



**UEPB**

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

**EMERSON SOUSA PEREIRA**

**VACINA CAMPINA APP: APLICATIVO PARA CONTROLE DE VACINAS EM UM  
AMBIENTE FAMILIAR NA CIDADE DE CAMPINA GRANDE**

**CAMPINA GRANDE  
2021**

EMERSON SOUSA PEREIRA

**VACINACAMPINA: APLICATIVO PARA CONTROLE DE VACINAS EM UM  
AMBIENTE FAMILIAR NA CIDADE DE CAMPINA GRANDE)**

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.

P436v Pereira, Emerson Sousa.  
Vacina Campina *app* [manuscrito] : aplicativo para controle de vacinas em um ambiente familiar na cidade de Campina Grande / Emerson Sousa Pereira. - 2021.  
68 p. : il. colorido.

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. Aplicativos. 2. Engenharia de software. 3. Vacinação. I.  
Título

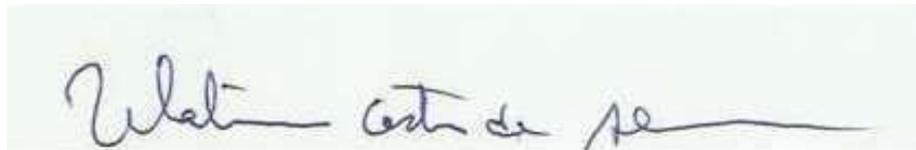
21. ed. CDD 005.3

**EMERSON SOUSA PEREIRA**

**Vacina Campina App: Aplicativo para Controle de Vacinas em um Ambiente Familiar na Cidade de Campina Grande**

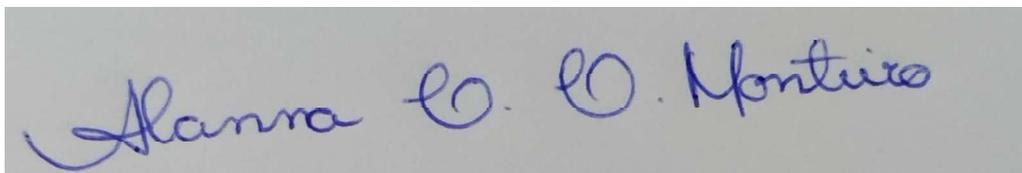
Trabalho de Conclusão de Curso de Graduação em Ciência da Computação da Universidade Estadual da Paraíba, como requisito à obtenção do título de Bacharel em Ciência da Computação.

Aprovada em 01 de Outubro de 2021.



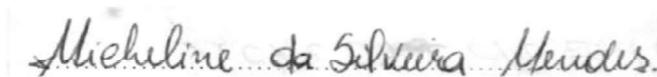
---

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



---

Prof. Msc. Alanna Camylla Coelho Monteiro (DC - 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**

Agradeço primeiramente a Deus que é minha fortaleza e o meu refúgio por me proporcionar perseverança durante o trajeto da minha vida acadêmica.

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.

Aos meus pais, Jailma Sousa Pereira e Erinaldo Gomes Pereira, por estarem sempre me apoiando e me incentivando durante toda esta longa jornada.

Aos meus irmãos, Everton Sousa Pereira e Elton Sousa Pereira por toda a preocupação e Carinho.

À minha Noiva, e futura esposa, Cinthya Raquel dos Santos Lima, pelo seu apoio, amor, carinho, por ser paciente comigo. Por estar sempre me incentivando e acreditando no meu potencial.

## RESUMO

A vacinação é hoje o meio mais avançado no controle de doenças infecciosas, porém na maior parte do país o seu controle é feito através de cadernetas feitas de papel, material que está sujeito a degradação e extravio além de tornar o gerenciamento uma atividade complexa em famílias onde somente um indivíduo tem que realizar controle vacinal de todos os membros. Neste contexto o presente trabalho teve como objetivo a implementação do VacinaCampina, um aplicativo para informar e conscientizar as pessoas sobre a importância da vacina e realizar gerenciamento da caderneta de vacina em um ambiente familiar na cidade de Campina Grande. O aplicativo foi desenvolvido para plataforma Android utilizando a linguagem Java, o *RealTime* para gravação dos dados, o conjunto de ferramentas do *Firebase* para controle de acesso, armazenamento de imagens e pôr fim a utilização do GIT para realizar o controle de versões.

**Palavras-Chave:** Aplicativo. Vacina. Engenharia de software.

## **ABSTRACT**

Vaccination is currently the most advanced means of controlling infectious diseases, but in most parts of the country its control is done through booklets made of paper, a material that is subject to degradation and loss, in addition to making management a complex activity in families where only one individual has to carry out vaccination control of all members. In this context, this work aims to implement VacinaCampina, an application to inform and raise awareness about the importance of the vaccine and manage the vaccine booklet in a family environment in the city of Campina Grande. The application was developed for the Android platform using the Java language, RealTime for data recording, the Firebase toolset for access control, image storage and putting an end to the use of GIT to perform version control.

**Keywords:** Application. Vaccine.

## LISTA DE ILUSTRAÇÕES

Figura 1 –	Diversidade de preço por sistema operacional.....	19
Figura 2 –	Camadas da Arquitetura Android.....	20
Figura 3 –	Fluxo de execução Java.....	22
Figura 4 –	Calendário Nacional de Vacinação.....	24
Figura 5 –	Diagrama de Caso de uso.....	26
Figura 6 –	Manter Usuário.....	27
Figura 7 –	Manter Parente.....	29
Figura 8 –	Visualizar Cartão de Vacina.....	30
Figura 9 –	Registrar Dose.....	31
Figura 10 –	Imprimir Cartão de Vacina.....	32
Figura 11 –	Imprimir Cartão de Vacina.....	33
Figura 12 –	Visualizar Informações de Vacinas.....	34
Figura 13 –	Diagrama de Classes.....	36
Figura 14 –	Diversidade de preço por sistema operacional.....	37
Figura 15 –	Fluxo de execução Java.....	39
Figura 16 –	Ferramentas de construção de aplicativos Firebase.....	40
Figura 17 –	Android Studio.....	41
Figura 18 –	Algoritmo de login aplicativo Vacina Campina.....	44
Figura 19 –	Template de verificação de e-mail e recuperação de senha.....	45
Figura 20 –	Estrutura de banco de dados aplicativos Vacina Campina.....	46
Figura 21 –	Consulta de vacinas por nome.....	46
Figura 22 –	Algoritmo de envio de imagem para o Firebase Storage.....	47
Figura 23 –	Console Firebase Storage.....	48
Figura 24 –	Tela de apresentação.....	49
Figura 25 –	Tela de redirecionamento.....	49
Figura 26 –	Tela de Login.....	50
Figura 27 –	Fluxo Tela de Cadastro.....	51
Figura 28 –	E-mail de Verificação.....	51
Figura 29 –	Tela de Recuperação de Senha.....	52
Figura 30 –	E-mail de recuperação de senha.....	52
Figura 31 –	Fluxo de Redefinição de Senha.....	53

Figura 32 – Listagem de Parentes.....	54
Figura 33 – Cadastro de Parente.....	55
Figura 34 – Cartão de Vacina.....	56
Figura 35 – Cadastro de Dose.....	56
Figura 36 – – Listagem e Busca de Vacinas.....	57
Figura 37 – Tela Informativa Sobre Vacina.....	58
Figura 38 – Tela de Postos de Vacinação.....	59
Figura 39 – Tela de Configuração de Usuário.....	60

## LISTA DE QUADROS

Quadro 1 –	Cadastrar Usuário.....	28
Quadro 2 –	Editar Usuário.....	28
Quadro 3 –	Manter Parente.....	29
Quadro 4 –	Visualizar Cartão de Vacina.....	30
Quadro 5 –	Registrar Dose.....	31
Quadro 6 –	Imprimir Cartão de Vacina.....	33
Quadro 7 –	Buscar Postos de Vacinação.....	34
Quadro 8 –	Visualizar Informações de Vacinas.....	35
Quadro 9 –	Requisitos Mínimos Android Studio.....	38
Quadro 10 –	Comandos GIT utilizados.....	42

## LISTA DE GRÁFICOS

Gráfico 1 – Utilização de sistemas operacionais no mundo.....	28
---	----

## LISTA DE ABREVIATURAS E SIGLAS

PNAD	Pesquisa Nacional por Amostra de Domicílios
API	<i>Application Programming Interface</i>
HAL	<i>Hardware abstraction layer</i>
DEX	<i>Dalvik Executable</i>
VM	<i>Virtual Machines</i>
SO	Sistema Operacional
SMS	<i>Short Message Service</i>
TCP	<i>Transmission Control Protocol</i>
IP	<i>Internet Protocol</i>
HTTP	<i>Hypertext Transfer Protocol</i>
FTP	<i>File Transfer Protocol</i>
JVM	<i>Java Virtual Machine</i>
UML	<i>Unified Modeling Language</i>
PNI	Programa Nacional de imunização
IDE	<i>Integrated Development Environment</i>
ADT	<i>Eclipse Android Development</i>
BAAS	<i>Backend as a Service</i>
SDK	<i>Software Development Kit</i>

## SUMÁRIO

1	INTRODUÇÃO .....	13
2	JUSTIFICATIVA .....	15
3	OBJETIVOS.....	16
3.1	Objetivo Geral .....	16
3.2	Objetivos Específicos.....	16
4	FUNDAMENTAÇÃO TEORICA.....	17
4.1	Android SO.....	17
4.1.1	<i>Arquitetura Android</i> .....	19
4.2	Java.....	21
5	METODOLOGIA.....	24
5.1	Levantamento de Requisitos.....	24
5.2	Modelagem e Documentação da Aplicação.....	24
5.2.1	<i>Diagrama de caso de uso</i> .....	25
5.2.2	<i>Diagrama de Classes</i> .....	35
5.3	Tecnologias e Materiais Envolvidos.....	37
5.3.1	<i>Android Studio</i> .....	37
5.3.2	<i>Firestore</i> .....	38
5.3.3	<i>Controle de Versão e GIT</i> .....	39
5.3.4	<i>Hardwares Utilizados</i> .....	42
6	RESULTADOS E DISCUSSÕES .....	43
6.1	Back-End e Banco de dados.....	43
6.1.1	<i>Segurança e Autenticação</i> .....	43
6.1.2	<i>Segurança e Autenticação</i> .....	45
6.2	Funcionalidades do Aplicativo.....	48
7	CONCLUSÃO .....	61
	REFERÊNCIAS .....	62
	APÊNDICE A – REQUISITOS FUNCIONAIS DO SISTEMA.....	64
	APÊNDICE B – REQUISITOS NÃO FUNCIONAIS DO SISTEMA.....	67
	APÊNDICE C – CARTÃO DE VACINA PARA IMPRESSÃO.....	68

## 1 INTRODUÇÃO

Desde o início da sociedade moderna, as doenças infecciosas encontraram um ambiente propício a proliferação nos grandes centros e esses patógenos foram responsáveis por dizimar milhares de vidas durante séculos. As doenças infecciosas têm sido uma proeminente e importante causa da mortalidade através da história da humanidade e sempre teve um papel chave no controle populacional em nosso planeta (BONNANI, 1998).

A vacinação foi a resposta encontrada pela humanidade para combater e se proteger contra pandemias e epidemias, um instrumento de imunização de extrema importância para nossa sociedade, que contribuiu significativamente para o desenvolvimento da humanidade, reduzindo a taxa de mortalidade e contribuindo para o aumento da expectativa de vida. Com exceção de água potável, nenhuma outra ação, nem mesmo antibióticos, teve tão grande impacto na redução da mortalidade e no crescimento da população (PLOTKIN, ORENSTEIN, OFFIT; 2004).

No Brasil, a vacinação teve excelentes avanços no último século. O sucesso da Campanha de Erradicação da Varíola (CEV) fortaleceu, dentro do Ministério da Saúde (MS), um movimento que defendia maiores investimentos no controle de doenças infecciosas preveníveis por imunização. A conclusão do programa, iniciado em 1962, e a criação do Programa Nacional de Imunizações (PNI) em 1973, marcaram de forma histórica a recente política de imunizações no país (TEMPORÃO, 2003, p. 602).

Apesar de todo o avanço que o Programa Nacional de Imunização trouxe para o país alguns pontos ainda são falhos, um deles é o fato de o controle da vacinação ser feito através do registro em cartões feitos de papel conhecidos como caderneta ou cartão de vacinação, juntamente com o calendário nacional de vacinação, ambos criados e adotados em 1997 pelo Ministério da Saúde em todo o território nacional.

O controle é simples e rápido onde a pessoa recebe a vacina e o aplicador carimba a caderneta de papel, entretanto o papel degrada rapidamente e facilmente pode ser perdido ou extraviado além de tornar difícil o controle de doses em um ambiente onde uma única pessoa é responsável por gerenciar a caderneta de todos os seus familiares. Para facilitar a administração das cadernetas foi pensado na

implementação de um aplicativo para aparelho celular, visto que o mesmo tem ampla utilização do público em geral.

É notório o avanço da tecnologia e inovação nas mais diversas áreas, e a popularização dos celulares com acesso à internet faz com que os cidadãos no geral sejam beneficiados por todo esse avanço.

Os resultados da Pesquisa Nacional por Amostra de Domicílios (PNAD), de 2017, identificou que 93,2% dos domicílios brasileiros utilizavam smartphone, destes, 97% o tinham como principal meio de acesso à Internet, de forma que a mudança de comportamento na forma de acesso à internet culminou na substituição de computadores, tablets, televisões inteligentes e outros equipamentos pelas versões móveis, sobretudo smartphones (INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA, 2018).

Em concordância com o contexto apresentado acima, o projeto proposto tem como objetivo o uso de um aplicativo para celular onde seja possível realizar o controle da caderneta de vacinação, auxiliando chefes de família a manter o controle de vacinação de todos em seu ambiente familiar, podendo mesmo ter a informação de todos em qualquer smartphone com acesso à internet.

É importante ressaltar que este sistema não tem como objetivo a substituição da carteirinha de vacinação, mas oferecer às pessoas uma alternativa para mesma além da opção de não precisar se revacinar caso haja extravio.

## 2 JUSTIFICATIVA

O presente trabalho foi construído de forma a suprir a necessidade tanto de controle quanto de informação sobre vacinas, no decorrer dos últimos anos vem diminuindo a quantidade de pessoas que buscam a vacinação. De acordo com o Ministério da Saúde, sete das oito vacinas obrigatórias na infância tiveram a cobertura menor do que o esperado em 2018. Um dos principais fatores causadores são as dúvidas na hora de tomar a vacina, conhecido como hesitação vacinal, o fenômeno costuma acontecer pela falta de informação ou, até mesmo, por informações falsas (DOMINGUES et al., 2019).

Outro fator importante na construção do trabalho é o gerenciamento familiar de vacinas, o fato das cadernetas serem de papel tornam esse trabalho um pouco complexo, a perda ou extravio desse documento podem gerar um certo desconforto. Ao perder a caderneta a pessoa deve se dirigir ao posto de saúde para tirar uma segunda via. Porém, de acordo com pesquisa realizada em 2019, 17% dos postos de saúde do nordeste brasileiro não tem acesso à internet e quando expandimos esses dados para o país inteiro este número sobe para 18% (CETIC, 2020). O que em vários casos impossibilita o registro das doses no Sistema de Informações do Programa Nacional de Informações (SI-PNI), tornando a caderneta de papel o único meio de manter um histórico de vacinação, devido a este fato pode ocorrer o extravio das informações.

Caso seja confirmado o extravio das informações, tornando impossível recuperar esse histórico, um novo esquema vacinal poderá ser iniciado, então o profissional da unidade de saúde fará uma avaliação considerando a faixa etária do indivíduo e as vacinas recomendadas para a sua faixa etária, e procederá com a vacinação e agendamento de possíveis doses subsequentes (LUIZA BARUFI, 2020). O que conseqüentemente acaba gerando custos desnecessários para o governo, além de causar um certo transtorno ao indivíduo, já que o mesmo estará sujeito a sofrer com os desconfortos das reações que a maioria das vacinas causam.

### **3 OBJETIVOS**

#### **3.1 Objetivo Geral**

O objetivo deste trabalho é construir uma aplicação que ajude no gerenciamento dos cartões de vacina dentro do ambiente familiar na cidade de Campina Grande – PB.

