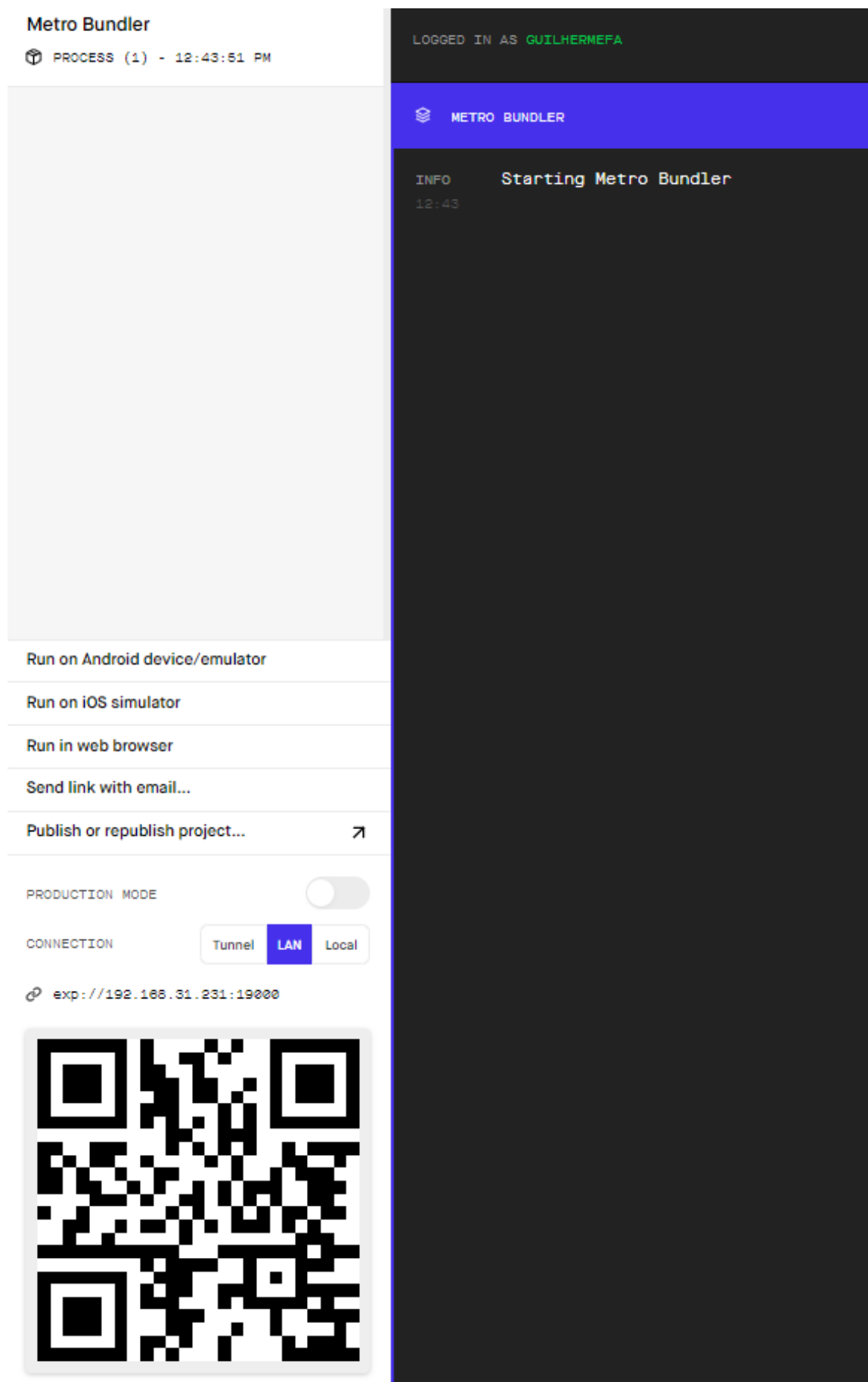
To run the app, choose one of:
> Scan the QR code above with the Expo app (Android) or the Camera app (iOS).
> Press a for Android emulator, or w to run on web.
> Press e to send a link to your phone with email.

Press ? to show a list of all available commands.
Logs for your project will appear below. Press Ctrl+C to exit.
```

Fonte: Autor do trabalho (2020)

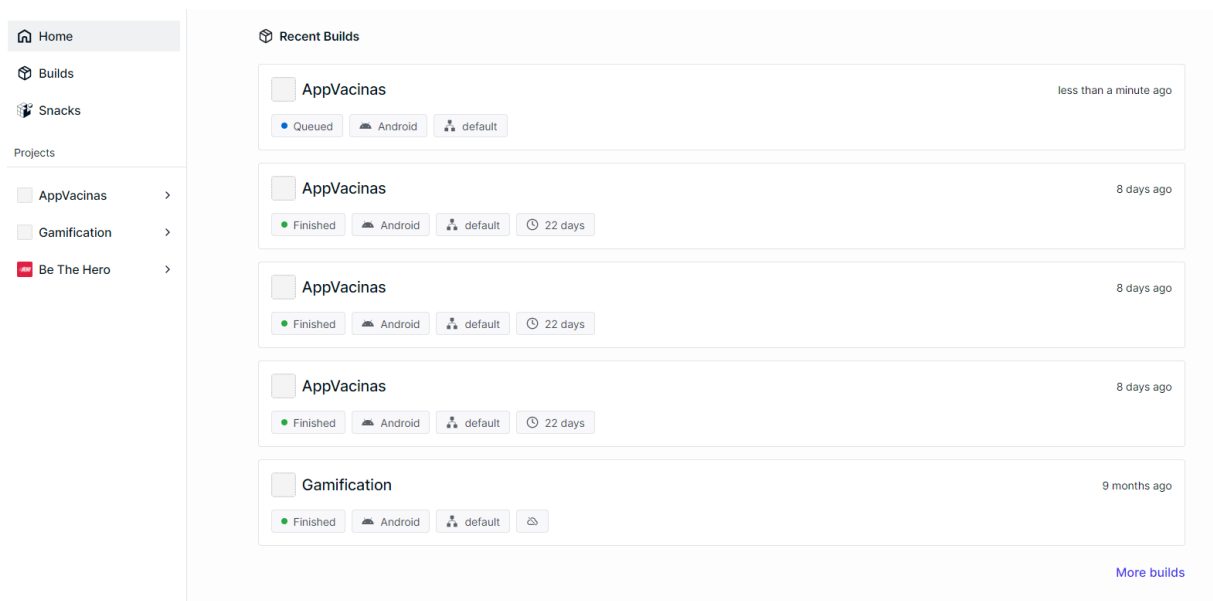
O Expo também possibilita a sua utilização através do DevTools que pode ser visto na imagem [Figura 10](#). Esta ferramenta possibilita o usuário utilizar as mesmas funções do terminal de comandos no navegador, possuindo uma interface muito mais amigável ao usuário.

Figura 10 – Inicialização do Expo através do Devtools no navegador



Uma outra função fornecida pelo Expo é a possibilidade de gerar um APK (*Android Package*), que é um arquivo de pacote do Android, pelos servidores do Expo ao invés da utilização do computador do próprio usuário. A vantagem de se fazer isso é que o usuário pode fazer *build* de aplicativos iOS sem necessariamente possuir um computador com macOS. Na [Figura 11](#), é possível ver no site do Expo as últimas *builds* geradas, estando marcadas com um círculo azul as que estão na fila aguardando para que o APK seja gerado, e em verde, aquelas para as quais esse processo já foi concluído.

