



**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

THAYNNARA RAIANY UCHÔA GONÇALVES

**CUIDAR:
SISTEMA PARA AUXÍLIO NO TRATAMENTO DE CRIANÇAS
COM MICROCEFALIA EM AMBIENTE DOMICILIAR**

CAMPINA GRANDE - PB

2021

THAYNNARA RAIANY UCHÔA GONÇALVES

**CUIDAR:
SISTEMA PARA AUXÍLIO NO TRATAMENTO DE CRIANÇAS
COM MICROCEFALIA EM AMBIENTE DOMICILIAR**

**Trabalho de Conclusão Curso
apresentado ao Curso Bacharelado em
Ciência da Computação do Centro de
Engenharia Elétrica e Informática da
Universidade Federal de Campina
Grande, como requisito parcial para
obtenção do título de Bacharela em
Ciência da Computação.**

Orientador: Professor Dr. Wilkerson de Lucena Andrade

CAMPINA GRANDE - PB

2021



G635C Gonçalves, Thaynnara Raiany Uchôa.
Cuidar: sistema para auxílio no tratamento de
crianças com microcefalia em ambiente domiciliar. /
Thaynnara Raiany Uchôa Gonçalves. - 2021.

9 f.

Orientador: Prof. Dr. Wilkerson de Lucena Andrade.
Trabalho de Conclusão de Curso - Artigo (Curso de
Bacharelado em Ciência da Computação) - Universidade
Federal de Campina Grande; Centro de Engenharia Elétrica
e Informática.

1. Microcefalia. 2. Tecnologia aplicada à saúde. 3.
Startup. 4. Desenvolvimento de aplicação. 5. Estimulação
motora de crianças. 6. Centro Especializado em
Reabilitação. I. Andrade, Wilkerson de Lucena. II.
Título.

CDU:004(045)

Elaboração da Ficha Catalográfica:

Johnny Rodrigues Barbosa
Bibliotecário-Documentalista
CRB-15/626

THAYNNARA RAIANY UCHÔA GONÇALVES

CUIDAR:

**SISTEMA PARA AUXÍLIO NO TRATAMENTO DE CRIANÇAS
COM MICROCEFALIA EM AMBIENTE DOMICILIAR**

**Trabalho de Conclusão Curso
apresentado ao Curso Bacharelado em
Ciência da Computação do Centro de
Engenharia Elétrica e Informática da
Universidade Federal de Campina
Grande, como requisito parcial para
obtenção do título de Bacharela em
Ciência da Computação.**

BANCA EXAMINADORA:

Professor Dr. Wilkerson de Lucena Andrade

Orientador – UASC/CEEI/UFCG

Professor Dr. Maxwell Guimarães de Oliveira

Examinador – UASC/CEEI/UFCG

Professor Tiago Lima Massoni

Professor da Disciplina TCC – UASC/CEEI/UFCG

Trabalho aprovado em: 20 de Outubro de 2021.

CAMPINA GRANDE - PB

RESUMO (ABSTRACT)

In October 2015, the Health Department of Pernambuco issued a state of emergency statement for an epidemic of microcephaly. In November of the same year, the Zika virus (ZIKV) was identified as responsible for microcephaly in babies. In this context, the CER (Specialized Center in Rehabilitation) of the city of Campina Grande-PB, works in the rehabilitation of children with microcephaly caused by Zika virus (MSC-ZIKV). Part of the treatment is carried out through motor stimulation procedures to be continued in the home environment. After interviews and visits to the CER, it was discovered that this care is not being done properly, with a difference between the prescribed tasks and those actually performed by the caregivers. Such differences can compromise the effectiveness of the treatment and impair the patient's evolution. This work aims to develop an application capable of helping caregivers in the process of motor stimulation of children with MSC-ZIKV. The application was developed on two different platforms, web and mobile. The web application aims to enable CER health professionals to register activities and prescribe them to their patients. The mobile part is used by caregivers, in order to obtain better instructions regarding the activities to be performed in the home environment and the recording of these activities.

Cuidar: sistema para auxílio no tratamento de crianças com microcefalia em ambiente domiciliar

Thaynnara Raiany Uchôa Gonçalves
thaynnara.goncalves@ccc.ufcg.edu.br
Universidade Federal de Campina Grande
Campina Grande, Paraíba

Wilkerson de Lucena Andrade
wilkerson@computacao.ufcg.edu.br
Universidade Federal de Campina Grande
Campina Grande, Paraíba

RESUMO

Em outubro de 2015, a Secretaria de Saúde de Pernambuco emitiu uma declaração de estado de emergência para uma epidemia de microcefalia. Em novembro do mesmo ano foi identificado o vírus da Zika (ZIKV) como responsável pela microcefalia ocorrida em bebês. Nesse contexto, o CER (Centro Especializado em Reabilitação) da cidade de Campina Grande-PB, atua na reabilitação de crianças com microcefalia por Zika vírus (MSC-ZIKV). Parte do tratamento é conduzido por meio de procedimentos de estimulação motora a serem continuados em ambiente domiciliar. Após entrevistas e visitas realizadas ao CER, descobriu-se que esses cuidados não vêm sendo feitos de forma adequada, havendo uma diferença entre as tarefas prescritas e as que são realizadas de fato pelos cuidadores. Tais diferenças podem comprometer a eficácia do tratamento e prejudicar a evolução do paciente. Este trabalho tem como objetivo o desenvolvimento de uma aplicação capaz de auxiliar os cuidadores no processo de estimulação motora de crianças com MSC-ZIKV. A aplicação foi desenvolvida em duas plataformas distintas, sendo elas web e mobile. A aplicação web tem como objetivo possibilitar aos profissionais de saúde do CER o cadastro de atividades e prescrevê-las aos seus pacientes. Já a parte mobile é usada pelos cuidadores, a fim de obter melhores instruções quanto às atividades a serem executadas em ambiente domiciliar e o registro dessas atividades.