#### **3.2 Objetivos Específicos**

- Fornecer uma alternativa aos tradicionais cartões de vacinação feitos de papel;
- Informar e conscientizar sobre a importância e necessidade do uso das vacinas;
- Exibir dados e localização sobre postos de vacinação de forma simples e dinâmica;

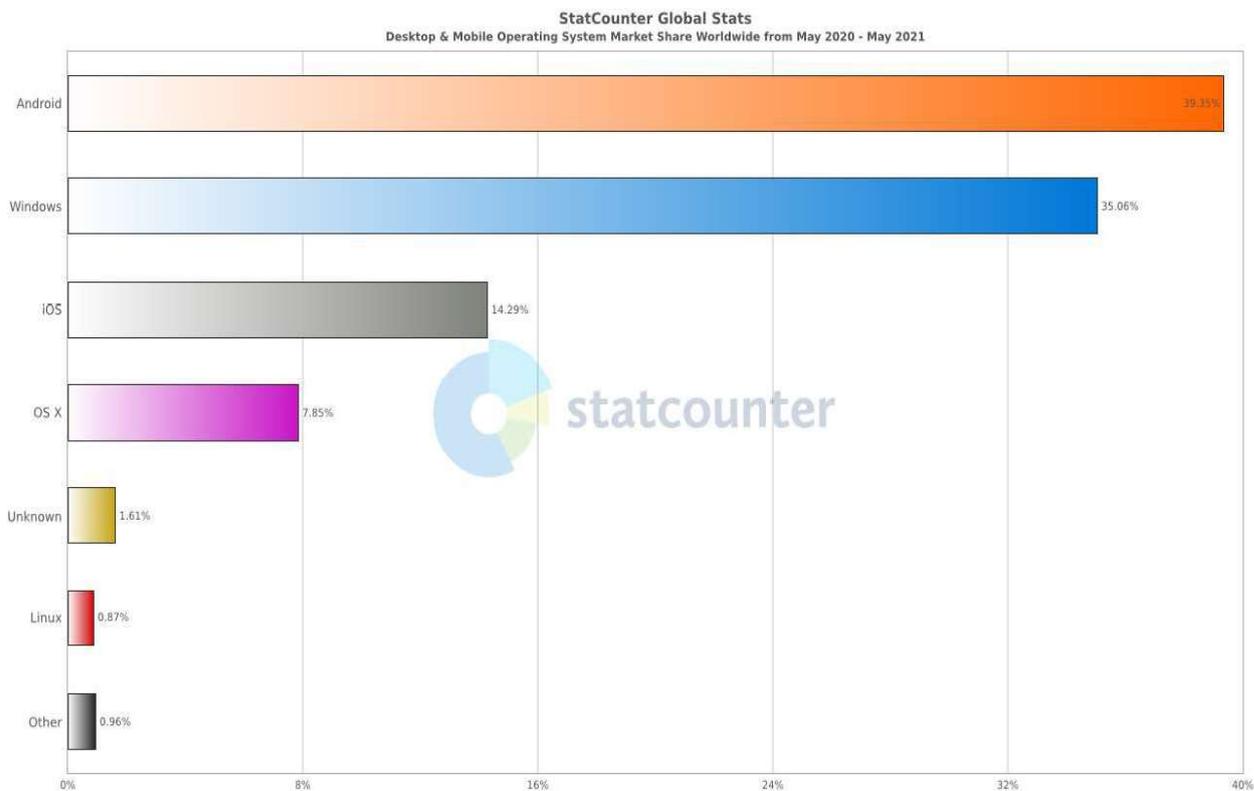
## 4 FUNDAMENTAÇÃO TEÓRICA

Para a execução do projeto, se fez necessário levantar estudos sobre uma série de conceitos e tecnologias nesta seção serão apresentadas as principais.

### **4.1 Android SO**

O Android é um sistema operacional para aplicativo móvel de código aberto baseado no kernel do Linux e atualmente é mantido pelo Google. Criado em 2003 na cidade de Palo Alto na Califórnia e foi desenvolvido Andy Rubin, Rich Miner, Nick Sears e Chris White com o nome de Android Inc, tinha como ideia original ser um sistema para câmeras digitais, porém ao perceberem que o mercado tinha limitações migraram o foco do projeto para dispositivos mobile. Comprada pelo Google em 2005, desde então o Android conquistou espaço e se tornou o sistema mais utilizado no mundo e no Brasil não é diferente, de acordo com Moura e Camargo (2020) em estudo divulgado pela consultoria global Bain & Company em parceria com o Google.

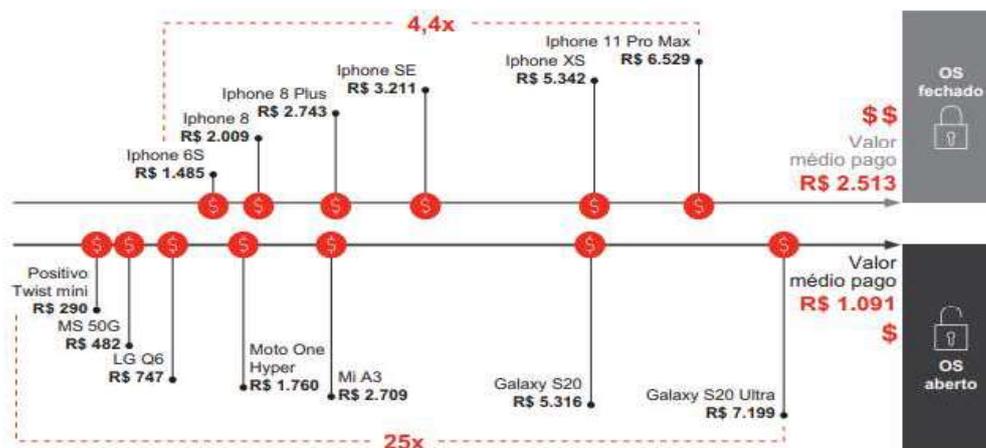
O sistema operacional Android está presente em mais de 90% dos celulares brasileiros, ou seja, a cada dez aparelhos, nove possuem o sistema operacional do Google, até mesmo em comparação com sistemas operacionais de outras plataformas podemos ver que o Android é o sistema operacional mais utilizado em todo o mundo, como podemos observar no gráfico 1.

**Gráfico 1 – Utilização de sistemas operacionais no mundo.**

**Fonte:** <https://gs.statcounter.com/os-market-share#monthly-202005-202105-bar,2021>.

O Android contribui de forma expressiva no aumento do acesso à internet no Brasil, no período de cinco anos que antecedem 2020, 24 milhões de brasileiros foram introduzidos à internet por meio de um dispositivo Android (MOURA; CAMARGO, 2020). O Android busca trazer acessibilidade a internet, construído para funcionar em hardwares de baixo custo reflete diretamente no custo de acesso à tecnologia de smartphones no Brasil como podemos ver na Figura 1.

**Figura 1 – Diversidade de preço por sistema operacional**



Fonte: Bain, 2020.

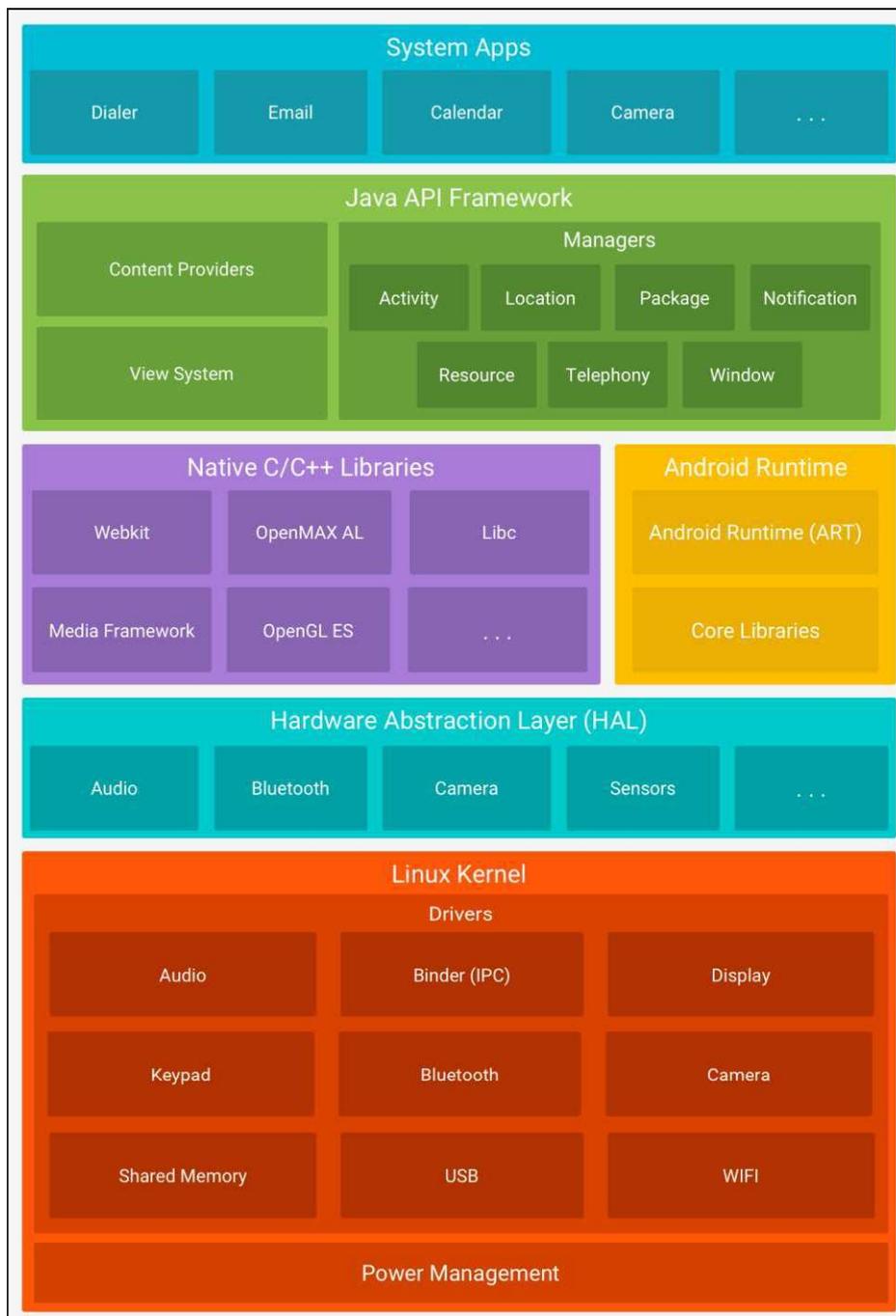
#### 4.1.1 Arquitetura Android

A arquitetura da plataforma Android pode ser dividida em seis partes sendo elas: Kernel Linux, Camada de abstração de bibliotecas, *Runtime* (ambiente de execução), framework Java API e por fim as aplicações do sistema (Figura 2).

Segundo Ableson *et al.* (2012) o **kernel Linux** fornece uma camada fundamental de abstração de hardware, proporciona vários serviços essenciais como gerenciamento de processos, memória e sistema de arquivos. É considerada a base do sistema operacional Android.

A **camada de abstração de hardware** também conhecida por HAL, é a responsável por fornecer interfaces padrões que expõem as capacidades de hardware para a camada de framework Java API. De acordo com a documentação oficial do Android (2021), a HAL é constituída em módulos de biblioteca, que implementam uma interface para um tipo específico de componente de hardware, como o módulo de câmera ou Bluetooth por exemplo.

**Figura 2 – Camadas da Arquitetura Android**



**Fonte:** <https://developer.android.com/guide/platform?hl=pt-br>.

O **Android Runtime** é projetado para executar múltiplas 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. Esses arquivos são correspondentes lógicos dos códigos de byte Java, mas permitem ao Android executar seus aplicativos na sua própria máquina virtual

(VM), que é uma plataforma aberta na qual o Google, e a comunidade de código-fonte aberto, pode evoluir conforme necessário (ABLESON *et al.*, 2012).

**A camada de bibliotecas** disponibiliza bibliotecas nativas programadas em C e C++ que são utilizadas pelos principais serviços do Android como o Android Runtime e pela camada de abstração de hardware (HAL), além fornecer tecnologia de navegador do WebKit, suporte a banco de dados via SQLite, suporte a gráfico avançado e suporte a áudio e vídeo com o OpenCORE.

O **Framework da Java API** é um conjunto completo de recursos do SO que fornece serviços sendo eles:

- **Sistema de visualização** – Extremamente útil para programar a Interface de usuário de aplicativos, com listas, grades, caixas de texto e botões.
- **Gerenciador de notificação** – permite que os aplicativos exibam alertas de forma personalizada na barra de status.
- **Gerenciador de atividade** – Que faz o controle do ciclo de vida dos aplicativos e fornece uma pilha de navegação inversa.
- **Provedores de conteúdo** – que permite que os aplicativos possam acessar dados de outros aplicativos.

Por fim, temos a **camada de aplicações de sistemas**, que disponibiliza um conjunto de ferramentas também conhecidos como aplicativos padrão do sistema, sendo eles para e-mail, envio de SMS, calendário, contanto entre outros. Os aplicativos do sistema atuam como aplicativos para os usuários e disponibilizam os principais recursos que os desenvolvedores podem acessar pelos próprios aplicativos (ANDROID, 2021).

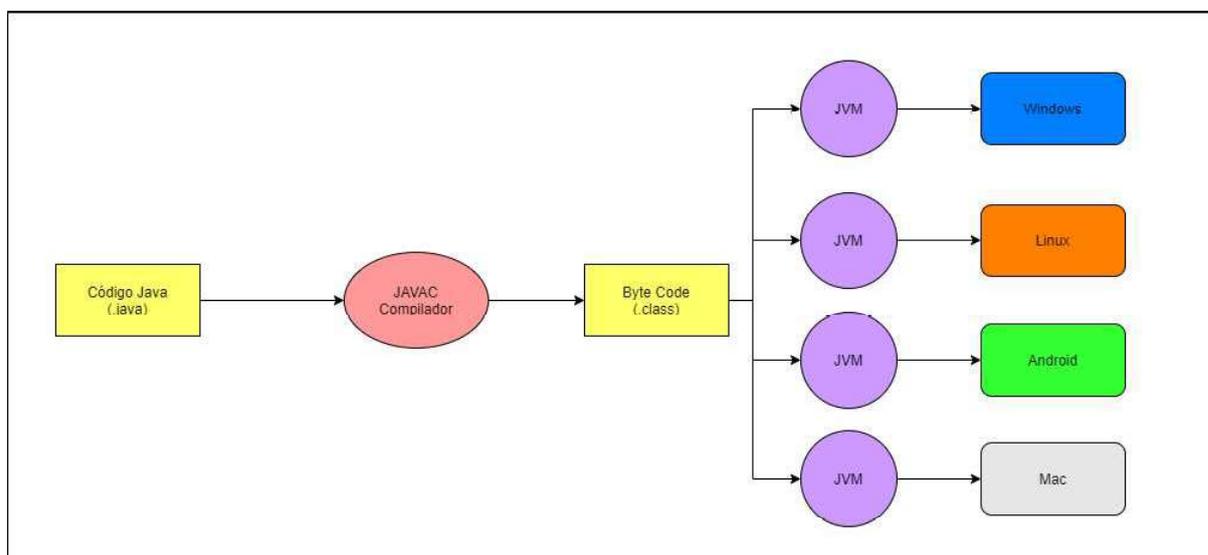
## 4.2 Java

Java é uma linguagem de programação multiplataforma orientada a objetos de propósito geral. Criada em 1990 pela Sun Microsystems, para ser executada nas mais diversas plataformas de hardware visando à Internet (FEDELI, 2003). Em 2008 a Oracle Corporation compra a Sun Microsystems e se torna a empresa responsável por manter e evoluir o ecossistema Java. A linguagem foi projetada almejando os seguintes objetivos:

- **Orientação a objetos** - baseado na composição e interação entre diversas unidades chamadas de objetos.
- **Portabilidade** - Independência de plataforma - "escreva uma vez, execute em qualquer lugar";
- **Recursos de Rede** - Possui extensa biblioteca de rotinas que facilitam a cooperação com protocolos TCP/IP, como HTTP e FTP;
- **Segurança** - Pode executar programas via rede com restrições de execução.

O Java faz uso de um conceito diferente, ao invés de gerar um código binário diferente para cada plataforma, é gerado um binário que pode ser executado em qualquer plataforma, dentro de uma máquina virtual Java conhecido popularmente como JVM (*Java Virtual Machine*). Ao compilar o código Java, são gerados arquivos objetos chamados de *byte-codes*, e estes podem ser executados por meio da JVM para cada tipo de plataforma (FEDELI, 2003). Na Figura 3 podemos observar o fluxo da execução de um código Java.

**Figura 3 – Fluxo de execução Java**



**Fonte:** Elaborada pelo Autor, 2021.

Além de todos os benefícios que o Java traz ao ser executado em uma máquina virtual, podem-se destacar outras vantagens apresentadas pela linguagem:

- Facilidades de Internacionalização – Tem suporte nativo a caracteres Unicode;
- Especificação simples, tanto da linguagem como do "ambiente" de execução (JVM);
- É distribuída com um vasto conjunto de bibliotecas (ou APIs);
- Possui recursos facilitadores para criação de programas distribuídos e multitarefa;
- Gerenciamento automático de memória através do *Java Garbage Collection*;
- Carga Dinâmica de Código - Programas em Java são formados por uma coleção de classes armazenadas de forma independentemente e que podem ser carregadas somente no momento de utilização.

## 5 METODOLOGIA

Neste capítulo é descrita a metodologia utilizada para o desenvolvimento do aplicativo VacinaCampina.

### 5.1 Levantamento de Requisitos

Os requisitos de um sistema são as especificações do que o sistema deve fazer, os serviços que deve disponibilizar e as restrições a seu funcionamento (SOMMERVILLE, 2011). O levantamento de requisitos o projeto foi feito através do estudo do Programa Nacional de imunizações, tendo como principal base o Calendário Nacional de Vacinação (Figura 4), além da necessidade local de uma aplicação que gerenciasse o cartão de todas as pessoas que convivem no mesmo ambiente familiar, através destas informações foram elaborados os requisitos funcionais descritos no APÊNDICE A, e não-funcionais, descritos no APÊNDICE B.