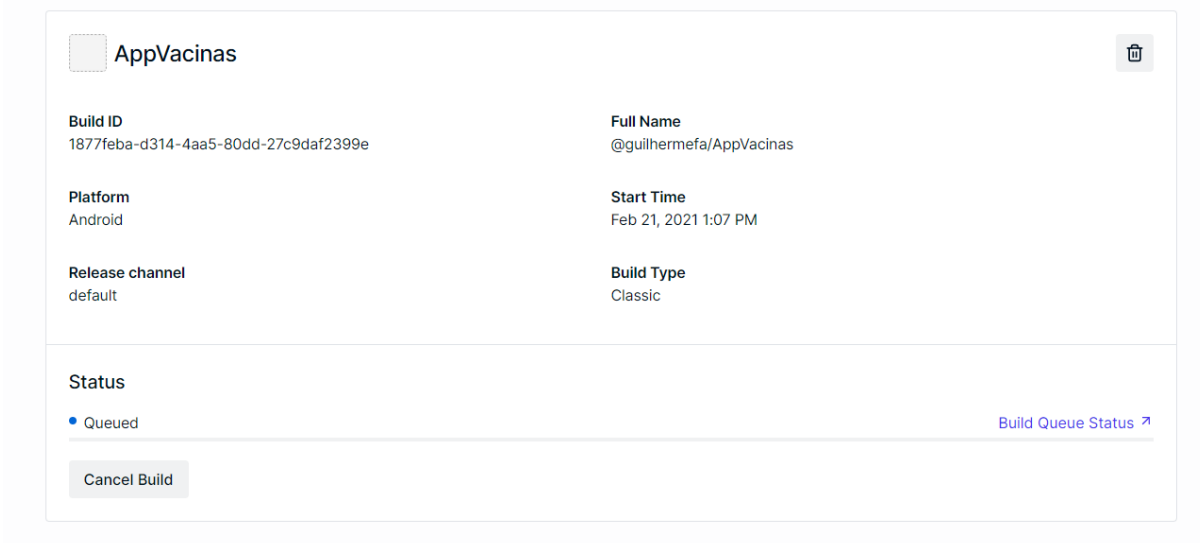
Figura 11 – Últimas *builds* criadas



Fonte: ([EXPO, 2021](#))

Na [Figura 12](#), é possível ver os detalhes da *build* que está sendo gerada para Android através do site do Expo, entre eles nome, id, data de início, status, plataforma alvo e tipo de *build*.

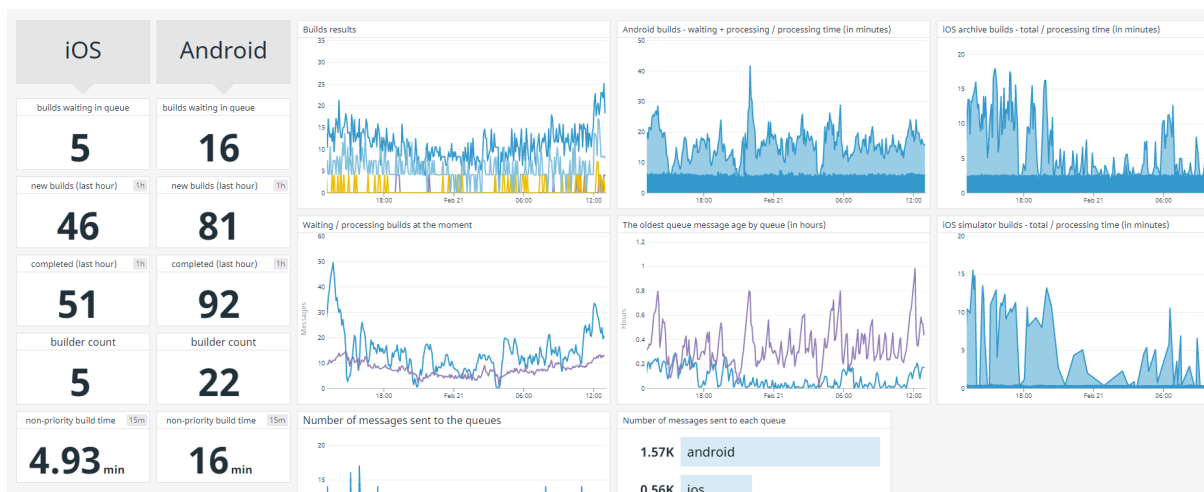
Figura 12 – Detalhes da build



Fonte: (EXPO, 2021)

Mesmo utilizando servidores próprios do Expo, a criação do aplicativo acontece de forma gratuita. Porém, como diversos desenvolvedores utilizam o Expo, é necessário esperar em uma fila para que a *build* da aplicação seja feita. Para o desenvolvimento deste trabalho foi utilizada a versão gratuita do Expo que não possui prioridade na fila de espera. Entretanto, existe um plano pago do Expo que possibilita ter prioridade na fila de espera. Na Figura 13, é possível ver detalhes da fila de espera.

Figura 13 – Fila para que o aplicativo seja criado



Fonte: (EXPO, 2021)

4 Resultados

Este capítulo expõe o que foi alcançado com o desenvolvimento do aplicativo, descrevendo e mostrando o aplicativo, comparando as diferenças entre as versões do Android e do iOS, e também apresentando o resultado dos testes.

4.1 O aplicativo

Nessa seção, será apresentado o aplicativo em sua versão para os dispositivos Android que foi desenvolvido utilizando o React Native, Expo e o MongoDB. As imagens do aplicativo que serão exibidas abaixo foram capturadas através do emulador presente no Android Studio, que simula o dispositivo Google Pixel 3.

Os dados que serão exibidos nesta seção presentes nas figuras do aplicativo não se tratam de dados reais contendo o número correto de doses de cada vacina, o número do lote de vacinas ou datas de aplicação, se tratam de dados modelos utilizados para exemplificar e simular o uso do aplicativo.

4.1.1 Login

Ao entrar no aplicativo, a primeira tela a ser exibida é a tela de login, que pode ser exibida na [Figura 14](#), que exige senha e email do usuário como credencial para o uso do aplicativo e, caso esse usuário não tenha um cadastro, ele pode ser redirecionado para a tela de cadastro e realizar esse registro. A credencial do aplicativo para o login ficam armazenadas no MongoDB que é um banco de dados não relacional com armazenamento em nuvem. Para a segurança dos usuários, as senhas armazenadas no banco de dados estão criptografadas com uso da biblioteca `bcrypt` do Javascript, que utiliza uma função *hash* para criptografar as senhas.

Figura 14 – Tela de Login

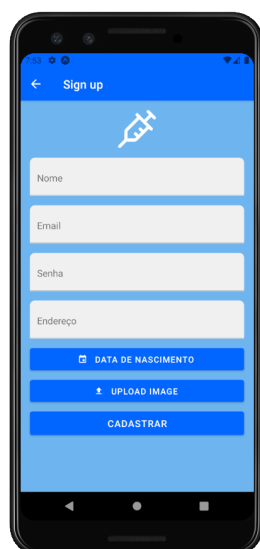


Fonte: Autor do trabalho (2021)

4.1.2 Cadastro de usuário

Caso o usuário entre no aplicativo e não seja registrado, ele irá para a tela de cadastro que pode ser vista na [Figura 15](#). Nessa tela, o usuário encontrará vários campos de texto onde ele deve inserir seus dados. Esses dados são nome, email, senha, endereço, data de nascimento e foto. O único dado que não é obrigatório para o cadastro é a foto de perfil. Caso dados obrigatórios não sejam inseridos ou inseridos de forma incorreta, o aplicativo exibirá uma mensagem avisando o erro ou a falta de preenchimento de campos obrigatórios.