**Figura 4 - Calendário Nacional de Vacinação**

CALENDÁRIO NACIONAL DE VACINAÇÃO															
Grupo Alvo	Idade	BCG	Hepatite B	Pentax/DTP	VIP/VOP	Pneumocócica 10V (conjugada)	Rotavírus Humano	Meningocócica C (conjugada)	Febre Amarela	Hepatite A	Tríplice Viral	Tetra Viral*	HPV	Dupla Adulto	dTpa**
Crianças	Ao nascer	Dose única	Dose ao nascer												
	2 meses			1ª dose	1ª dose (com VIP)	1ª dose	1ª dose								
	3 meses							1ª dose							
	4 meses			2ª dose	2ª dose (com VIP)	2ª dose	2ª dose								
	5 meses							2ª dose							
	6 meses			3ª dose	3ª dose (com VIP)										
	9 meses								Uma dose						
	12 meses					Reforço		Reforço			1ª dose				
	15 meses			1ª reforço (com DTP)	1ª reforço (com VOP)					Uma dose		Uma dose			
	4 anos			2ª reforço (com DTP)	2ª reforço (com VOP)				Reforço						
9 anos															
Adolescente	10 a 19 anos		3 doses (a depender da situação vacinal)						Uma dose e um reforço, a depender da situação vacinal		2 doses		2 doses (9 a 13 anos)	Reforço a cada 10 anos	
Adulto	20 a 59 anos		3 doses (a depender da situação vacinal)						Uma dose e um reforço, a depender da situação vacinal		1 dose (até 49 anos)			Reforço a cada 10 anos	
Idoso	60 anos ou mais		3 doses (a depender da situação vacinal)						Uma dose e um reforço, a depender da situação vacinal					Reforço a cada 10 anos	
Gestante			3 doses (a depender da situação vacinal)											3 doses (a depender da situação vacinal)	Uma dose a cada gestação entre a 27ª e a 36ª semana

\*A vacina tetra viral corresponde à segunda dose da tríplice viral e à dose da vacina varicela.  
 \*\* A vacina dTpa também será oferecida para profissionais de saúde que atuam em maternidade e em unidade de internação neonatal (UTI/UCI convencional e UCI Canguru) atendendo recém-nascidos e crianças menores de 1 ano de idade.  
 Para informações adicionais recomenda-se consultar a Instrução Normativa do Calendário Nacional de Vacinação disponível no seguinte endereço: [www.saude.gov.br](http://www.saude.gov.br)

Fonte: Ministério da Saúde, 2019.

### 5.2 Modelagem e Documentação da Aplicação

Segundo Sommerville (2011), a modelagem de um sistema é definida como o processo de desenvolvimento de modelos abstratos de um sistema, em que cada modelo apresenta uma visão ou perspectiva, diferente do sistema.

Para modelar e documentar a aplicação em alto nível, foi utilizado a Linguagem de Modelagem Unificada (UML), com o objetivo de representar o sistema de forma padronizada além de facilitar a compreensão e implementação do sistema, onde foram construídos os diagramas de caso de uso que é um diagrama comportamental e o diagrama de classes que é um diagrama estrutural.

### **5.2.1 Diagrama de caso de uso**

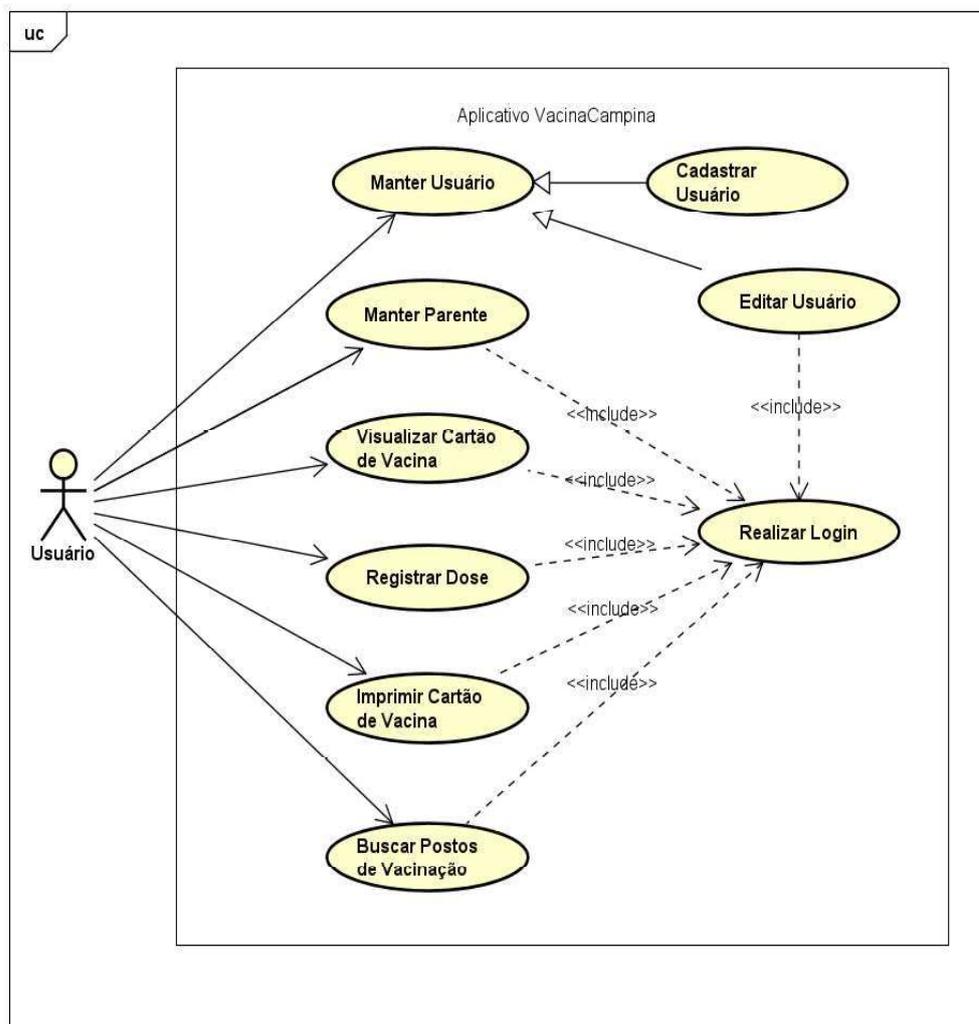
Com o objetivo de organizar os comportamentos de um sistema, o diagrama de caso de uso exibe um conjunto de atores e funcionalidades e como eles se relacionam dentro da aplicação. Na sua forma mais simplificada, um caso de uso identifica os atores envolvidos em uma interação e dá nome ao tipo de interação (SOMMERVILLE, 2011). O diagrama pode ter suas informações complementadas por uma descrição em forma de texto, o conjunto de casos de uso representa todas as possíveis interações que serão descritas nos requisitos de sistema.

Para criar um diagrama de caso de uso (Figura 5), fazemos uso das seguintes formas de representação:

- **Atores:** Representado por um boneco humano na forma de palitos
- **Classe de Iteração:** Representados por uma elipse.
- **Linhas:** Realizam a ligação entre os atores e as iterações, podem conter flechas na pontas para indicar a direção da iteração.

Para a diagramação do sistema, foi construído um diagrama geral que mostra todas as iterações e posteriormente cada iteração foi separada e especificada de forma única para que cada cenário fosse tratado individualmente.

**Figura 5 – Diagrama de Caso de uso**



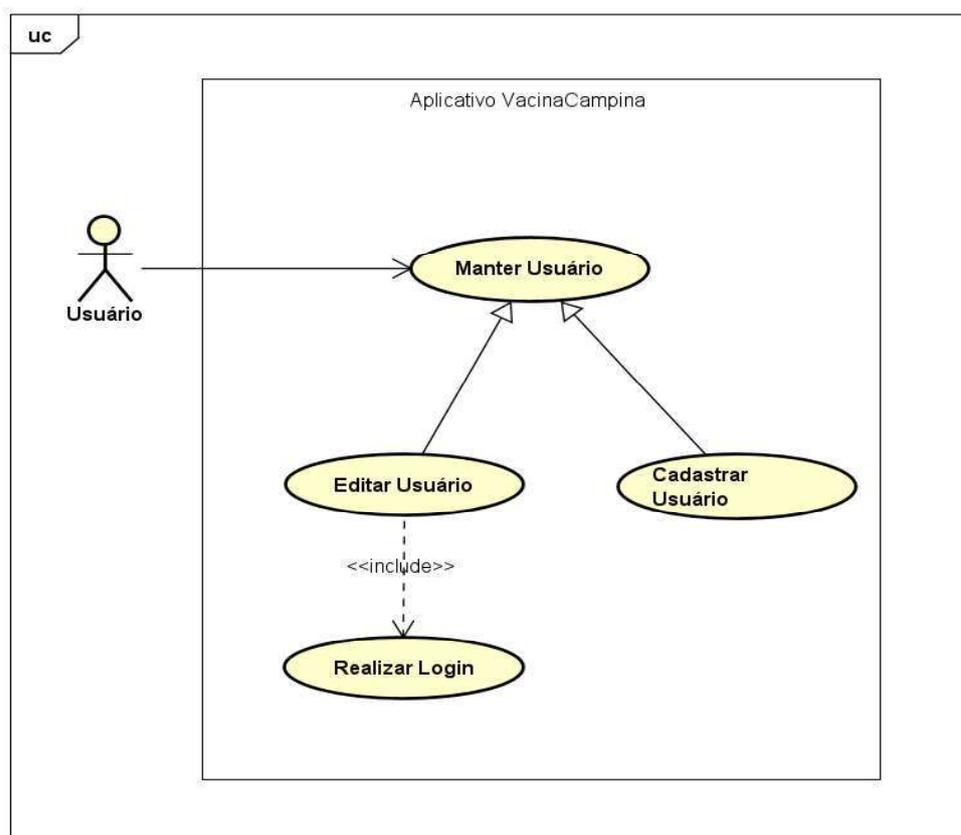
Fonte: Elaborada pelo autor, 2020.

O Diagrama de caso de uso do aplicativo é composto por apenas um ator que é o próprio usuário do sistema que pode executar 6 casos:

- **Manter Usuário:** O ator poderá cadastrar e editar do seu próprio usuário.
- **Manter Parente:** Permite que o ator realize cadastro e edição e remoções dos seus parentes no sistema.
- **Visualizar Cartão de Vacina:** Permite que o ator visualize tanto o seu próprio cartão de vacina como o de seus parentes.
- **Registrar Dose:** O ator pode registrar uma dose da vacina no seu próprio cartão ou no cartão dos parentes cadastrados

- **Imprimir Cartão de Vacina:** Permite que o ator imprima o seu próprio cartão ou o de seus parentes
- **Buscar Postos de Vacinação:** Possibilita que o ator realize busca por postos de saúde próximos na cidade de Campina Grande

**Figura 6 – Manter Usuário**



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

O caso de uso Manter o usuário (Figura 6) permite que usuário possa realizar o próprio cadastro e é um pré-requisito para o que o mesmo tenha acesso as funcionalidades do aplicativo, ao realizar o cadastro será enviado um e-mail de confirmação, logo após o usuário realizar este passo poderá acessar o fluxo de login, já logado na aplicação o usuário terá acesso a um menu onde uma das opções irá direcionar para tela de informações da conta onde o mesmo poderá realizar alteração nos dados cadastrais. As ações do usuário e as respostas do sistema estão descritas nos Quadros 1 e 2.

**Quadro 1 – Cadastrar Usuário**

CDU001 – Cadastrar Usuário	
Interessados e interesses	Usuário do Aplicativo
Pré-condições	Não existe
Ação do Ator	Resposta do Sistema
Fluxo Principal	
Realiza o Cadastro do usuário	Envia um e-mail de confirmação de cadastro e exibe uma mensagem de sucesso.
Fluxos alternativos	
Não preencher corretamente o cadastro	Emitir mensagem de validação

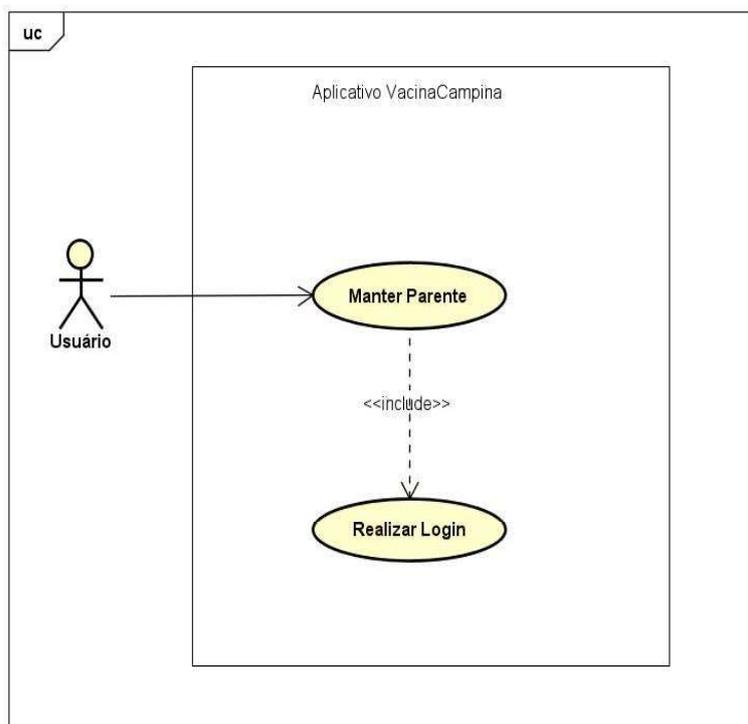
Fonte: Elaborada pelo autor, 2020.

**Quadro 2 – Editar Usuário**

CDU002 – Editar Usuário	
Interessados e interesses:	Usuário do Aplicativo
Pré-condições:	Esta logado
Ação do Ator	Resposta do Sistema
Fluxo Principal	
Realiza edição do usuário	Exibe uma mensagem de sucesso
Fluxos alternativos	
Deixar algum campo vazio	Emitir mensagem de validação

Fonte: Elaborada pelo autor, 2020.

**Figura 7 – Manter Parente**



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

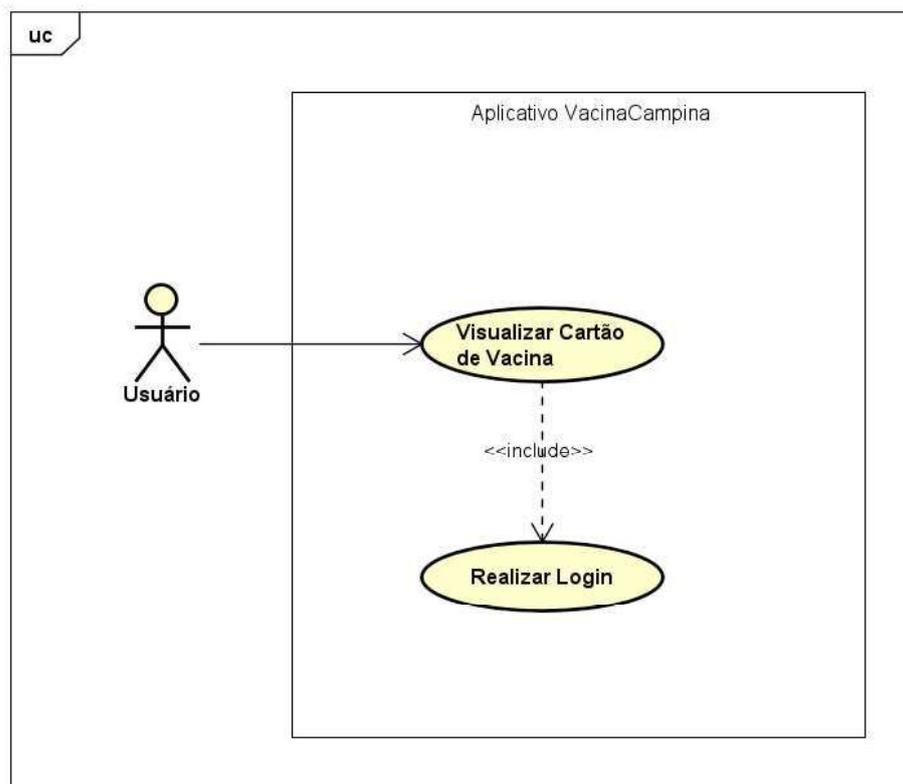
O caso de uso Manter Parente (Figura 7) representa o fluxo de cadastrar um membro da família, mediante o login o usuário terá acesso a tela de parentes onde será possível realizar o cadastro de parentes, o cadastro poderá ser feito apenas com as informações de nome e grau de parentesco, após o cadastro o usuário poderá gerenciar o cartão de vacina do mesmo. O Quadro 3 descreve as ações do ator e as respostas do sistema.

**Quadro 3 – Manter Parente**

CDU003 – Editar Usuário	
Interessados e interesses:	Usuário do Aplicativo
Pré-condições:	Está autenticado
Ação do Ator	Resposta do Sistema
Fluxo Principal	
Realiza o cadastro de um parente	Exibe uma mensagem de sucesso, e a lista de parentes cadastrados
Fluxos alternativos	
Deixar algum campo vazio	Emite mensagem de validação

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

**Figura 8 – Visualizar Cartão de Vacina**



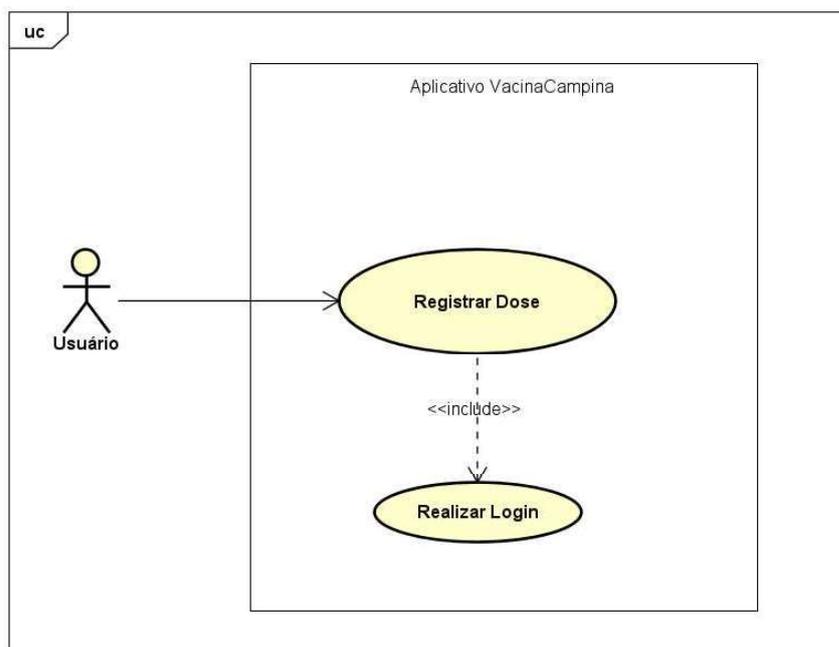
**Fonte:** Elaborada pelo autor, 2020.

No caso de uso Visualizar Cartão de Vacina (Figura 8) a partir da opção cartão o usuário poderá selecionar a si mesmo ou a um dos parentes sendo em seguida direcionado para tela de visualizar o cartão de vacina onde estará disponível as informações de datas e doses tomadas. O quadro 4 descreve as ações do usuário e as respectivas respostas do sistema.

**Quadro 4 – Visualizar Cartão de Vacina**

CDU004 – Visualizar Cartão de Vacina	
Interessados e interesses:	Usuário do Aplicativo
Pré-condições:	Está autenticado
Ação do Ator	Resposta do Sistema
Fluxo Principal	
Seleciona um parente ou a si mesmo na listagem	Exibe o cartão de vacina
Fluxos alternativos	
Não existe	

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

**Figura 9 – Registrar Dose**

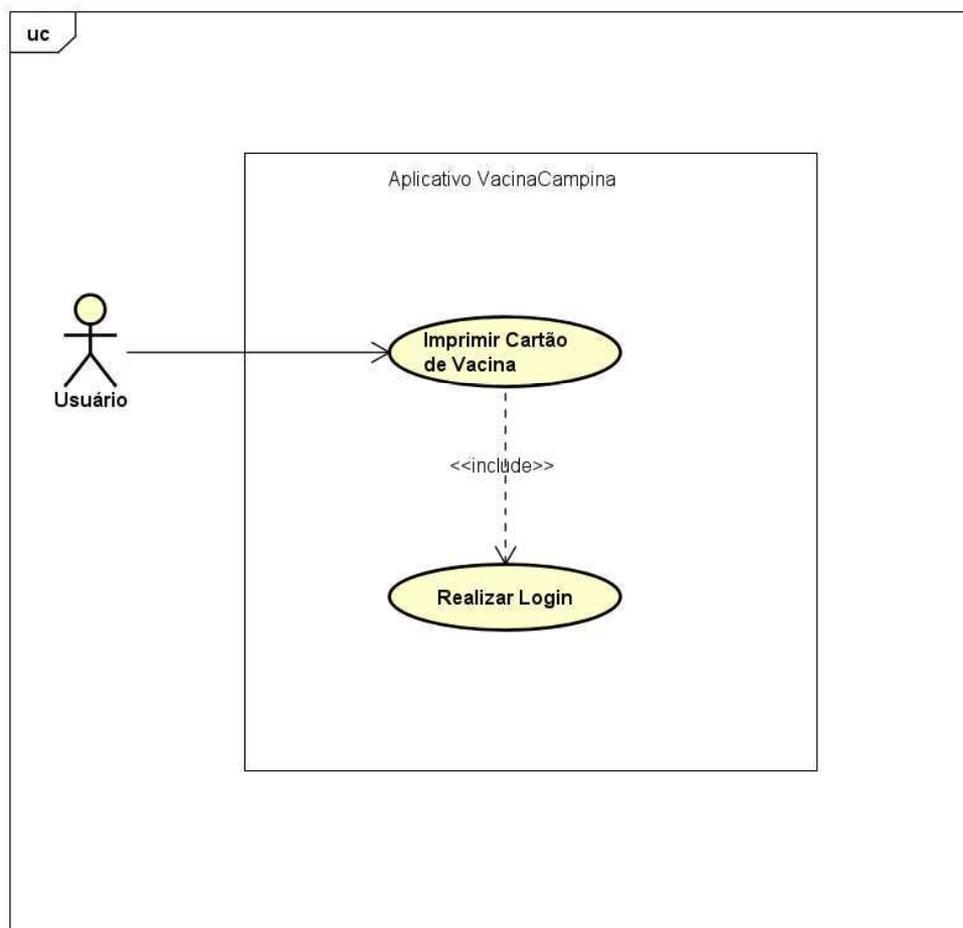
**Fonte:** Elaborada pelo autor, 2020.

O caso de uso Registrar Dose deve permitir que o usuário possa registrar uma dose no seu próprio cartão ou em no cartão de um parente cadastrado, após realizar o fluxo descrito no caso de uso Visualizar Cartão de Vacina (Figura 8) o usuário poderá selecionar uma das vacinas apresentadas para ser direcionado para tela de registro de dose. Ações do ator e respostas dos sistemas estão descritas no Quadro 5.

**Quadro 5 – Registrar Dose**

CDU005 – Registrar Dose	
Interessados e interesses:	Usuário do Aplicativo
Pré-condições:	Está autenticado
Ação do Ator	Resposta do Sistema
Fluxo Principal	
Realiza o registro de uma dose no cartão de vacina	Registra a dose e exibe uma mensagem de sucesso
Fluxos alternativos	
Não existe	

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

**Figura 10 – Imprimir Cartão de Vacina**

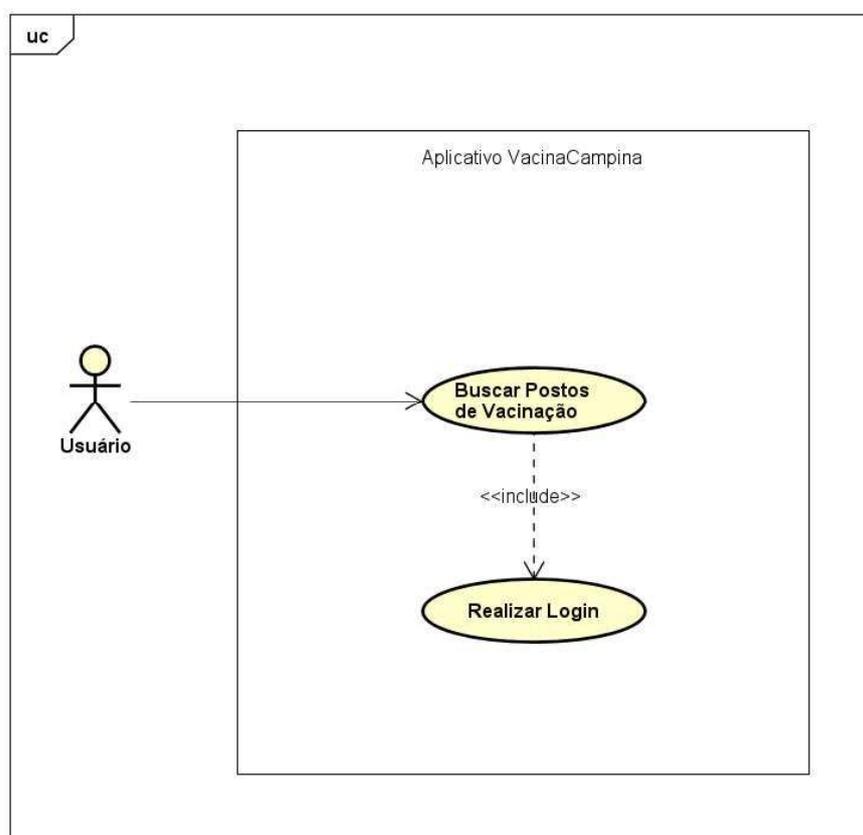
**Fonte:** Elaborada pelo autor, 2020.

O Caso de uso Imprimir Cartão de Vacina (Figura 10) permite que o usuário consiga realizar a impressão ou gerar um PDF (Formato Portátil de Documento), ao realizar o fluxo descrito no caso de uso Visualizar Cartão de Vacina (Figura 8) o usuário estará na tela de visualizar cartão onde será exibido um botão que executa o procedimento de geração do cartão para impressão. No Quadro 6 estão descritas as ações do autor e as respectivas respostas do sistema para essas ações.

**Quadro 6 – Imprimir Cartão de Vacina**

CDU006 – Imprimir Cartão de Vacina	
Interessados e interesses:	Usuário do Aplicativo
Pré-condições:	Está autenticado
Ação do Ator	Resposta do Sistema
Fluxo Principal	
Realiza a impressão do cartão	Exibe um PDF que pode ser impresso para o usuário
Fluxos alternativos	
Não existe	

Fonte: Elaborada pelo autor, 2020.

**Figura 11 – Buscar Postos de Vacinação**

Fonte: Elaborada pelo autor, 2020.

O caso de uso Buscar Postos de Vacinação (Figura 11) permite que usuário logado tenha acesso a funcionalidade de buscar postos de vacinação, ao escolher a opção no menu, o usuário deve ser direcionado para uma tela com o mapa da

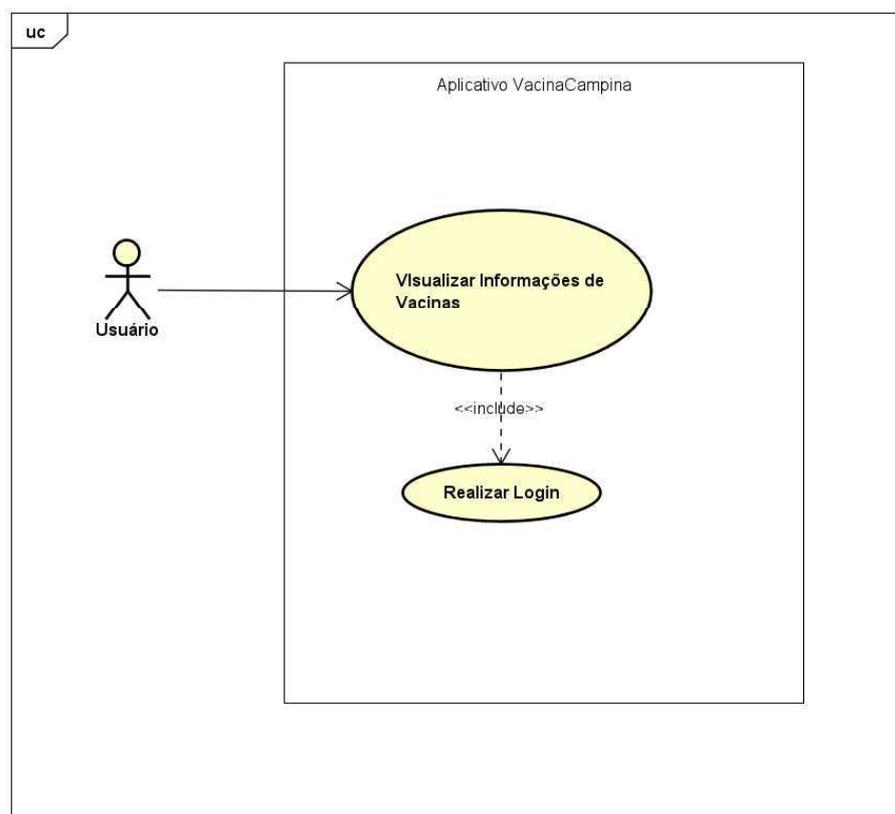
cidade de Campina Grande com todos os pontos de vacinações marcados. O Quadro 7 descreve a ação do autor e a respostas do sistema para o caso de uso descrito.

### Quadro 7 – Buscar Postos de Vacinação

CDU007 – Buscar Postos de Vacinação	
Interessados e interesses:	Usuário do Aplicativo
Pré-condições:	Está autenticado
Ação do Ator	Resposta do Sistema
Fluxo Principal	
Realiza ação de buscar postos de vacinação	Exibe um mapa com marcação em todos os postos de vacinação na cidade de Campina Grande - PB
Fluxos alternativos	
Não existe	

Fonte: Elaborada pelo autor, 2020.

### Figura 12 – Visualizar Informações de Vacinas



Fonte: Elaborada pelo autor, 2020.

O caso de uso Visualizar Informações de Vacinas (Figura 12) permite que usuário logado consiga através do menu principal visualizar a tela de informativa sobre vacinas. A tela deverá listar todas as vacinas disponíveis no Programa Nacional de Imunização (PNI), ao selecionar uma das vacinas deve exibir as principais informações sobre a mesmas. O Quadro 8 descreve a ação do autor e a respectiva resposta do sistema para o caso de uso.

#### Quadro 8 – Visualizar Informações de Vacinas

CDU008 – Visualizar Informações de Vacinas	
Interessados e interesses:	Usuário do Aplicativo
Pré-condições:	Está autenticado
Ação do Ator	Resposta do Sistema
Fluxo Principal	
Realiza ação escolher a opção vacinas no menu	Exibe uma lista com todas as vacinas disponíveis no PNI (Programa Nacional de Imunização)
Fluxos alternativos	
Não existe	

Fonte: Elaborada pelo autor, 2020.

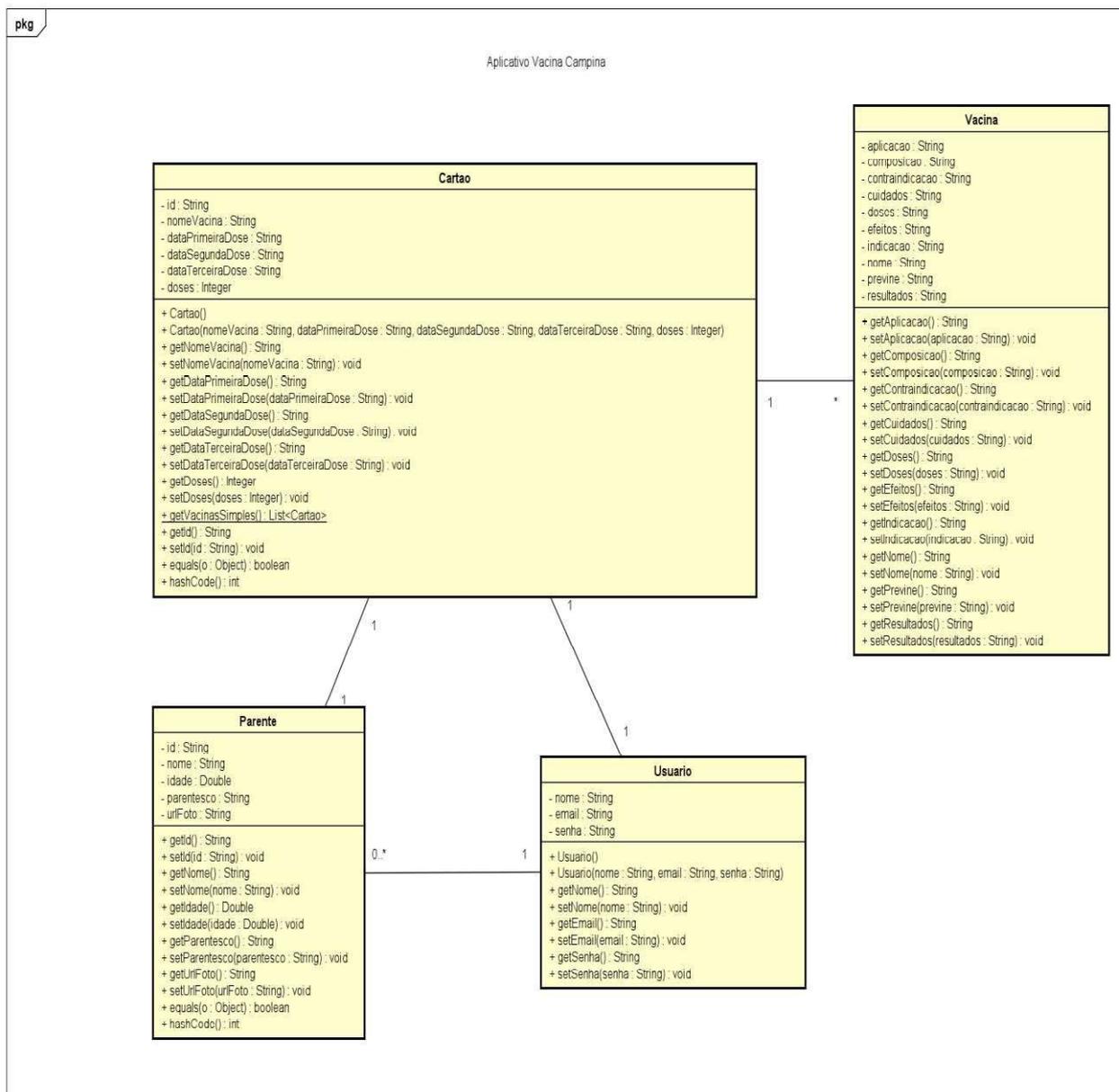
#### 5.2.2 Diagrama de Classes

Os diagramas de classe são usados na elaboração de um modelo de aplicação orientado a objetos para mostrar tanto as classes quanto as associações entre essas classes dentro do sistema (SOMMERVILLE, 2011).