Figura 15 – Tela de Cadastro de novo usuário



Fonte: Autor do trabalho (2021)

4.1.3 Página inicial

A página inicial é a tela principal do aplicativo, que pode ser vista na [Figura 16](#). É nela onde o usuário pode ver todas as suas vacinas que estão cadastradas e pode navegar para as outras telas. No quadro de vacina, é exibido o nome de cada vacina cadastrada em conjunto com sua data de aplicação e o número de doses. Também é possível tocar no quadro da vacina para ser encaminhado para a tela daquela vacina específica, onde é possível ver os dados completos da vacina. O usuário também pode ir para uma tela de cadastro de novas vacinas ou a tela que exhibe o tempo restante para as próximas doses das vacinas que possuem a necessidade da aplicação de mais de uma dose.

Figura 16 – Página inicial



Fonte: Autor do trabalho (2021)

4.1.4 Registro de vacina

A partir da página inicial, é possível acessar a tela de cadastro de vacina, que é exibida na [Figura 17](#). Nesta tela, é onde todas as informações das vacinas serão cadastradas pelo usuário. Primeiramente, o usuário irá selecionar qual vacina deseja cadastrar. Destas vacinas que serão selecionadas, somente o administrador do sistema pode cadastrar novos tipos de vacinas através do MongoDB Atlas. O próximo item a ser cadastrado é o número de doses. Ao selecionar uma quantidade maior que uma surge um novo campo de data, que é referente a próxima dose. Também há um campo de texto referente ao lote da vacina e, por fim, serão cadastradas as datas referentes a última aplicação e as datas das próximas doses a serem tomadas.

Figura 17 – Registro de vacina

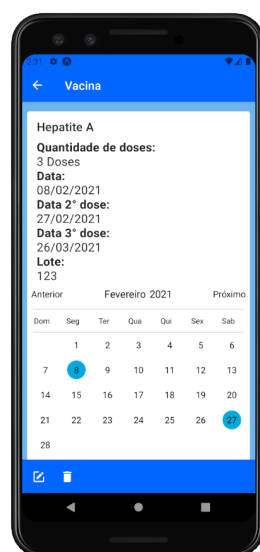


Fonte: Autor do trabalho (2021)

4.1.5 Tela de vacina

Após o cadastro de uma vacina, é possível selecionar uma vacina específica no quadro de vacinas e ser encaminhado para a tela onde serão exibidos todos os dados daquela vacina. Isso pode ser visto na [Figura 18](#). Na tela de vacina, é exibido o nome dessa vacina, a quantidade de doses, a data de aplicação da vacina, datas das próximas doses caso a vacina possua mais de uma dose, o lote da vacina e também é exibido um calendário onde é possível ver, marcadas em azul, as datas referentes a aplicação da vacina e suas próximas doses e, em cinza, o dia atual.

Figura 18 – Tela de vacina

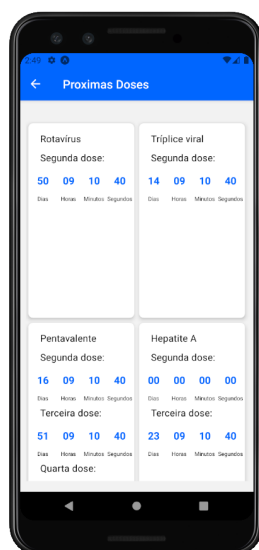


Fonte: Autor do trabalho (2021)

4.1.6 Tela de próximas doses

A tela de próximas doses que pode ser vista na [Figura 19](#), exibe para o usuário várias contagens regressivas tendo como data final as datas das próximas doses das vacinas, sendo que quando essas datas chegam ao fim é enviada uma notificação para o usuário o informando que chegou a data de tomar uma nova dose de uma vacina, depois que notificação é enviada a contagem some dessa dela. Quando o usuário toca em uma dessas contagens o usuário é encaminhado para tela específica daquela vacina onde ele pode ver as datas no calendário ou ver os demais dados daquela vacina, que pode ser vista na [Figura 18](#).

Figura 19 – Tela de próximas doses

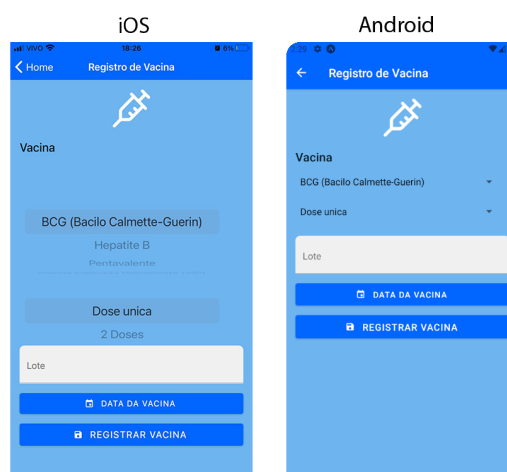


Fonte: Autor do trabalho (2021)

4.2 Comparação entre as versões de Android e iOS

Essa seção tem o foco em mostrar as diferenças entre as versões de Android e iOS. Como a aplicação é feita a partir de um único código em Javascript e JSX, as diferenças entre as versões serão sutis. Porém, como o React Native é capaz de fazer chamada de elementos nativos de cada plataforma, acaba ocorrendo algumas diferenças como pode ser visto na [Figura 20](#). A única diferença entre as versões ocorreu no formulário de cadastro de vacinas, onde o componente Picker do React Native tem uma aparência diferente no Android e no iOS.

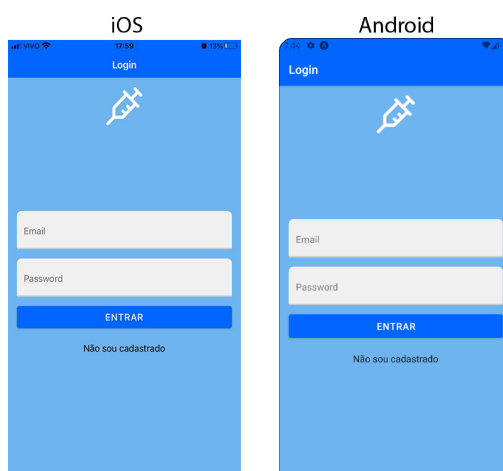
Figura 20 – Principal diferença entre a versão de Android e iOS



Fonte: Autor do trabalho (2021)

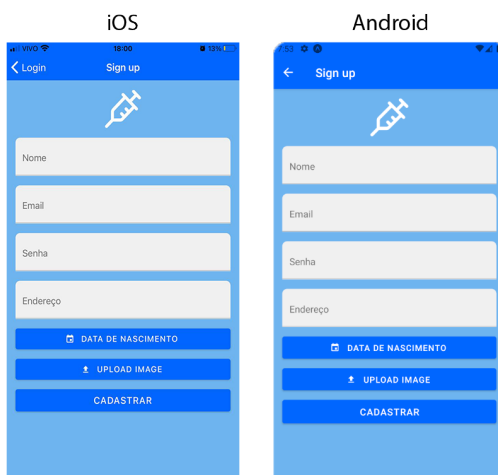
As imagens do iOS foram capturadas de um iPhone 7 Plus através do aplicativo Expo Go, enquanto as imagens do Android foram feitas através de um emulador rodando no Windows 10 que simula o celular Google Pixel 3. As figuras 21, 22, 23, 24 e 25 representam o restante das telas do aplicativo quando comparadas entre o iOS e o Android. É possível ver que não há uma diferença visual expressiva entre essas telas.