O diagrama de classe é uma modelagem muito útil para o desenvolvimento da aplicação, pois é uma representação estrutural e relacional das classes que servem de modelo para o sistema descrevendo seus atributos e métodos. A representação do diagrama de classes é definida da seguinte maneira (Figura 13):

- O nome da classe é exibido na parte superior.
- Os atributos são exibidos no meio.
- Na parte inferior são listados os métodos da classe.

Figura 13 – Diagrama de Classes



Fonte: Elaborada pelo autor, 2020.

Na Figura 13 está representado o diagrama de classes do aplicativo que possui 4 classes que se relacionam entre si:

- **Classe Usuário:** Representa o usuário na aplicação se relaciona de forma zero para muitos com a classe Parente e um para um com a classe cartão.
- **Classe Parente:** Representa os parentes dos usuários que podem ser cadastrados no aplicativo, se relaciona um para um com a classe cartão.

- **Classe Cartão:** Representa o cartão de vacina na aplicação e se relaciona um para muitos com a classe vacina.
- **Classe Vacina:** Representa uma vacina na aplicação e compõe o cartão de vacina.

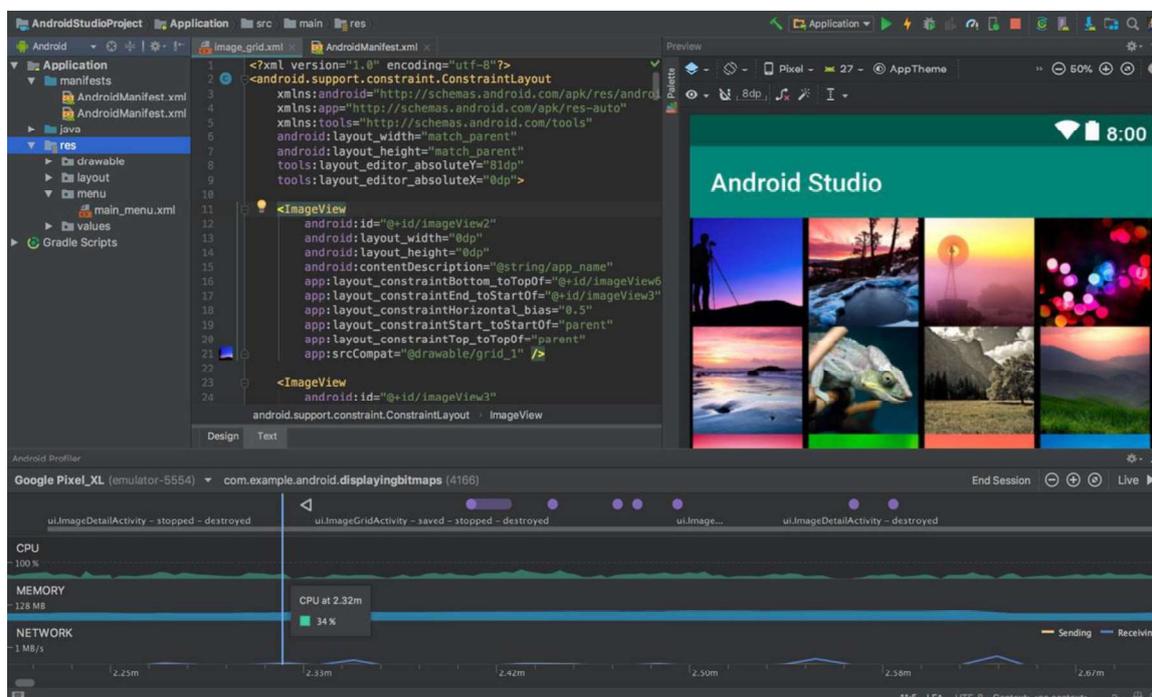
### 5.3 Tecnologias e Materiais Envolvidos

Neste tópico serão apresentadas e descritas as tecnologias e ferramentas que foram necessárias para elaboração e construção da aplicação entre elas o GIT, IDE utilizada, linguagem de programação e tecnologias de nuvem.

#### 5.3.1 Android Studio

O Android Studio é um ambiente de desenvolvimento (IDE), para plataforma Android baseado no software IntelliJ IDEA da JetBrains, lançado em 2013 para substituir o Eclipse Android Development Tools (ADT) como a IDE primária do Google de desenvolvimento nativo para Android, é disponibilizado gratuitamente sob a Licença Apache 2.0 para Windows, Mac OS e Linux (ANDROID STUDIO, 2021).

Figura 14 – Android Studio



Fonte: <https://developer.android.com/studio>

O Android Studio disponibiliza um conjunto de ferramentas que facilitam o trabalho do desenvolvedor, um excelente editor de texto com recurso de autocomplete, permite criar e visualizar os layouts sem necessidade de compilação de código, disponibiliza ainda um emulador que permite testar a aplicação em uma infinidade de dispositivos Android, além de dar suporte ao *Gradle* que constrói e gera o executável do aplicativo. Por fim uma das desvantagens da ferramenta são os seus requisitos para ser executada (Quadro 9).

**Quadro 9 – Requisitos Mínimos Android Studio**

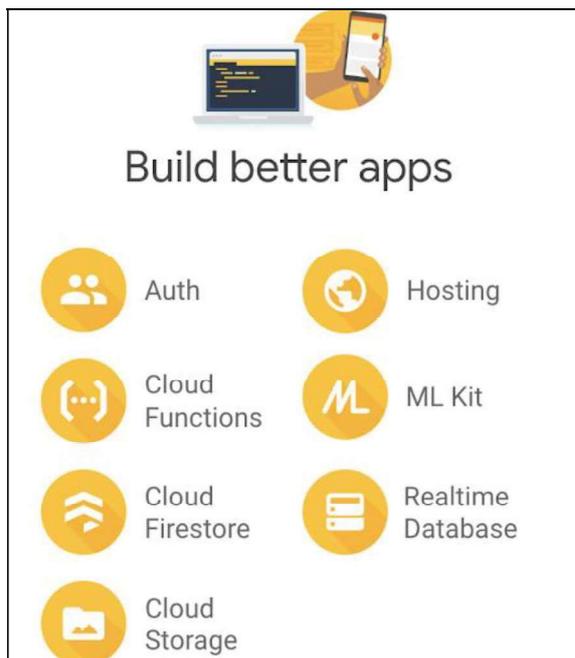
	Windows 8/10	Mac OS 10.14 ou superior	Linux
Processador	Arquitetura de CPU x86_64; Intel Core de 2ª geração ou mais recente, ou CPU AMD com suporte para um Windows Hypervisor	Chips baseados em ARM ou Intel Core de 2ª geração ou mais recente com suporte para Hypervisor.Framework	Arquitetura de CPU x86_64; Intel Core de 2ª geração ou mais recente, ou processador AMD com suporte para AMD Virtualization (AMD-V) e SSSE3.
Memória RAM	8 GB de RAM ou mais	8 GB de RAM ou mais	8 GB de RAM ou mais
Memória em Disco	8 GB de espaço em disco	8 GB de espaço em disco	8 GB de espaço em disco
Resolução da Tela	1280 x 800	1280 x 800	1280 x 800

Fonte: Elaborada pelo autor, 2021.

### 5.3.2 Firebase

O Firebase é uma plataforma de desenvolvimento de aplicativos móveis criado pelo Google que permite criar, melhorar e expandir aplicativos, é considerado um Backend-as-a-Service (Baas) que pode ser integrado com aplicações Android, IOS e Web. (GOOGLE FIREBASE, 2021). Possui uma vasta opção de ferramentas que ajudam na construção de uma aplicação (Figura 15).

**Figura 15** – Ferramentas de construção de aplicativos Firebase.



**Fonte:** <https://firebase.google.com>

Para desenvolvimento do aplicativo VacinaCampina foram usados o *Auth* para autenticação, *Realtime Database* para armazenamento de dados, e o *Cloud Storage* para armazenamento de imagens. Os serviços utilizados:

- **Auth** – Adiciona uma solução de identidade ponta a ponta ao aplicativo para facilitar autenticação, o login e a integração do usuário em apenas algumas linhas de código.
- **Realtime Database** – Armazena e sincroniza dados JSON entre os usuários da aplicação quase em tempo real, com forte segurança baseada no usuário.
- **Cloud Storage** – Armazena e fornece conteúdo gerado pelo usuário como imagens, arquivos entre outros.

### 5.3.3 Controle de Versão e *GIT*

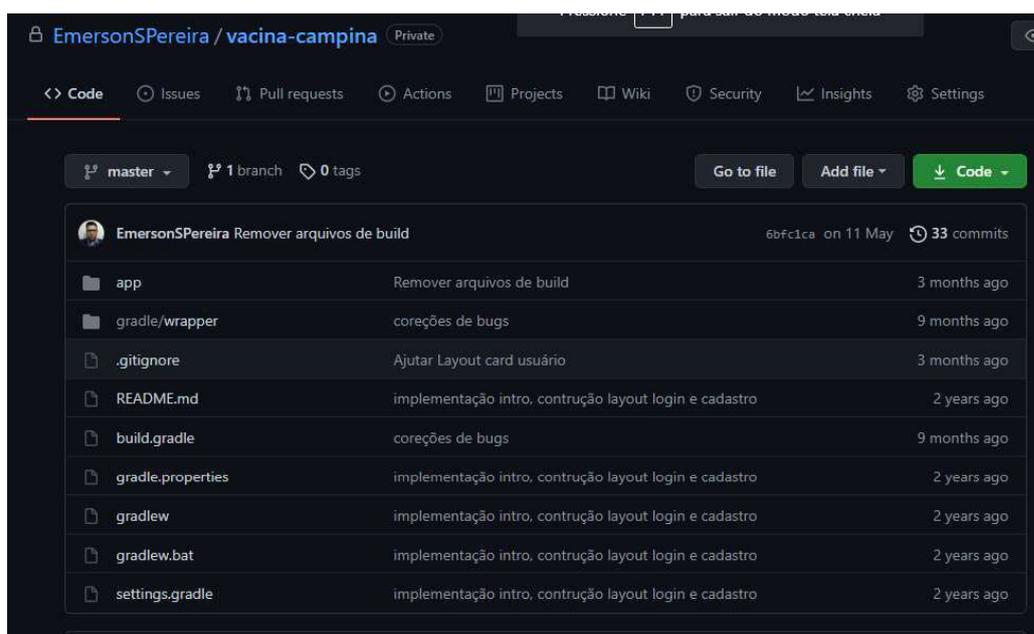
O controle de versão é a atividade de rastrear e gerenciar alterações em um determinado arquivo que na maioria das vezes são códigos fonte de softwares, com objetivo de automatizar esse processo surgiram as ferramentas de controle de versão. Elas permitem acompanhar alterações

desde as versões iniciais, também é possível detectar e mesclar as alterações que são feitas no mesmo arquivo, além de identificar os conflitos entre as alterações de forma automatizada (AQUILES; FERREIRA, 2014).

Uma das ferramentas mais utilizadas para realização de controle de versões é o GIT, um projeto de código aberto maduro e de manutenção contínua criado em 2005 por Linus Torvalds, o famoso criador do Linux. De acordo com Aquiles e Ferreira (2014) o GIT pode ser considerado uma máquina do tempo extremamente rápida e um mecanismo de integração de alta competência.

Por ser uma ferramenta tão completa, o GIT foi a ferramenta utilizada para realizar o controle de versão do código do aplicativo VacinaCampina, além do versionamento local foi utilizado o GitHub para hospedar remotamente o código, tornando possível o acesso de qualquer lugar do mundo. O GitHub traz uma interface muito amigável onde é possível visualizar as principais informações do projeto como estrutura de arquivos, últimas alterações realizadas, histórico de commits (Figuras 16) além de disponibilizar um conjunto de funcionalidades que facilitam o trabalho em equipe.

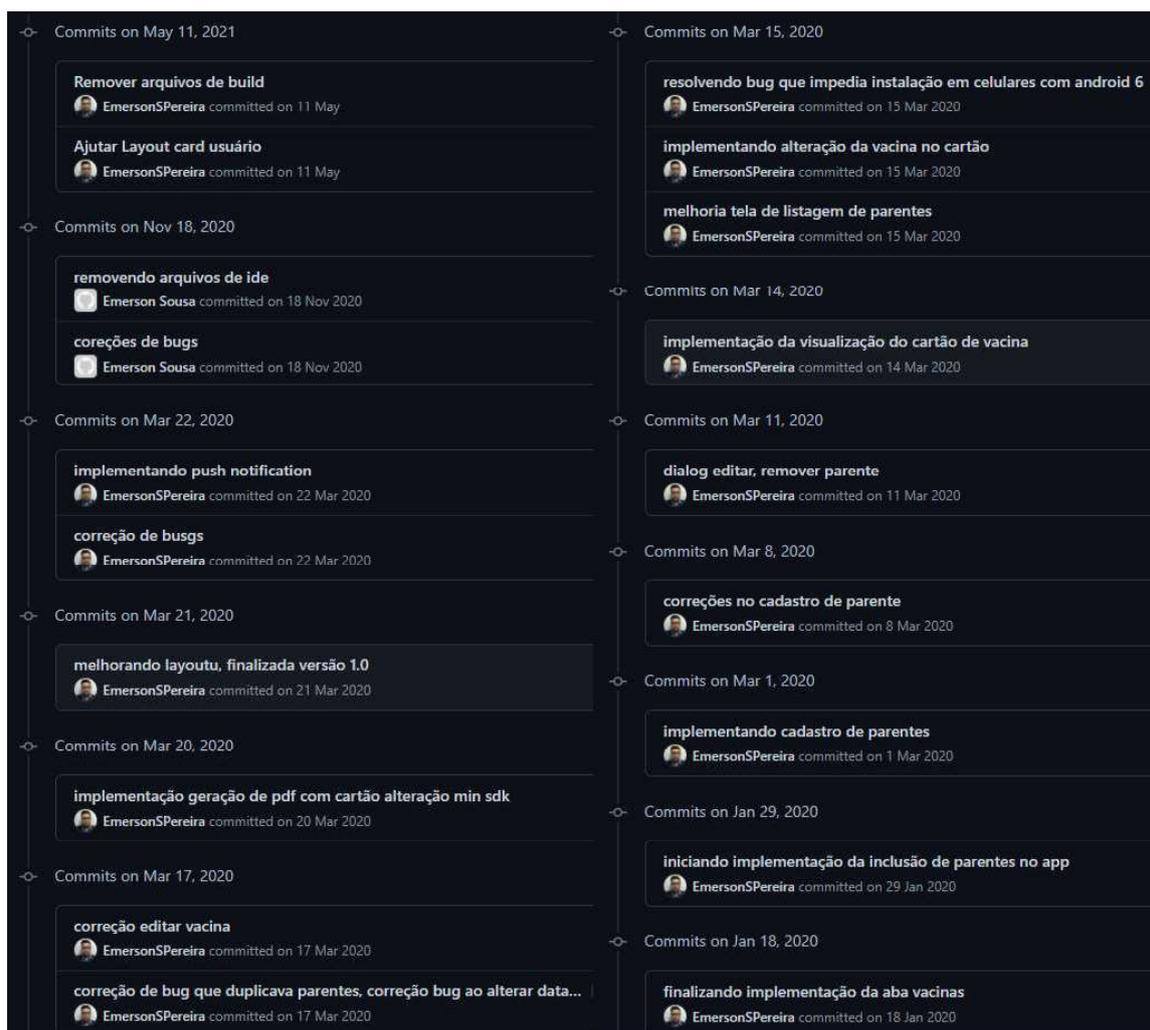
**Figura 16** – Página de detalhes do GitHub.



Fonte: <https://firebase.google.com>

Durante o desenvolvimento da aplicação apenas uma *branch* foi necessária (master) para versionar o código, foram realizados 33 *commits* até a finalização do projeto (Figura 17). Os principais comando do GIT utilizados durante o processo de versionamento de código estão disponíveis no Quadro 10.

**Figura 17** – Histórico de *commits* GitHub.



Fonte: <https://github.com/EmersonSPereira/vacina-campina/commits/master>

**Quadro 10** – Comandos Git utilizados

<b>Comando</b>	<b>Ação</b>
<b>git init</b>	Cria um novo repositório git local
<b>git add</b>	Adiciona novos arquivos e alterações ao índice
<b>git commit</b>	Confirma as alterações realizadas
<b>git clone</b>	Clona um repositório
<b>git status</b>	Exibe a lista de arquivo alterados juntamente com os arquivos que ainda não foram confirmados
<b>git pull</b>	Traz as alterações presentes no repositório remoto para o local
<b>git push</b>	Envia as alterações do repositório local para o remoto
<b>git log</b>	Exibe a lista de commits

Fonte: Elaborada pelo autor, 2021

#### **5.3.4 Hardwares Utilizados**

Apesar da IDE utilizada disponibilizar uma série de emuladores para o desenvolvedor, foram utilizados uma série de celulares físicos para que os testes de funcionalidades pudessem ser feitos em diferentes hardwares e versões do sistema operacional Android. Abaixo estão listados todos esses hardwares:

- **Redmi Note 8** – *smarthphone* com sistema operacional Android 10 Q, processador Octa-core 2.01 GHz, 4Gb de memória RAM.
- **Redmi Note 8 Pro** – *smarthphone* com sistema operacional Android 10 Q, processador Octa-core 2.01 GHz, 6Gb de memória RAM.
- **Moto G 5 Plus** - *smarthphone* com sistema operacional Android 8.1, processador Octa-core 1.2 GHz, 2Gb de memória RAM.
- **LG K10** - *smarthphone* com sistema operacional Android 6.0, processador Octa-core 1.3 GHz, 2Gb de memória RAM.
- **Redmi Note 9 Pro** - *smarthphone* com sistema operacional Android 10 Q, processador Octa-core 1.9 GHz, 6Gb de memória RAM.

## 6 RESULTADOS E DISCUSSÕES

Neste capítulo serão apresentados os resultados atingidos com o desenvolvimento do aplicativo proposto, que será apresentado em duas etapas a primeira contendo o *Back-End* e Banco de dados e na etapa posterior será mostrado as funcionalidades do aplicativo e o design do Front-End do projeto.

### 6.1 *Back-End* e Banco de dados

No decorrer desta seção serão abordados os temas referentes a construção do *back-end* do projeto, como segurança e autenticação, modelagem e construção do banco de dados, serão apresentados alguns trechos de código que ilustrarão algumas etapas.

#### 6.1.1 Segurança e Autenticação

Para autenticação foi utilizado o *Firebase Auth* que prover autenticação e autorização de forma segura e simplificada. Através do seu SDK foi possível:

- Gerenciar usuários da aplicação.
- Prover Login através de e-mail e senha.
- Realizar verificação de e-mail através de envio de link.
- Recuperar senha através do e-mail.

O processo de Login ocorre de forma simples (Figura 18), uma solicitação contendo o e-mail e a senha do usuário é enviada para o Auth, caso retorne com sucesso segue o fluxo do login, caso o contrário, o mesmo pode retornar *FirebaseAuthInvalidUserException* caso o e-mail não esteja cadastrado ou *FirebaseAuthInvalidCredentialsException* quando a senha do usuário está incorreta, ambas exceções são tratadas exibidas para o usuário em forma de mensagem de validação.

**Figura 18** – Código Java de login aplicativo Vacina Campina

```

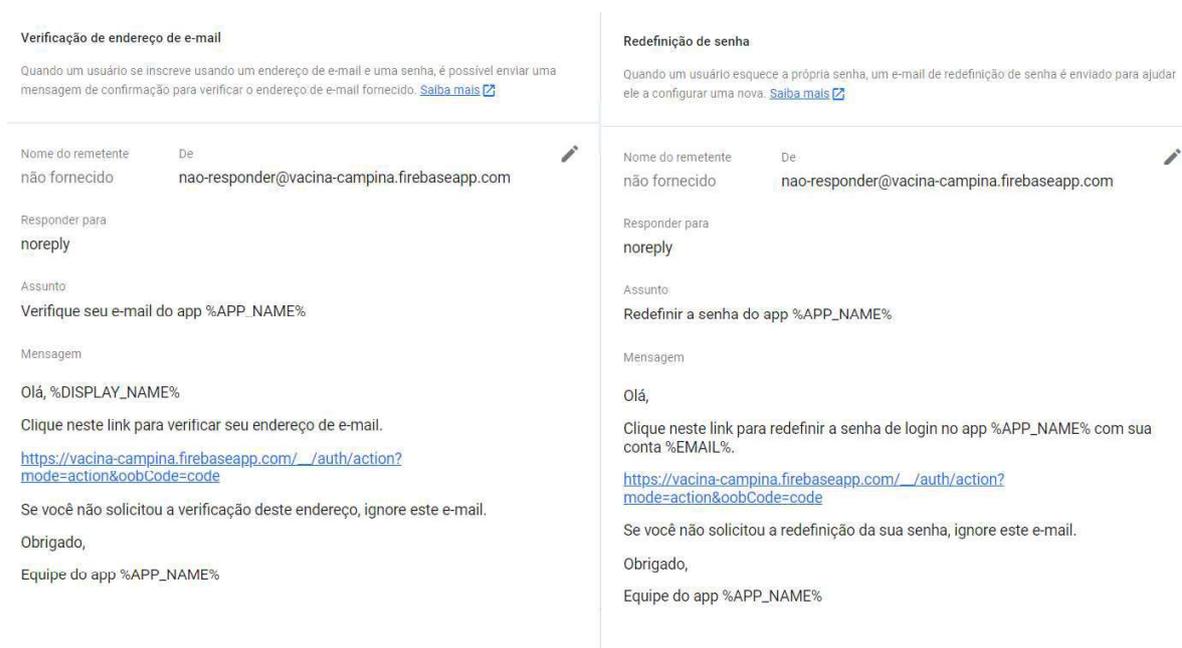
82  /**
83   * Realiza o login na aplicação
84   */
85  private void logar() {
86      progressBar.setVisibility(View.VISIBLE);
87      FirebaseAuth.getInstance().signInWithEmailAndPassword(campoEmail.getText().toString(), campoSenha.getText().toString())
88          .addOnCompleteListener(LoginActivity.this, new OnCompleteListener<AuthResult>() {
89              @Override
90              public void onComplete(@NonNull Task<AuthResult> task) {
91                  if(task.isSuccessful()){
92                      //salvando primeiro login localmente
93                      editorPreferencias.putBoolean(UsuarioService.getUsuarioLogado().getUid(),true);
94                      editorPreferencias.commit();
95                      startActivity(new Intent(LoginActivity.this, TelaPrincipalActivity.class));
96                  }else {
97                      progressBar.setVisibility(View.INVISIBLE);
98                      try {
99                          throw task.getException();
100                     }catch (FirebaseAuthInvalidUserException authInvalidUserException){
101                         Toast.makeText(LoginActivity.this,
102                             EMAIL_NÃO_CADASTRADO, Toast.LENGTH_LONG).show();
103                     }catch (FirebaseAuthInvalidCredentialsException authInvalidCredentialsException){
104                         Toast.makeText(LoginActivity.this,
105                             SENHA_INCORRETA, Toast.LENGTH_LONG).show();
106                     } catch (Exception e) {
107                         Toast.makeText(LoginActivity.this,
108                             ERRO_AO_REALIZAR_LOGIN_VERIFIQUE_OS_DADOS_E_TENTE_NOVAMENTE,
109                             Toast.LENGTH_LONG).show();
110                         e.printStackTrace();
111                     }
112                 }
113             }
114         });
115     }
116 }
117

```

Fonte: Elaborada pelo autor, 2021.

Tanto para recuperação de senha quanto para confirmação de cadastro foram definidos modelos de e-mail que são enviados de acordo com cada situação, ao realizar o cadastro na aplicação o usuário recebe um e-mail com um link de confirmação e ao optar por utilizar a funcionalidade de recuperação de senha é enviado um e-mail com um link de recuperação (Figura 19).

**Figura 19 – Template de verificação de e-mail e recuperação de senha**



Fonte: Elaborada pelo autor, 2021.

### 6.1.2 Banco de Dados e Armazenamento

O banco de dados foi construído utilizando o *Real Time Database*, onde os dados são estruturados em forma documentos JSON para armazenar pares de chave-valor, o mesmo foi pensado e construído para atender perfeitamente a necessidade da aplicação, tendo em vista que os dados do usuário são gerenciados pelo *Firestore*. Apenas três entidades foram suficientes na modelagem:

- **Cartão** - Onde são registradas as doses das vacinas.
- **Parente** – Onde são salvos os registros dos parentes do usuário da aplicação.
- **Vacinas** – Onde são registradas todas as informações referentes as vacinas.

Na Figura 20 temos a representação estrutural do banco de dados, por exemplo, a classe Parente que está definida no diagrama de classes (Figura 13), seus registros tem “parente” como nó principal o nó imediatamente interno a ele é o identificador do usuário principal responsável por gerenciar as cadernetas e interno a esse nó estão os nós de todos os parentes cadastrados, com essa estrutura é possível consultar todos os parentes do usuário logado navegando na estrutura da

seguinte maneira, Nó Principal (Parente) → Nó Interno (Identificador do usuário) → Nó Interno (Contendo todos os parentes do usuário).

**Figura 20** – Estrutura de banco de dados aplicativos Vacina Campina



Fonte: Elaborada pelo autor, 2021.

A SDK (Kit de Desenvolvimento de Software) do Firebase prover uma API completa de consultas, que permitiu construir consultas de formas simples. Na Figura 21, podemos ver o trecho de código que realiza a consulta de vacinas por nome, podendo ser escrito apenas parte do mesmo.

**Figura 21** – Consulta de vacinas por nome

```

36     if (!textoDigitado.isEmpty()) {
37         Query query = VacinaService.getDatabaseReference().orderByChild(NOME)
38             .startAt(textoDigitado.toUpperCase()).endAt(textoDigitado.toLowerCase() + "\uf8ff");
39     }

```

Fonte: Elaborada pelo autor, 2021.

O armazenamento de imagens do projeto foi construído fazendo com o *Firebase Storage* uma ferramenta que possibilitou o armazenamento em nuvem com alta disponibilidade e segurança, além de trazer consigo um SDK que facilitou a implementação de upload de arquivos no projeto.

**Figura 22** – Algoritmo de envio de imagem para o Firebase Storage

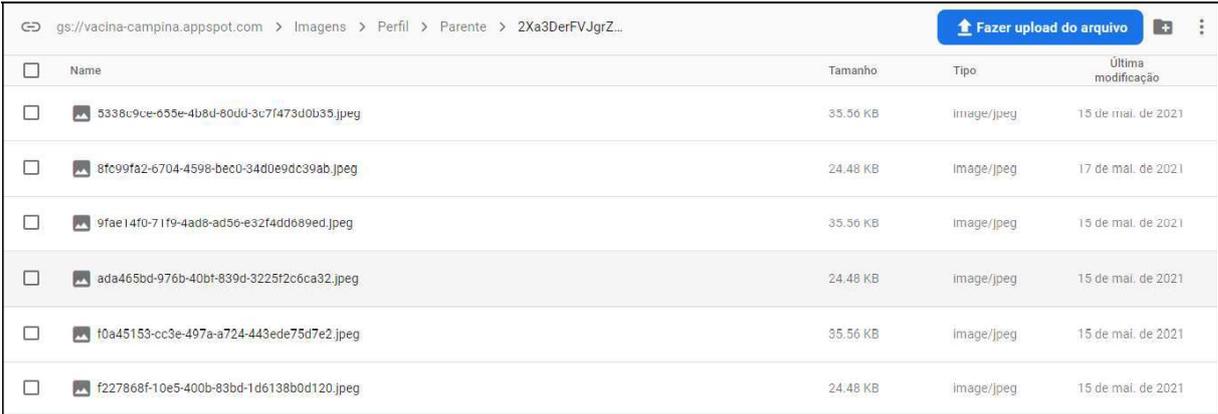
```

165     final StorageReference imagemRef = FirebaseConfig.getStorageReference().child(IMAGENS)
166         .child(PERFIL)
167         .child(PARENTE)
168         .child(FirebaseConfig.getAuth().getCurrentUser().getUid())
169         .child(UUID.randomUUID().toString() + JPEG);
170
171     final UploadTask uploadTask = imagemRef.putBytes(dadosImagem);
172     uploadTask.addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {
173         @Override
174         public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
175             uploadTask.continueWithTask(new Continuation<UploadTask.TaskSnapshot, Task<Uri>>() {
176                 @Override
177                 public Task<Uri> then(@NonNull Task<UploadTask.TaskSnapshot> task) throws Exception {
178                     if (!task.isSuccessful()) {
179                         throw task.getException();
180                     }
181                 }
182                 return imagemRef.getDownloadUrl();
183             }
184         })
185         .addOnCompleteListener(new OnCompleteListener<Uri>() {
186             @Override
187             public void onComplete(@NonNull Task<Uri> task) {
188                 if (task.isSuccessful()) {
189                     parente.setNome(editTextNome.getText().toString());
190                     parente.setUrlFoto(task.getResult().toString());
191                     parente.setParentesco(spinnerGrauParentesco.getSelectedItem().toString());
192                     ParenteService.salvarAtualizarParente(parente);
193                     progressBarCadastroParente.setVisibility(View.GONE);
194                     Toast.makeText(CadastroParenteActivity.this, PARENTE_SALVO_COM_SUCESSO, Toast.LENGTH_LONG).show();
195                     CadastroParenteActivity.super.onBackPressed();
196                 }
197             }
198         });
199     });
200 }

```

Fonte: Elaborada pelo autor, 2021.

Para o envio da imagem (Figura 22) é criado um objeto *StorageReference* onde será definido o caminho onde será feito o upload do arquivo, em seguida é criado o objeto *UploadTask*, esse responsável por enviar o caminho e o arquivo a ser salvo, como retorno termos a URL de download de imagem que é salva no banco de dados e também pode ser vista e gerenciada através do console do Firebase (Figura 23).

**Figura 23 – Console Firebase Storage**


The screenshot shows the Firebase Storage console interface. At the top, there is a breadcrumb navigation path: 'gs://vacina-campina.appspot.com > Imagens > Perfil > Parente > 2Xa3DerFVJgrZ...'. A blue button labeled 'Fazer upload do arquivo' is visible in the top right corner. Below the navigation, there is a table listing the stored files. Each row includes a checkbox, a file icon, the file name, the size, the type, and the last modification date.

<input type="checkbox"/>	Name	Tamanho	Tipo	Última modificação
<input type="checkbox"/>	 5338c9ce-655e-4b8d-80dd-3c7f473d0b35.jpeg	35.56 KB	image/jpeg	15 de mai. de 2021
<input type="checkbox"/>	 8fc99fa2-6704-4598-bec0-34d0e9dc39ab.jpeg	24.48 KB	image/jpeg	17 de mai. de 2021
<input type="checkbox"/>	 9fae14f0-71f9-4ad8-ad5e-e32f4dd689ed.jpeg	35.56 KB	image/jpeg	15 de mai. de 2021
<input type="checkbox"/>	 ada465bd-976b-40bt-839d-3225f2c6ca32.jpeg	24.48 KB	image/jpeg	15 de mai. de 2021
<input type="checkbox"/>	 f0a45153-cc3e-497a-a724-443ede75d7e2.jpeg	35.56 KB	image/jpeg	15 de mai. de 2021
<input type="checkbox"/>	 f227868f-10e5-400b-83bd-1d6138b0d120.jpeg	24.48 KB	image/jpeg	15 de mai. de 2021

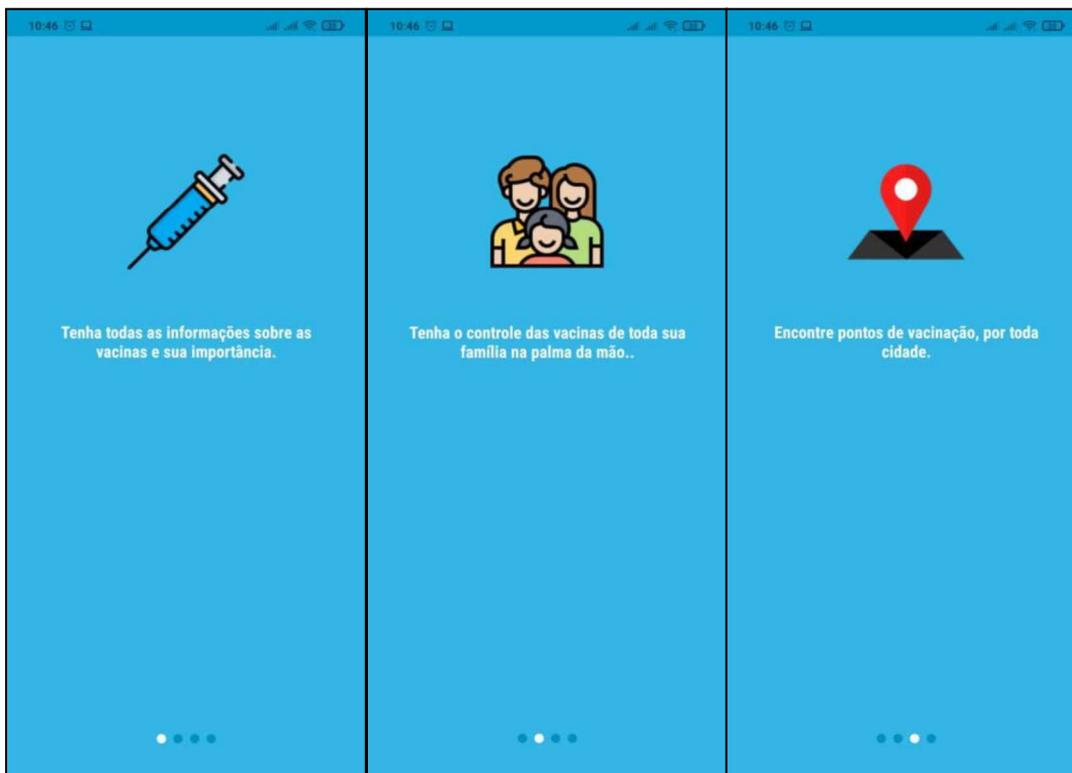
**Fonte:** Elaborada pelo autor, 2021.