Figura 21 – Comparação do login



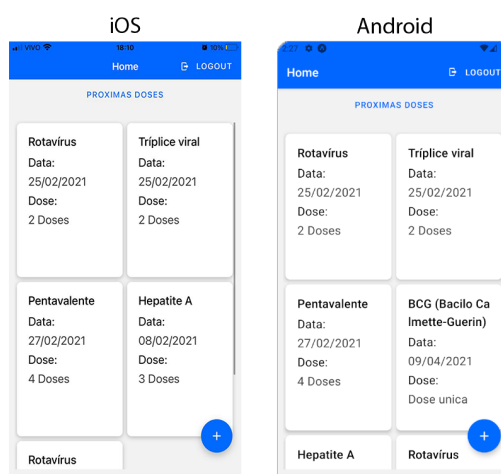
Fonte: Autor do trabalho (2021)

Figura 22 – Comparação do registro de novos usuários



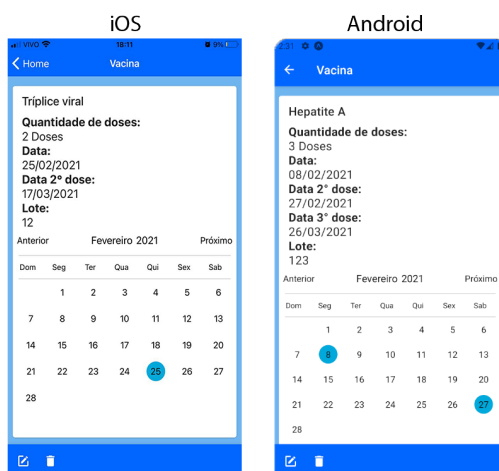
Fonte: Autor do trabalho (2021)

Figura 23 – Comparação da página principal



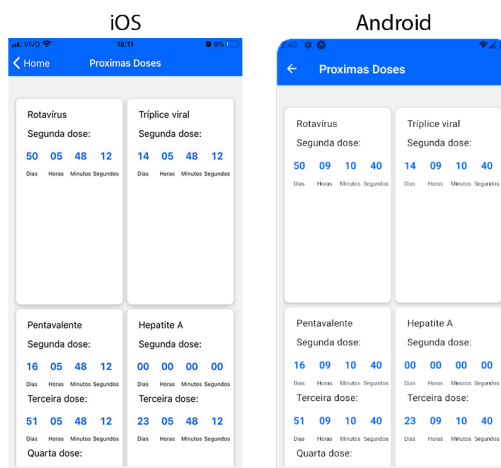
Fonte: Autor do trabalho (2021)

Figura 24 – Comparação da tela de vacina



Fonte: Autor do trabalho (2021)

Figura 25 – Comparação da tela próximas doses



Fonte: Autor do trabalho (2021)

4.3 Comparação entre o aplicativo desenvolvido e os aplicativos similares

Nesta seção, será comparado o aplicativo que foi desenvolvido durante o decorrer deste trabalho com os aplicativos que atendem necessidades semelhantes, que foram citados anteriormente na [seção 3.1](#). A comparação ocorre através da [Tabela 2](#).

Tabela 2 – Comparativo com aplicativos similares.

Características	Conecte SUS	Calendário de vacinação	Carteirinha de Criança	Aplicativo desenvolvido pelo autor do trabalho
Plataformas	iOS e Android	Android	Android	iOS e Android
Vincula os dados armazenados a uma conta?	Sim	Não	Não	Sim
Fornece ao usuário informações específicas de cada vacina?	Não	Sim	Sim	Não. Durante o desenvolvimento desse trabalho ainda não foi implementado porém será implementado em trabalhos futuros.
Apresenta um calendário com as datas das vacinas e próximas doses?	Não	Não	Não	Sim
Possibilita ao usuário gerenciar seus próprios dados?	Não	Não	Sim	Sim
Envia notificações ao usuários quando chegar a data da próxima aplicação de uma dose?	Não	Não	Não	Sim
Pode exportar vacinas como PDF ou outros formatos?	Não	Não	Sim	Não. Durante o desenvolvimento desse trabalho ainda não foi implementado. Porém, será implementado em trabalhos futuros.

Fonte: Elaborado pelo autor (2021)

Os tópicos escolhidos para comparação dos aplicativos similares com o aplicativo desenvolvido presentes na [Tabela 2](#) estão relacionados com a coleta de requisitos, presentes na [seção 3.2](#). Durante a coleta de requisitos foi observado quais os pontos essenciais que um aplicativo de saúde focado na gestão de vacinações precisa atender, mesmo tendo o feito toda a coleta de dados é importante destacar que na fase atual do aplicativo ele também não atende todos esses requisitos, como pode ser visto na tabela, porém ele existe um

planejamento para que ele possa atender esses requisitos, como pode ser visto na [seção 5.2](#).

As vantagens principais do aplicativo desenvolvido neste trabalho quando comparado aos outros aplicativos é a disponibilidade em múltiplas plataformas, possuir um sistema de notificações de próximas vacinas ou vacinas em atraso e permitir seus usuários fazer o gerenciamento de seus próprios dados, essas vantagens não são exclusivas do aplicativo, o aplicativo do ministério da saúde também é disponível para múltiplas plataformas e o aplicativo carteirinha de criança possibilita ao usuário fazer gerenciamento de seus dados.

As maiores desvantagens que o aplicativo possui são de funções que ainda não foram implementadas, porém essas funções estão sendo planejadas para serem implementadas futuramente, entre essas funções estão a capacidade de exportar o cartão de vacinas como PDF e fornecer informações e notícias específicas sobre saúde aos usuários.

4.4 Testes

Após a conclusão do desenvolvimento, foi iniciada a fase de testes. Para isso, foi escolhido o *framework* Firebase, que é uma ferramenta com várias funcionalidades que dão apoio ao desenvolvedor durante o processo de criação de um software. Para a execução dos testes, foi utilizada a infraestrutura do Firebase Test Lab. Ela possui um plano gratuito que permite ao usuário selecionar um dispositivo entre uma grande variedade de celulares. Nele é feito o teste Robo, que segundo o (FIREBASE, 2021), é um tipo de teste que explora a UI do aplicativo de modo sistemático, simulando o comportamento de um usuário. Esse teste também valida a correção de erros e testa a regressão.

De acordo com (DUSTIN; RASHKA; PAUL, 1999), o uso de testes automatizados em softwares se popularizou devido ao cronograma apertado que os desenvolvedores enfrentam no processo de desenvolvimento. Como o teste é uma necessidade fundamental para garantir a qualidade do software, que não pode ser negligenciado. Sendo assim, os desenvolvedores optam pelos testes automatizados por eles conseguirem evitar o uso de testes manuais.

Durante os testes feitos pelo Firebase Test Lab, foi utilizado o dispositivo Google Pixel 2 e o dispositivo Lenovo k520, onde o Pixel 2 estava utilizando a API 28 do Android e o Lenovo k520 utilizando a API 26. O teste do Pixel teve a duração de 5 minutos e 17 segundos e o do Lenovo foi de 5 minutos e 2 segundos como pode ser observado nas figuras [26](#) e [27](#). Os testes podem ter a duração máxima de 20 minutos.

Figura 26 – Teste Robo Lenovo k520, API 26



Fonte: Elaborado pelo autor (2021)

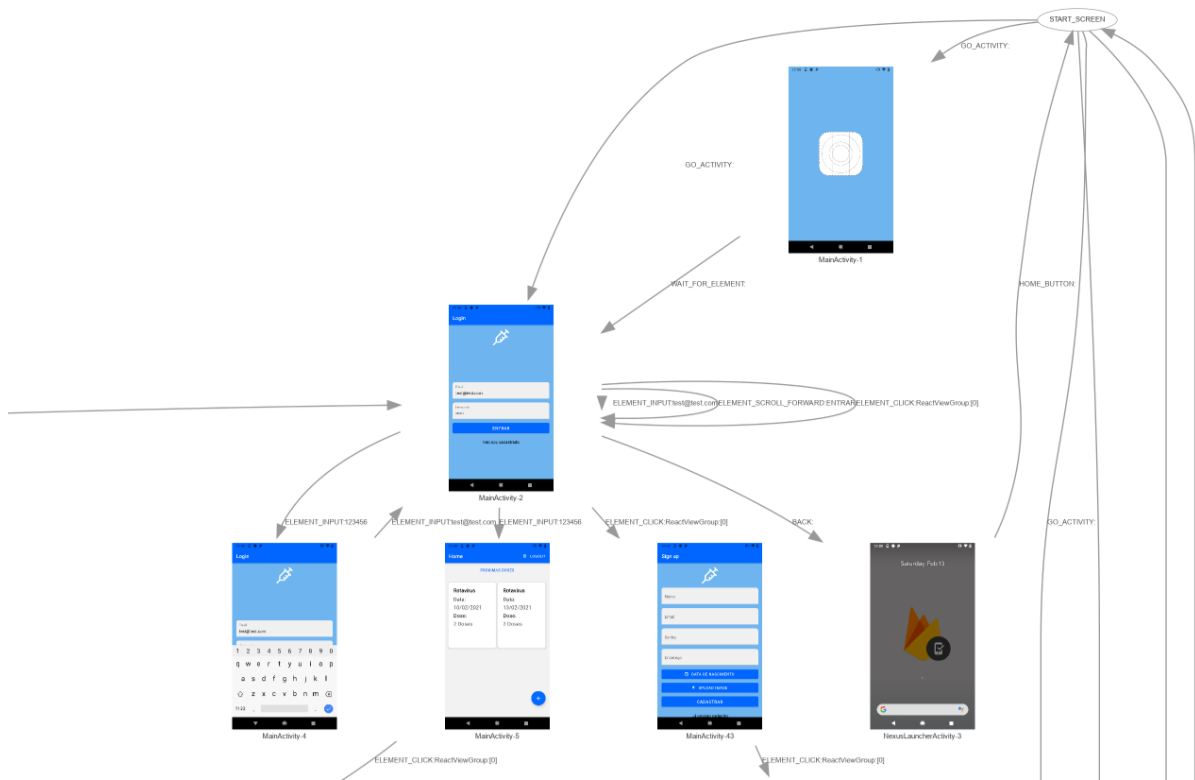
Figura 27 – Teste Robo Pixel 2, API 28



Fonte: Elaborado pelo autor (2021)

Na [Figura 28](#), é possível ver os primeiros níveis do gráfico de rastreamento (A imagem só mostra os primeiros níveis do gráfico devido ao seu tamanho muito grande, que possui mais de 18 níveis), O gráfico inteiro pode ser visto no apêndice na [seção A.1](#). Ele mostra as telas que o teste explorou de forma automática e a ordem dessa exploração. Esse tipo de teste pode verificar se todas as telas foram alcançadas e se o comportamento aconteceu conforme era esperado. No teste realizado, foi possível observar que todas as telas do aplicativo puderam ser alcançadas pelo teste, sem a ocorrência de nenhum comportamento estranho.

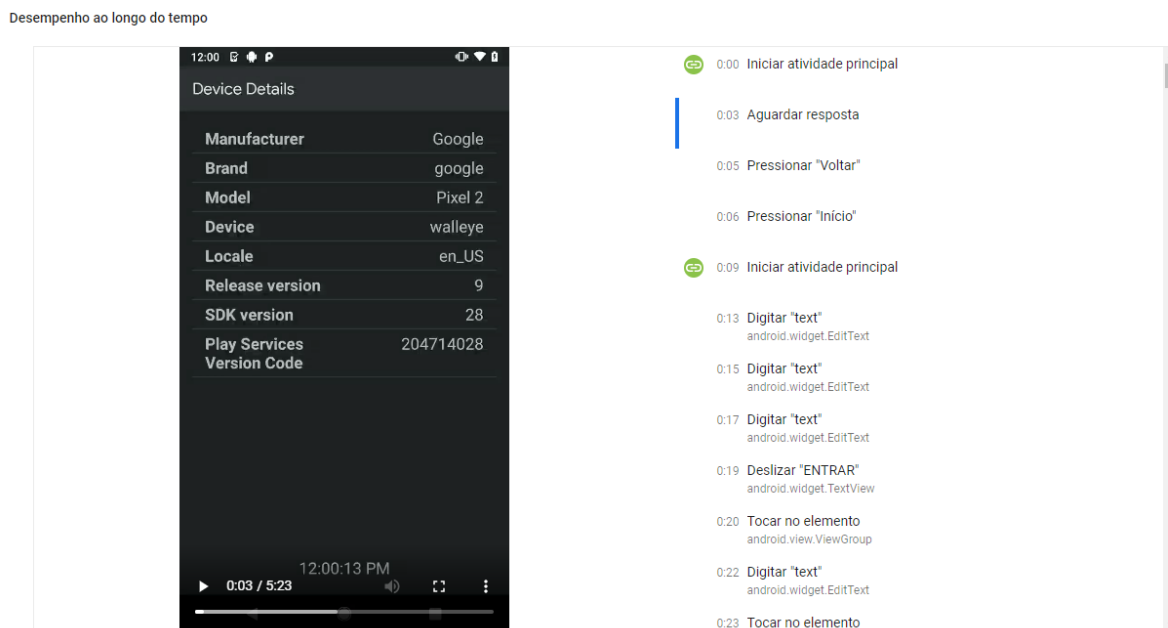
Figura 28 – Gráfico de rastreamento



Fonte: Elaborado pelo autor (2021)

Além de ser possível verificar todas as telas que foram acessadas pelo teste, com o gráfico de rastreamento, também é possível ver o que foi feito no teste. Para isso, o Firebase Test Lab grava um vídeo do processo de teste, e ainda exibe por escrito o que foi feito e com qual componente do Android aconteceu a interação. É possível ver isso na Figura 29.

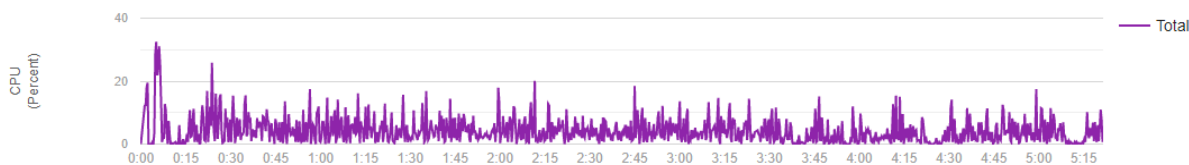
Figura 29 – Desempenho ao longo do tempo



Fonte: Elaborado pelo autor (2021)

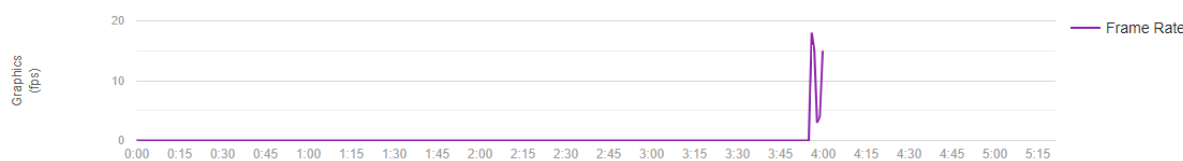
O Teste Robo também monitora o desempenho do dispositivo durante o uso do aplicativo. Como é possível ver nas figuras 30, 31, 32, 33, são exibidos os gráficos com os desempenhos do processador, gráficos (Taxa de quadros por segundos), memória e uso de rede. A partir dos gráficos é possível ver que o aplicativo possui um desempenho aceitável onde a etapa que teve maior uso do processador no Pixel 2, foi em 00:05 segundos e teve um pico de 40% de uso da *CPU*.