## 6.2 Funcionalidades do Aplicativo

O aplicativo de controle de cartão de vacinação foi desenvolvido de acordo com os requisitos funcionais e não funcionais descritos respectivamente no APÊNDICE A, e no APÊNDICE B. Um único usuário que será responsável por gerenciar o cartão de vacina de toda família poderá se cadastrar no app e em seguida incluir todos os outros membros possibilitando ao mesmo cadastrar todas as vacinas, gerar um cartão de vacina para impressão que pode ser visto no APÊNDICE C, como também ter todas as informações sobre as vacinas como por exemplo quantidade de doses, reações adversas e prazo entre uma dose e outra quando houver. Logo abaixo serão descritas detalhadamente as funcionalidades.

### Tela de Apresentação:

Uma tela informativa em formato de slides que é exibida apenas no primeiro acesso após a instalação onde será possível para o usuário entender de forma inicial as funcionalidades que o aplicativo disponibiliza para o mesmo, no final da breve apresentação será exibido para o usuário uma tela (Figura 24), onde o mesmo poderá ir para tela de cadastro, caso não tenha conta ou ir para tela de login caso já possua uma conta.

**Figura 24 – Tela de apresentação**

Fonte: Elaborada pelo autor, 2021.

**Figura 25 – Tela de redirecionamento**

Fonte: Elaborada pelo autor, 2021.

### Tela de Login:

O usuário terá acesso a tela de login no primeiro acesso por meio da tela final de apresentação do aplicativo clicando em “Já tenho conta”, ou será direcionado diretamente para a mesma a partir do segundo acesso e por fim também terá acesso após clicar no botão sair dentro da aba de usuário (Figura 26).

Nesta tela será possível realizar o login, caso o usuário não tenha conta ainda o mesmo poderá clicar na opção “Não tem conta? Cadastre-se” para ser direcionado para fluxo de cadastro de novos usuários (Figura 27), para usuários cadastrados que tenham esquecido sua senha será possível escolher a opção “Esqueceu a senha?” para ser direcionado para tela de recuperação (Figura 29).

**Figura 26 – Tela de Login**



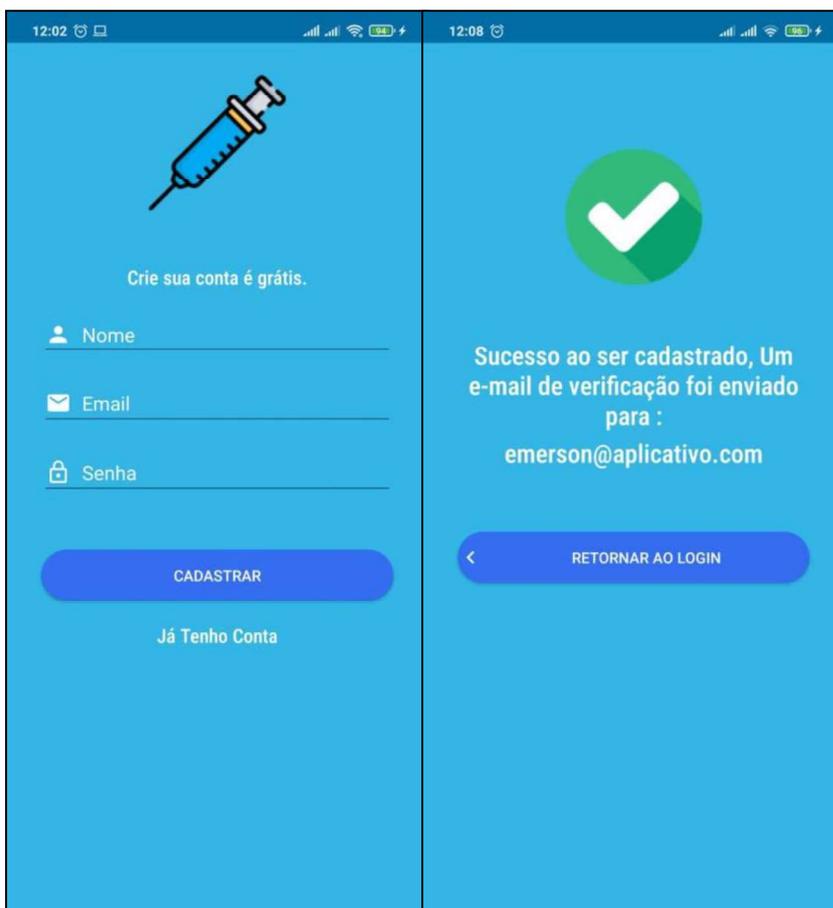
**Fonte:** Elaborada pelo autor, 2021.

### Tela de Cadastro de Usuário:

Esta tela exige do usuário um cadastro bem simples onde somente são necessários: o nome, um e-mail e uma senha, para que o usuário tenha acesso

a todas as funcionalidades do aplicativo, ao realizar a etapa de cadastro com sucesso o usuário receberá um e-mail com um link para que seja possível verificar se o mesmo e-mail digitado no cadastro é válido (Figuras 27 e 28).

**Figura 27 – Fluxo Tela de Cadastro**



Fonte: Elaborada pelo autor, 2021.

**Figura 28 – E-mail de Verificação**

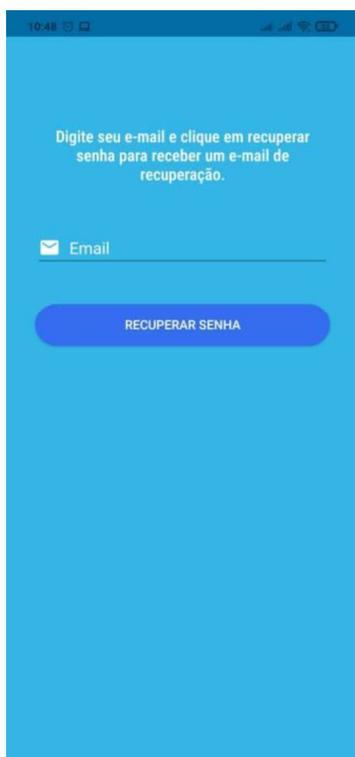


Fonte: Elaborada pelo autor, 2021.

### Tela de Recuperação de senha:

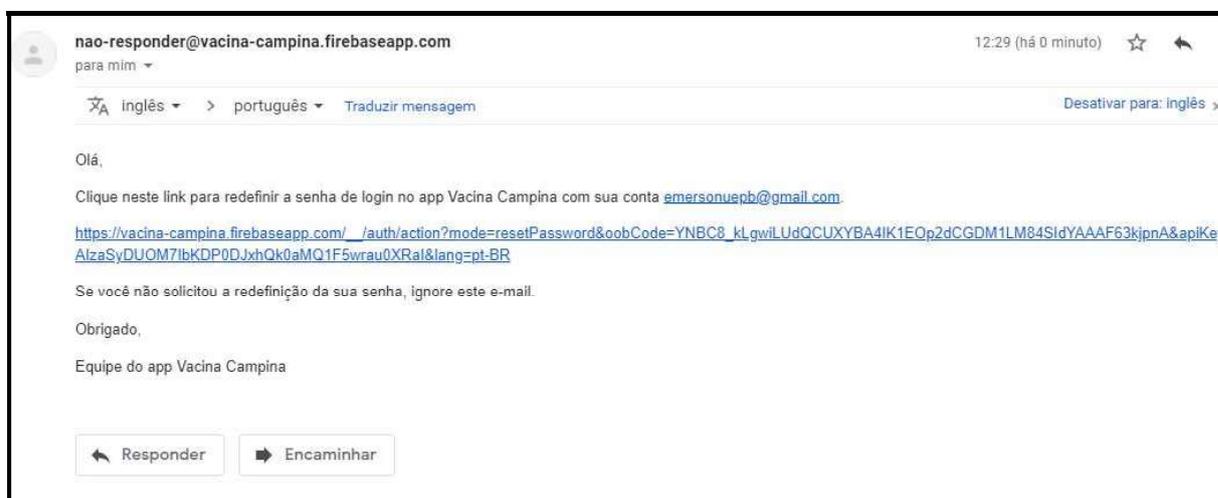
Nesta tela que pode ser acessada através da tela de login (figura 22) clicando na opção “Esqueceu sua senha?”, o usuário que esqueceu sua senha poderá recuperar a mesma a partir deste fluxo, ao digitar o e-mail e clicar no botão recuperar senha, o mesmo recebe um e-mail com todas as instruções necessárias para como pode ser visto nas Figuras 29 e 30.

**Figura 29** – Tela de Recuperação de Senha



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

**Figura 30** – E-mail de recuperação de senha



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

Ao clicar no link o usuário será direcionado para uma página onde poderá digitar uma nova senha e após salvar poderá usar a nova senha para entrar no aplicativo.

**Figura 31** – Fluxo de Redefinição de Senha

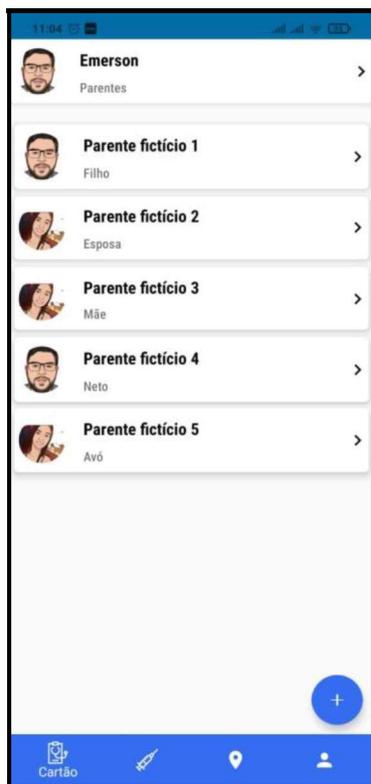


The image displays two sequential screens from a password reset process. The first screen, on the left, is titled "Redefinir sua senha" and is for the email address "emersonuepb@gmail.com". It features a text input field labeled "Nova senha" with a toggle icon to its right, and a blue button labeled "SALVAR" at the bottom. The second screen, on the right, is titled "Senha alterada" and contains the instruction "Faça login com a sua nova senha".

Fonte: Elaborada pelo autor, 2021.

#### **Tela Listagem de Parentes:**

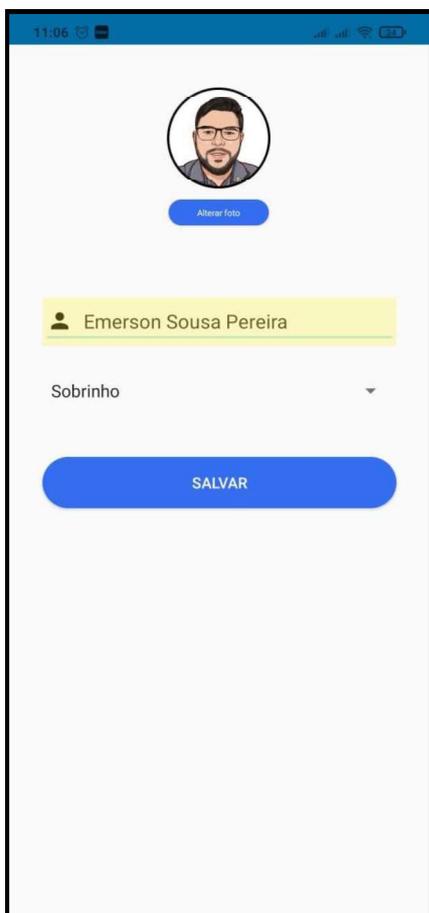
Permite que usuário visualize todos os parentes cadastrados por ele, ao pressionar e segurar por segundos um dos parentes na lista será exibido opções de editar ou remover o mesmo, ao dar um toque simples em cima do parente o usuário será direcionado para o cartão de vacina do parente e pôr fim ao clicar no botão "+" permite ao usuário realizar o cadastro de um novo parente.

**Figura 32 – Listagem de Parentes**

Fonte: Elaborada pelo autor, 2021.

**Tela Cadastro de Parentes:**

Nesta tela é possível realizar o cadastro de um parente para posteriormente poder gerenciar o cartão de vacina do mesmo, para inclusão de um novo parente é necessário apenas fornecer uma foto o nome e o grau de parentesco que o usuário tem com o mesmo, após ser salvo o parente será exibido na tela de listagem de parentes (Figura 32).

**Figura 33 – Cadastro de Parente**

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

### **Tela Cartão de Vacina:**

Pode ser considerada a principal funcionalidade do aplicativo aqui são exibidas todas as vacinas disponibilizadas pelo Programa Nacional de Imunização, onde cada vacina é exibida em um *card*, onde é possível verificar a quantidade de doses que são necessárias para cada imunizante ao tocar em *cards*, o usuário será direcionado para a tela de cadastro de doses, onde o mesmo poderá realizar o registro da dose tomada (Figura 35).

Outra funcionalidade importante que esta tela proporciona é a geração de um cartão de vacina para ser impresso (APÊNDICE C) onde serão exibidas as informações do parente ou do usuário contendo todas as doses já tomadas e as datas das mesmas.

**Figura 34 – Cartão de Vacina**

The figure displays three screenshots of a mobile application interface for a vaccination card, titled "Parente fictício 1".

- Left Screenshot:** Shows the BCG vaccine with 1st, 2nd, and 3rd doses. The 1st dose is marked as "Dose não tomada" (red X), while the 2nd and 3rd are "Não necessário" (black minus). Below are Hepatite B (1st, 2nd, 3rd doses, all "Dose não tomada") and Penta/DTP (1st, 2nd doses, both "Dose não tomada").
- Middle Screenshot:** Shows the 2nd and 3rd doses of BCG (both "Dose não tomada"). Below are Meningocócica Conjugada (1st, 2nd, 3rd doses, all "Dose não tomada") and Febre Amarela (1st, 2nd, 3rd doses, all "Dose não tomada").
- Right Screenshot:** Shows Febre Amarela (1st, 2nd, 3rd doses, all "Dose não tomada"). Below are Hepatite A (1st, 2nd, 3rd doses, all "Dose não tomada") and Triplíce Viral (1st, 2nd doses, both "Dose não tomada").

Each vaccine entry includes a status indicator (red X for "Dose não tomada", black minus for "Não necessário") and a blue "SALVAR" button at the bottom right of the card.

Fonte: Elaborada pelo autor, 2021.

**Figura 35 – Cadastro de Dose**

The figure shows a screenshot of the "Cadastro de Dose" screen for the BCG vaccine. It features three input fields for the 1st, 2nd, and 3rd doses, each with a date format "dd/mm/aaaa" placeholder. A blue "SALVAR" button is located at the bottom of the screen.

Fonte: Elaborada pelo autor, 2021.

### Tela Informativa Sobre Vacinas:

Esta funcionalidade do aplicativo tem o objetivo de informar e conscientizar o usuário da importância das vacinas, exibe uma listagem com todas as vacinas disponíveis no PNI além de possibilitar a realização de uma busca por nome.