Figura 30 – Desempenho do processador



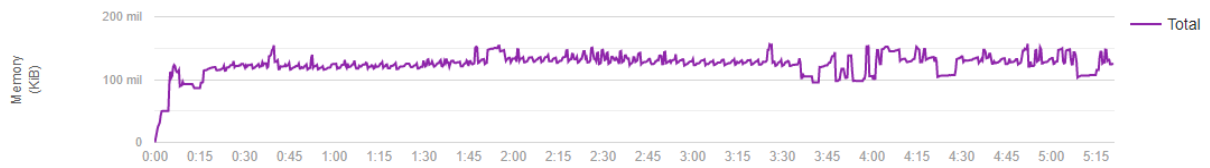
Fonte: Elaborado pelo autor (2021)

Figura 31 – Desempenho gráfico



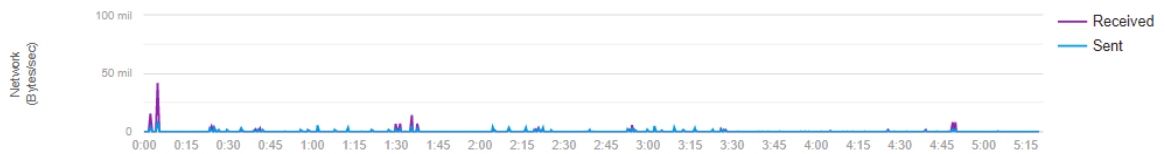
Fonte: Elaborado pelo autor (2021)

Figura 32 – Desempenho de uso de memória



Fonte: Elaborado pelo autor (2021)

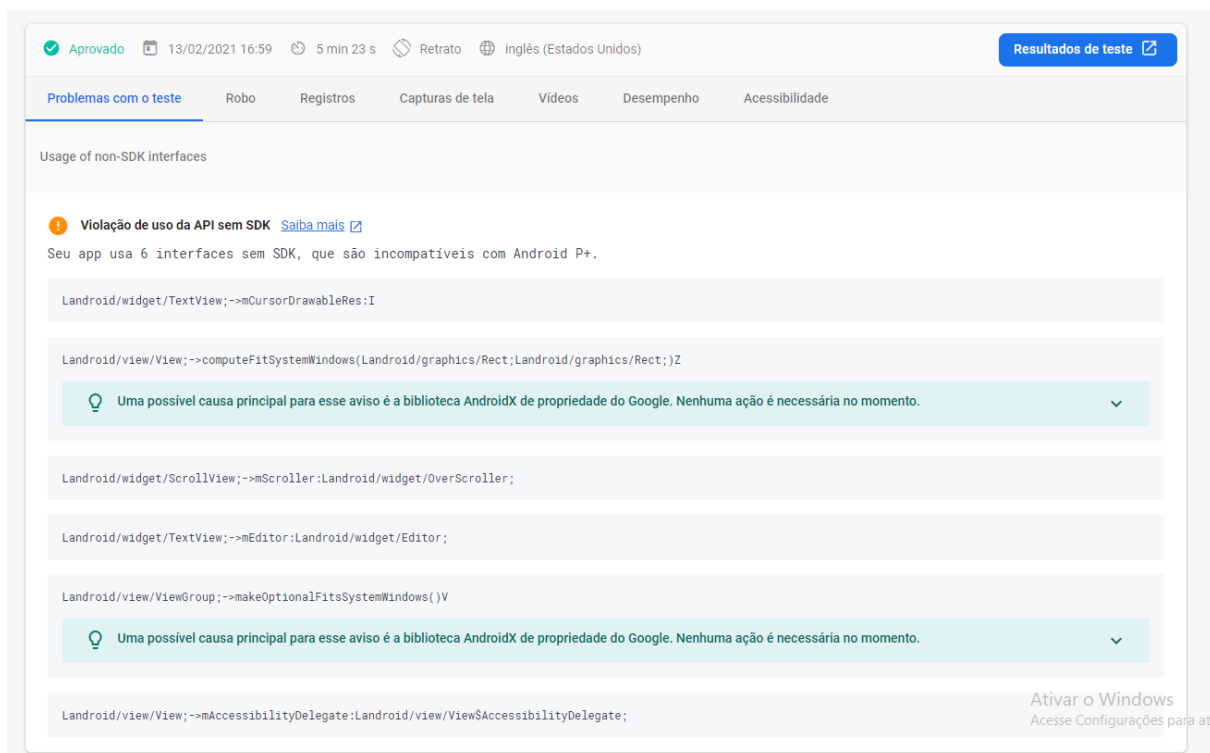
Figura 33 – Uso de banda



Fonte: Elaborado pelo autor (2021)

Após a execução dos testes, ambos os testes realizados tanto no Lenovo k520 quanto no Pixel 2 foram aprovados. Entretanto, a versão do Pixel 2 trouxe uma mensagem de aviso, que alerta sobre o uso de SDK incompatível com API 28 utilizada no Pixel 2. Mas o próprio Firebase Test Lab informa que nenhuma alteração é necessária no momento, o que pode significar alguma incompatibilidade com a nova API do Android, que deve ser resolvida em atualizações futuras do React Native. A mensagem de problemas com o teste pode ser vista na [Figura 34](#).

Figura 34 – Problemas com o teste



Fonte: Elaborado pelo autor (2021)

5 Considerações Finais

Como foi proposto nas etapas iniciais desse trabalho, foi feita a criação de um aplicativo multiplataforma com a capacidade de auxiliar seus usuários a fazer o gerenciamento de suas vacinas. Com o objetivo de também ajudar o usuário a se lembrar de suas próximas doses, o sistema também usa de notificações para deixar o usuário sempre atento às próximas doses de suas vacinas. O cadastro de vacinas também funciona conforme um cartão de vacinas onde o usuário pode ver em uma lista todas as suas vacinas cadastradas.

Como era esperado do desenvolvimento, foi possível alcançar o objetivo de desenvolver um aplicativo funcional tanto para Android quanto para iOS, sendo também utilizado um banco de dados não relacional com o armazenamento em nuvem. Onde os dados dos usuários ficam vinculados a sua conta, e mesmo que ele desinstale o aplicativo ou mude de dispositivos, esses dados ainda podem ser recuperados, quando o usuário se logar novamente no aplicativo em outros dispositivos ou tendo o aplicativo novamente instalado.

Atendendo o objetivo proposto de atingir mais usuários e diminuir o tempo de desenvolvimento em plataformas diferentes, foi escolhido o desenvolvimento multiplataforma, onde utilizando o React Native foi possível a criação de um aplicativo para Android e para iOS através de um único código em Javascript. Tendo o mesmo comportamento nos aplicativos de ambas as plataformas e a mesma interface de usuário. O aplicativo também foi submetido a testes automatizados através do Firebase Test Lab, e teve um resultado satisfatório, tanto em performance quanto no processo de encontrar falhas.

Como importância da aplicabilidade, é destacada a sua capacidade de estar mandando notificações aos usuários quando estiver no dia de uma vacina, quando uma vacina estiver atrasada. Portanto, tendo o impacto de deixar seus usuários atentos e os auxiliar a se manterem imunizados. O aplicativo também tem outras funções que permitem ao usuário ter uma maior disponibilidade de seu cartão de vacina, onde ele pode também o manter disponível em qualquer dispositivo móvel com acesso a Internet.

A respeito das limitações do aplicativo, atualmente ele tem sua capacidade de armazenamento limitada no banco de dados, por utilizar um plano gratuito do MongoDB. Porém, essa escalabilidade pode ser feita de um modo fácil, somente alterando o plano de armazenamento do MongoDB. Outra limitação é a respeito das funções que ainda não foram implementadas, que serão implementadas em trabalhos futuros, como exportar o cartão virtual como PDF e adicionar eventos no dispositivo móvel do usuário. O aplicativo também tem a limitação de não ser integrado com os sistemas de saúde do governo, onde cabe aos usuários a inserção de seus dados.

5.1 Objetivos alcançados

O objetivo geral do trabalho é desenvolver um aplicativo multiplataforma capaz de auxiliar seus usuários no processo de gestão de vacinações. Esse objetivo foi alcançado, como mostrado no capítulo de resultados. O aplicativo desenvolvido é orientado para as plataformas Android e iOS, além de possuir diversas funções que permitem aos seus usuários fazer o cadastro de suas vacinas e estarem recebendo notificações das datas das próximas vacinas.

Com base nos objetivos específicos, foi possível fazer todo processo de coleta de requisitos com base nos dados coletados de aplicativos similares e do cartão de vacinação físico. Com esses dados, foi possível desenvolver todo diagrama de casos de uso, o diagrama entidade-relacionamento e os protótipos de tela. Ainda como objetivo específico, foi possível realizar os testes e validações que foram feitos por meio do *framework* do Firebase.

5.2 Trabalhos Futuros

Mesmo chegando ao fim do que foi proposto por esse trabalho, muito do processo de desenvolvimento ainda precisa ser feito para que o aplicativo possa atender seus usuários da melhor forma possível e além de auxiliar os usuários a gerenciarem suas vacinações ter a capacidade de conscientizar as pessoas sobre a importância da vacinação. Nos itens abaixo estão itens que serão trabalhados e melhorados no desenvolvimento futuro deste aplicativo.

- Trazer informações específicas de cada vacina.
- Indicar ao usuário os postos de vacinação mais próximos de sua localidade.
- Trazer um *feed* de notícias e informações do ministério da saúde sobre vacinas.
- Criar uma tela de perfil de usuário, onde ele possa ver e gerenciar seus dados além de trocar sua foto de perfil.
- Criar uma opção de exclusão da conta onde o usuário possa excluir todos os seus dados armazenados no banco de dados do sistema.
- Possibilitar o cadastro de próximas vacinações como eventos na agenda do dispositivo do usuário.

Referências

- ARELLANO, M. A. Preservação de documentos digitais. *Ciência da Informação*, SciELO Brasil, v. 33, n. 2, p. 15–27, 2004. Citado na página 15.
- ASTAH. *Premier Diagramming, Modeling Software & Tools | Astah*. 2020. Disponível em: <<https://www.astah.net/>>. (Acesso em: 12/11/2020). Citado na página 25.
- BRASIL. *Programa Nacional de Imunizações (PNI)*. 2015. <<http://www.blog.saude.gov.br/index.php/entenda-o-sus/50027-programa-nacional-de-imunizacoes-pni>>. (Acessado em 09/02/2021). Citado na página 19.
- BRASIL. *Saiba mais sobre a importância da vacinação oferecida pelo SUS*. 2017. <<http://www.blog.saude.gov.br/index.php/promocao-da-saude/52477-saiba-mais-sobre-a-importancia-da-vacinacao-oferecida-pelo-sus#:~:text=O%20Programa%20Nacional%20de%20Imuniza%C3%A7%C3%A3o,e%20bact%C3%A9rias%20que%20provocam%20doen%C3%A7as.>> (Acesso em: 31/01/2021). Citado na página 14.
- BRASIL. *Calendario2020.atualizado.pdf*. 2020. <<https://www.saude.gov.br/files/imunizacao/calendario/Calendario2020.atualizado.pdf>>. (Acessado em 09/02/2021). Citado na página 20.
- CETIC. *Pesquisa sobre o uso das Tecnologias de Informação e Comunicação nos Estabelecimentos de Saúde Brasileiros - TIC Saúde 2019*. 2020. <https://www.cetic.br/media/docs/publicacoes/2/20201123084414/tic_saude_2019_livro_eletronico.pdf>. (Acessado em 24/04/2021). Citado na página 16.
- CHODOROW, K. *MongoDB: the definitive guide: powerful and scalable data storage*. [S.l.]: "O'Reilly Media, Inc.", 2013. Citado na página 33.
- CORRAL, L.; JANES, A.; REMENCIUS, T. Potential advantages and disadvantages of multiplatform development frameworks—a vision on mobile environments. *Procedia Computer Science*, Elsevier, v. 10, p. 1202–1207, 2012. Citado 2 vezes nas páginas 14 e 16.
- DOMINGUES, C. M. A. S.; TEIXEIRA, A. M. d. S. Coberturas vacinais e doenças imunopreveníveis no Brasil no período 1982-2012: avanços e desafios do programa nacional de imunizações. *Epidemiologia e Serviços de Saúde*, Coordenação-Geral de Desenvolvimento da Epidemiologia em Serviços/Secretaria . . . , v. 22, n. 1, p. 9–27, 2013. Citado na página 19.
- DUSTIN, E.; RASHKA, J.; PAUL, J. *Automated software testing: introduction, management, and performance*. [S.l.]: Addison-Wesley Professional, 1999. Citado na página 56.
- EISENMAN, B. *Learning react native: Building native mobile apps with JavaScript*. [S.l.]: "O'Reilly Media, Inc.", 2015. Citado na página 22.
- EL-KASSAS, W. S. et al. Taxonomy of cross-platform mobile applications development approaches. *Ain Shams Engineering Journal*, Elsevier, v. 8, n. 2, p. 163–190, 2017. Citado 3 vezes nas páginas 14, 21 e 37.

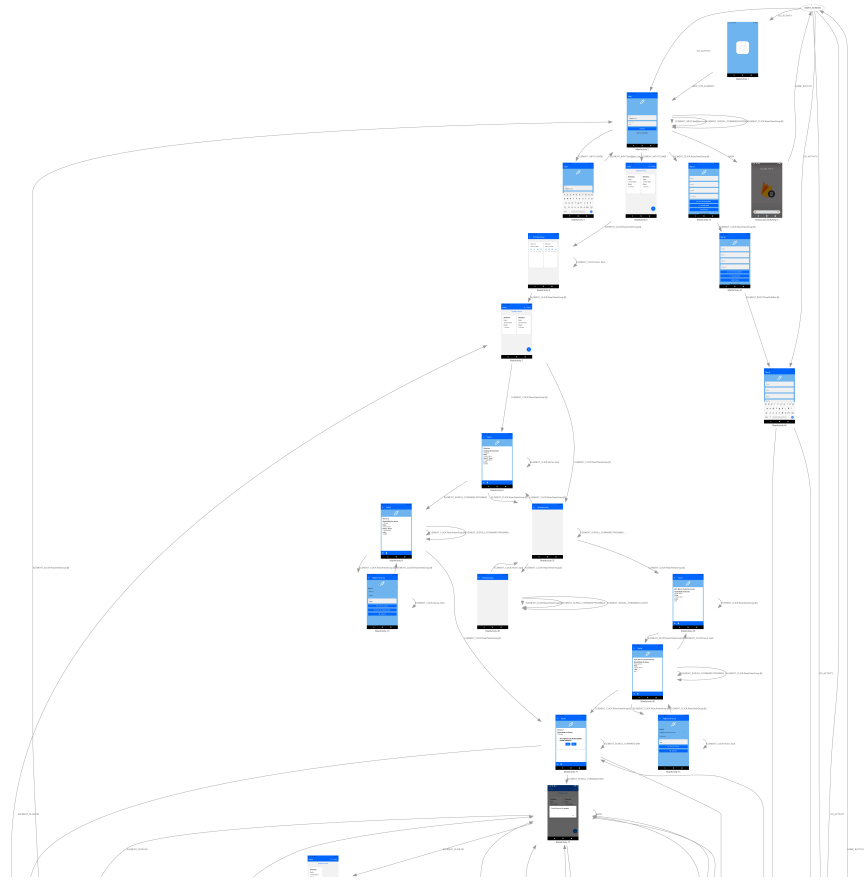
- EXPO. *Expo — Dashboard*. 2021. <<https://expo.io/>>. (Acessado em 21/02/2021). Citado 2 vezes nas páginas 42 e 43.