Ao selecionar uma das vacinas na lista será direcionado para tela de detalhe (Figura 36), onde irá visualizar informações importantes como:

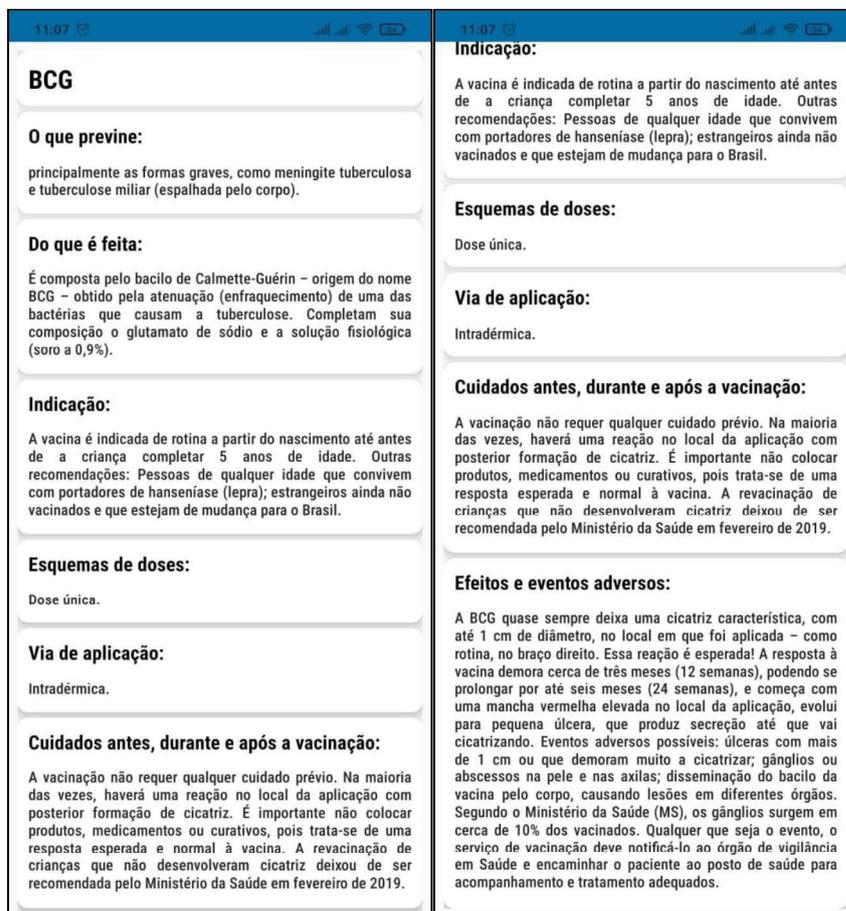
- O que previne
- Do que é feita
- Indicação
- Esquema de doses
- Via de aplicação
- Cuidados antes, durante e após a vacinação
- Efeitos e eventos adversos

**Figura 36 – Listagem e Busca de Vacinas**



Fonte: Elaborada pelo autor, 2021.

Figura 37 – Tela Informativa Sobre Vacina

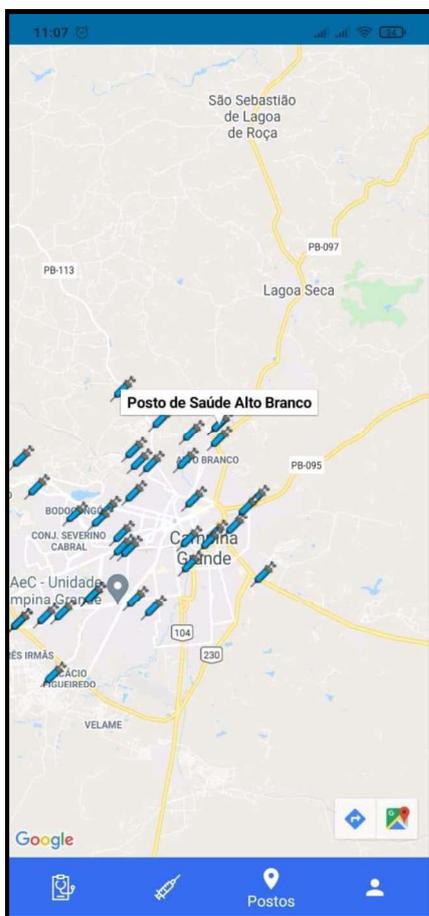


Fonte: Elaborada pelo autor, 2021.

### Tela Postos de Vacinação:

Disponibiliza para o usuário uma interface com um mapa da cidade de Campina Grande, onde foram mapeadas todas as unidades de básicas de saúde, onde é possível encontrar doses dos imunizantes disponibilizados pelo Programa Nacional de imunização. Ao selecionar um dos pontos de vacinação no mapa o usuário poderá tocar no botão de rota que aparece no canto inferior da tela e o aplicativo irá direcionar para o Google Maps com a rota traçada para o destino selecionado.

**Figura 38 – Tela de Postos de Vacinação**

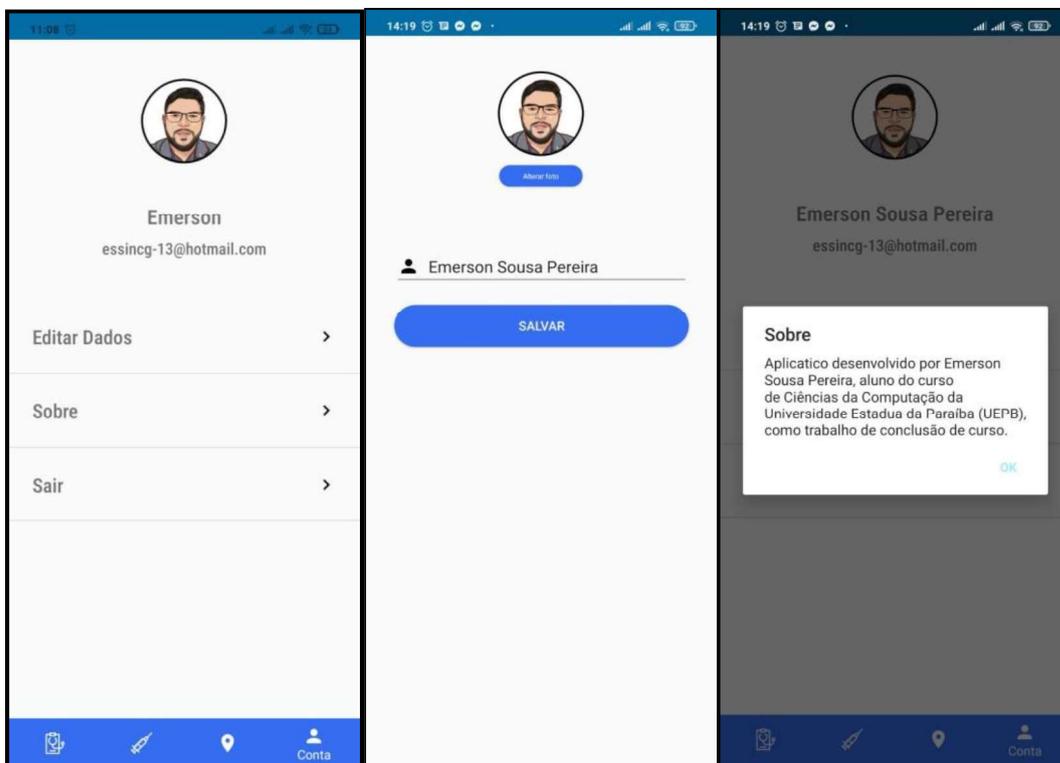


Fonte: Elaborada pelo autor, 2021.

### **Tela de Configuração de Usuário:**

Nesta tela são exibidas informações de foto, nome e e-mail pertencentes ao usuário logado, é mostrado também um menu onde é possível ter acesso as funcionalidades descritas abaixo:

- **Editar Dados** – Permite que usuário edite informações relacionadas ao seu perfil como nome e foto.
- **Sobre** – Exibe as informações referente ao desenvolvedor do aplicativo
- **Sair** – Desconecta o usuário logado atualmente e redireciona o mesmo para a tela de login

**Figura 38** – Tela de Configuração de Usuário

Fonte: Elaborada pelo autor, 2021.

## 7 CONCLUSÃO

A vacinação que cumpre um papel fundamental no controle e erradicação de doenças virais é ainda controlada por um documento chamado caderneta de vacinação, um meio físico e que para emissão de segunda via é necessário refazer a administração de novas vacinas o que gera um prejuízo econômico para o setor público de saúde, além do transtorno gerado ao paciente ao ter que tomar todas as vacinas novamente.

O aplicativo (Vacina Campina App) foi proposto como uma ferramenta de apoio ao gerenciamento da caderneta de vacinação em um ambiente familiar permitindo que uma única pessoa consiga ter acesso a informação de todos os membros da família centralizados em um único lugar, além de proporcionar informações e informar pontos de vacinação.

A necessidade de construção do aplicativo foi pensada mediante relatos de familiares e pessoas próximas sobre o constrangimento e dificuldade para se obter uma segunda via do cartão de vacinação, já que os mesmos não lembravam quais vacinas tinham tomado. Portanto com o objetivo de sanar os problemas relatados. O aplicativo possibilita que o próprio usuário possa realizar o controle pessoal da sua caderneta e permite também que o mesmo consiga gerenciar a caderneta de vacinação de todos os seus dependentes, além disponibilizar uma seção informativa que mostra detalhes importantes sobre cada vacina disponibilizada através do PNI (Programa Nacional de Imunização). Embora não seja possível substituir a carteirinha de vacinação disponibilizada pelo o governo, a facilidade de controlar essas informações através de um aplicativo proporciona controle e segurança para os seus usuários.

Para trabalhos futuros, espera-se que o aplicativo possa ser melhorado na funcionalidade onde é possível localizar postos de vacinação, expandindo a mesma para outros municípios além de Campina Grande – PB, sugere-se também que seja implementado uma funcionalidade que possa avisar os usuários de doses futuras que o mesmo deve tomar através de notificações ou envio de e-mail.

## REFERÊNCIAS

ABLESON, W.Frank *et al.* **Android em Ação**. 3. ed. Rio de Janeiro: Elsever Editora, 2012. 609 p. Tradução Eduardo Kraszczuk e Edson Furmankiewicz.

ANDROID STUDIO. **Site Oficial**. 2021. Disponível em: [https://developer.android.com/studio?hl=pt&gclid=CjwKCAjw64eJBhAGEiwABr9o2N30MtKm8\\_gI0PPF5F09\\_NkJ20fvxnkt5X6FZSPpRYH1QZ225tn3fRoCtewQAvD\\_BwE&gclidsrc=aw.ds](https://developer.android.com/studio?hl=pt&gclid=CjwKCAjw64eJBhAGEiwABr9o2N30MtKm8_gI0PPF5F09_NkJ20fvxnkt5X6FZSPpRYH1QZ225tn3fRoCtewQAvD_BwE&gclidsrc=aw.ds). Acesso em: 22 ago. 2021.

ANDROID. **Site Oficial**. 2021. Disponível em: <https://developer.android.com/guide/platform?hl=pt-br>. Acesso em: 22 ago. 2021.

AQUILES, Alexandre; FERREIRA, Rodrigo. **Controlando versões com Git e Github**. São Paulo: Casa do Código, 2014. 220 p.

Bonanni P. **Demographic impact of vaccination: a review**. *Vaccine*. 1998; 17: S120-5.

CETIC (São Paulo). **Pesquisa Sobre o Uso das Tecnologias de Informação e Comunicação nos Estabelecimentos de Saúde Brasil**. 2020. Disponível em: [https://cetic.br/media/docs/publicacoes/2/20201123084414/tic\\_saude\\_2019\\_livro\\_eletronico.pdf](https://cetic.br/media/docs/publicacoes/2/20201123084414/tic_saude_2019_livro_eletronico.pdf). Acesso em: 02 set. 2021.

DOMINGUES, Carla Magda Allan Santos *et al.* Vacina Brasil e estratégias de formação e desenvolvimento em imunizações. **Epidemiologia e Serviços de Saúde**, Brasília, v. 28, n. 2, p. 1-4, 03 out. 2019. Disponível em: <https://www.scielo.br/j/ress/a/XRqmb64fFWpBpCCnHCrQjcf/?lang=pt>. Acesso em: 22 ago. 2021.

FEDELI, Ricardo Daniel; POLLONI, Eurico Giulio Franco; PERES, Fernando Eduardo. **Introdução à ciência da computação**. São Paulo: Pioneira Thomson Learning, 2003.

INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA (Rio de Janeiro) (ed.). **Acesso à internet e à televisão e posse de telefone móvel celular para uso pessoal: 2017**. 2018. Disponível em: <https://biblioteca.ibge.gov.br/index.php/biblioteca-catalogo?view=detalhes&id=2101631>. Acesso em: 22 ago. 2021.

LUIZA BARUFI (Brasília). Ministério da Saúde. **Ministério da Saúde esclarece dúvidas sobre campanhas de vacinação**. 2020. Disponível em: <https://www.gov.br/saude/pt-br/assuntos/noticias/em-live-ministerio-da-saude-esclarece-duvidas-sobre-campanhas-de-vacinacao>. Acesso em: 22 ago. 2021.

MOURA, Lívia; CAMARGO, Gustavo. **Impacto econômico e social do Android no Brasil**. 2020. Disponível em: [https://www.bain.com/contentassets/20e0815cfd784b4a8dead63475b42380/v02\\_b-impactos-do-android-no-brasil.pdf](https://www.bain.com/contentassets/20e0815cfd784b4a8dead63475b42380/v02_b-impactos-do-android-no-brasil.pdf). Acesso em: 07 jun. 2021.

Plotkin SL, Plotkin SA. **A short history of vaccination**. In: Plotkin SA, Orenstein WA (eds.). Vaccines. 4th ed. Filadélfia: Saunders; 2004. p. 1-15.

SOMMERVILLE, Ian. **Engenharia de Software**. 9. ed. São Paulo: Pearson Education do Brasil Ltda, 2011. 521 p.

TEMPORÃO, J. G.: **O Programa Nacional de Imunizações (PNI): origens e desenvolvimento**. *História, Ciências, Saúde Manguinhos*, vol. 10 (suplemento 2): 601-17, 2003.

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

### **RF001 - Tela de Demonstração**

Demonstra para o usuário os objetivos e funcionalidades que estarão disponíveis no aplicativo.

### **RF002 - Tela de Cadastro**

Permite ao usuário se cadastrar na aplicação para ter acesso às funcionalidades da mesma.

### **RF003 - Tela de Login**

Permite ao usuário entrar na aplicação após realizar a validação de usuário e senha cadastrados anteriormente no sistema.

### **RF004 - Recuperação de senha**

Permite ao usuário do sistema enviar as instruções de recuperação de senha para o e-mail cadastrado.

### **RF005 - Listagem de Parentes**

Esta tela exibirá para o usuário uma lista contendo todos os parentes cadastrados por ele no aplicativo.

### **RF006 - Cadastro de Parentes**

Esta tela permitirá ao usuário cadastrar um parente na aplicação, mediante os dados de nome e grau de parentesco.

### **RF007 - Editar Parentes**

Esta tela permitirá ao usuário editar um parente na aplicação.

### **RF008 - Remover de Parentes**

Esta tela permitirá ao usuário remover um parente na aplicação.

**RF009 - Cartão de Vacina**

Esta tela exibirá para o usuário uma lista contendo todas as vacinas e doses que podem ser tomadas de acordo com o PNI.

**RF010 - Cadastro de doses**

Nesta tela será possível registrar a data e a dose da vacina tomada pelo usuário.

**RF011 – Alterar Cadastro de doses**

Nesta tela será possível registrar a data e a dose da vacina tomada pelo usuário.

**RF012 – Imprimir Cartão de Vacina**

Na tela de Cartão de Vacina conterà um botão que irá permitir que usuário imprima o cartão de vacina próprio ou do parente selecionado.

**RF013 - Tela Vacinas**

Esta tela exibirá para o usuário uma listagem contendo todos as vacinas inclusas no programa nacional de imunização.

**RF014 - Consultar Vacinas**

Permite consultar as vacinas na Tela de Vacinas utilizando o nome completo da vacina ou apenas partes do mesmo.

**RF015 - Detalhe Vacina**

Ao selecionar uma vacina na Tela de Vacinas, será exibida todas as informações sobre a mesma, como por exemplo: Fabricação, Efeitos adversos, esquema de doses, via de aplicação, indicações e cuidados antes durante e após a vacinação.

**RF016 - Postos de Vacinação**

Esta tela exibirá para o usuário um mapa iterativo contendo todos os postos de vacinação na cidade de Campina Grande, além de exibir uma rota até o mesmo.

**RF017 – Conta de Usuário**

Permite ao usuário editar seus dados pessoais.

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

### **RF001 – Sistema Operacional**

O sistema exibirá rodar em dispositivos moveis com sistema Android na versão 6 ou superior.

### **RF002 – Acesso à internet**

O sistema deverá funcionar normalmente desde que o dispositivo tenha conexão com a internet

### **RF003 - Desenvolvimento**

O sistema deverá ser desenvolvido utilizando somente componente nativos do Android.

### **RF004 – Serviços de Nuvem**

Para serviços de nuvem, sistema deverá usar o conjunto de serviços disponíveis junto ao Firebase Google.

## APÊNDICE C – CARTÃO DE VACINA PARA IMPRESSÃO

### Cartão de Vacina



Parente fictício 1

Vacina	Primeira Dose	Segunda Dose	Terceira Dose
BCG	15/05/2021	-----	-----
Hepatite B	17/05/2021	25/07/2021	Dose não tomada
Penta/DTP	21/05/2021	26/05/2021	31/05/2021
Pneumocócica(conjugada)	25/07/2021	Dose não tomada	Dose não tomada
Rotavirus Humano	25/07/2021	29/07/2021	31/07/2021
Meningocócica Conjugada	24/07/2021	Dose não tomada	-----
Febre Amarela	Dose não tomada	-----	-----
Hepatite A	02/07/2021	-----	-----
Triplíce Viral	25/07/2021	-----	-----
Tetra Viral	23/07/2021	-----	-----
Varicela	Dose não tomada	-----	-----
HPV	25/07/2021	-----	-----
Pneumocócica 23V	25/07/2021	-----	-----
Dupla Adulto	25/07/2021	-----	-----
DTPA	25/07/2021	-----	-----
Influenza	25/07/2021	-----	-----