- FIGMA. *Figma*. 2020. Disponível em: <<https://www.figma.com/>>. (Acesso em: 12/11/2020). Citado na página 31.
- FIREBASE. *Primeiros passos com testes Robo*. 2021. <<https://firebase.google.com/docs/test-lab/android/robo-ux-test?hl=pt-br>>. (Acessado em 04/03/2021). Citado na página 56.
- FORCE, C. P. S. T. et al. Recommendation for use of immunization information systems to increase vaccination rates. *Journal of Public Health Management and Practice*, LWW, v. 21, n. 3, p. 249–252, 2015. Citado 2 vezes nas páginas 14 e 19.
- GROOM, H. et al. Immunization information systems to increase vaccination rates: a community guide systematic review. *Journal of Public Health Management and Practice*, LWW, v. 21, n. 3, p. 227–248, 2015. Citado 2 vezes nas páginas 15 e 19.
- GSMA. *The Mobile Economy 2019*. 2019. <<https://data.gsminelligence.com/api-web/v2/research-file-download?id=39256194&file=2712-250219-ME-Global.pdf>>. (Acessado em 26/03/2021). Citado na página 16.
- IAN, S. Engenharia de software. *9a. edição, São Paulo: Pearson Prentice Hall*, 2011. Citado 4 vezes nas páginas 23, 24, 25 e 31.
- KENNETH, C. L.; LAUDON, J. P. Sistemas de informação gerenciais. *Editora Person. São Paulo*, 2011. Citado na página 18.
- MARIN, H. de F. Sistemas de informação em saúde: considerações gerais. *Journal of Health Informatics*, v. 2, n. 1, 2010. Citado na página 18.
- MONGODB. *MongoDB*. 2021. Disponível em: <<https://www.mongodb.com/>>. (Acessado em 18/02/2021). Citado 2 vezes nas páginas 35 e 36.
- MONGODB. *MongoDB Atlas — MongoDB Atlas*. 2021. <<https://docs.atlas.mongodb.com/>>. (Acessado em 18/02/2021). Citado na página 35.
- MYSQL. *MySQL :: MySQL Workbench*. 2020. Disponível em: <<https://www.mysql.com/products/workbench/>>. (Acesso em: 12/11/2020). Citado na página 32.
- NATIVE, R. *Core Components and Native Components · React Native*. 2021. <<https://reactnative.dev/docs/intro-react-native-components>>. (Acessado em 22/02/2021). Citado 2 vezes nas páginas 37 e 38.
- ONU. *OMS e Unicef alertam para riscos de queda nas taxas de imunização de crianças / ONU News*. 2020. <<https://news.un.org/pt/story/2020/11/1731962>>. (Acesso em: 01/02/2021). Citado na página 15.
- SILVA, B. S. et al. Estudo de avaliabilidade do sistema de informação do programa nacional de imunização. *Revista Brasileira de Enfermagem*, SciELO Brasil, v. 71, p. 615–624, 2018. Citado na página 18.

Apêndices

APÊNDICE A – Materiais elaborados pelo autor

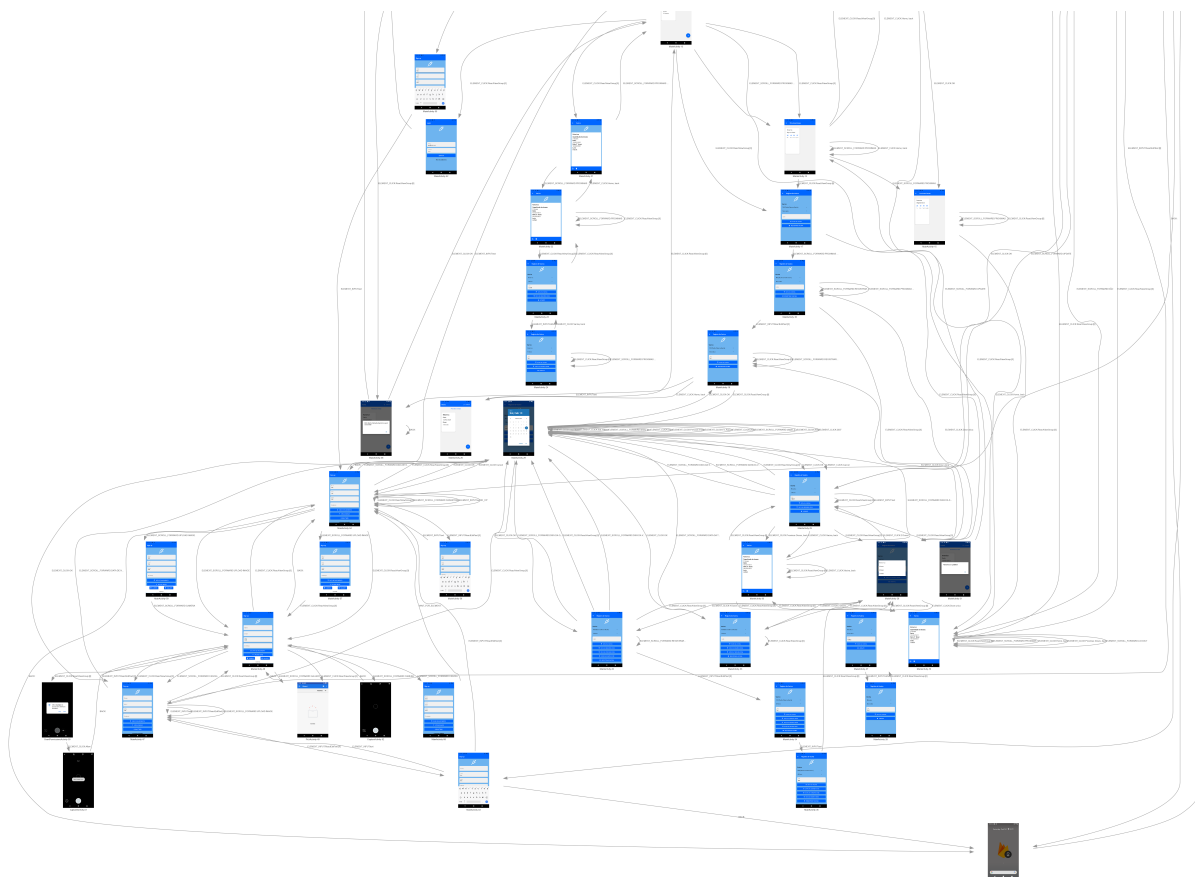
A.1 Gráfico de rastreamento

Figura 35 – Gráfico de rastreamento, Parte 1



Fonte: Elaborado pelo autor (2021)

Figura 36 – Gráfico de rastreamento, Parte 2



Fonte: Elaborado pelo autor (2021)

Link para melhor visualização: <<https://bit.ly/3fjMkhM>>