Palavras-chave: Cuidar, microcefalia, zika vírus, desenvolvimento de software, desenvolvimento web, desenvolvimento Android, aplicativo, CER, Centro Especializado em Reabilitação, auxílio.

Backend: https://github.com/thaynnara007/Cuidar_backend

frontend: https://github.com/thaynnara007/Cuidar_Admin

mobile: <https://github.com/thaynnara007/App-CER>

ACM Reference Format:

Thaynnara Raiany Uchôa Gonçalves and Wilkerson de Lucena Andrade. 2021. Cuidar: sistema para auxílio no tratamento de crianças com microcefalia em ambiente domiciliar. In *Proceedings of ACM Conference (Conference'17)*. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

Conference'17, July 2017, Washington, DC, USA

© 2021 Association for Computing Machinery.

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM... \$15.00

<https://doi.org/10.1145/nnnnnnn.nnnnnnn>

1 INTRODUÇÃO

O vírus da Zika (ZIKV) é causador de uma doença transmitida por mosquito, identificada pela primeira vez em 1947 no Vale do Zika, em Uganda, e no Brasil apenas em outubro de 2015, quando neuropediatras de Pernambuco emitiram uma declaração de estado de emergência para uma epidemia de microcefalia [7]. As investigações, até então, tinham conclusões preliminares que sugeriam uma relação entre a infecção pelo Zika e a microcefalia, porém era recebida com ceticismo no mundo. Em novembro do mesmo ano, a médica e pesquisadora Adriana Melo, de Campina Grande, na Paraíba, identificou o ZIKV no líquido amniótico de gestantes, trazendo a certeza de que esse vírus era o responsável pela microcefalia ocorrida em bebês [4].

Este distúrbio neurológico influencia a vida da criança como também a dos familiares, sendo muito importante uma instrução adequada aos cuidadores para que possam oferecer à criança uma atenção integral e adequada [23]. Segundo especialistas, as consequências dessa síndrome resultam em um conjunto de sintomas, além da microcefalia fetal ou pós-natal, tais como o atraso no desenvolvimento neuropsicomotor, anormalidades auditivas e visuais, desproporção craniofacial, epilepsia, irritabilidade, dentre outros [1].

É nesse contexto que o Centro Especializado de Reabilitação (CER), da cidade de Campina Grande, atua no processo de reabilitação clínica de crianças com MSC-ZIKV. Além disso, são também prescritos cuidados e procedimentos que devem ser aplicados em ambiente domiciliar, por mães e cuidadoras, visando a continuidade do tratamento e dos avanços obtidos na reabilitação.

Após entrevistas e visitas realizadas ao CER, por estudantes do curso de Design da UFCG, para uma pesquisa de mestrado, descobriu-se que esses cuidados não vêm sendo feitos de forma adequada, havendo uma diferença entre as tarefas prescritas e as que são realizadas de fato pelos cuidadores. A necessidade do apoio de artefatos de tecnologia assistiva em algumas das atividades de estimulação se torna um fator de desestimulação para as cuidadoras na realização dos cuidados prescritos, visto que tais artefatos possuem custo elevado. Somado a isso, as diferenças entre as atividades realizadas e as prescritas podem comprometer a eficácia do tratamento e prejudicar a evolução do paciente.

Com a pandemia do COVID-19 no Brasil e a necessidade de distanciamento social, gerou-se uma dificuldade maior no atendimento às crianças com MSC-ZIKV. Como consequência, as atividades a serem executadas em ambiente domiciliar ganharam ainda mais importância, tornando mais necessário o engajamento do cuidador e uma comunicação clara e objetiva quanto às atividades a serem realizadas no ambiente domiciliar.

É nesse contexto que surgiu uma pesquisa do curso de Design da UFCG, intitulada "APLICATIVO PARA AUXÍLIO E MONITORAMENTO DE ESTIMULAÇÃO MOTORA DE CRIANÇAS COM MICROCEFALIA EM AMBIENTE DOMICILIAR.", cujo objetivo era o desenvolvimento de uma aplicação móvel, de interface amigável e intuitiva, capaz de auxiliar os cuidadores no processo de estimulação motora de crianças com MSC-ZIKV em ambiente domiciliar.

Logo, este trabalho foi realizado em cooperação com a pesquisa acima mencionada. Dessa forma, aos pesquisadores de Design ficou a responsabilidade de entender melhor as necessidades do CER e projetar um aplicativo móvel que pudesse auxiliar na resolução desse problema. Sendo assim, o resultado da pesquisa foi a prototipação das telas desse aplicativo, que definem 7 atividades a serem realizadas no dia-a-dia pelo cuidador junto a criança. Já este trabalho teve o objetivo de desenvolver o aplicativo. Somado a isso, este trabalho acrescenta à solução proposta pela pesquisa de Design ao inserir um site que funcione de forma a administrar o aplicativo *mobile*, possibilitando, dessa forma, que todas as informações presentes neste sejam dinâmicas e controladas pelo CER, como também a criação de um histórico de atividades, possibilitando também o acompanhamento das atividades realizadas por cada paciente do CER.

2 TRABALHOS RELACIONADOS

Em Pernambuco foi desenvolvido, pela Fundação Altino Ventura (FAV) em parceria com a Universidade Federal de Pernambuco (UFPE) e o Tecnologias Estratégicas do Nordeste (Cetene), um aplicativo para monitorar crianças com microcefalia, podendo ser utilizada pelos pais das crianças, profissionais e gestores da saúde pública em Pernambuco, para ajudar no tratamento.

Chamado de Mobicare, o aplicativo contém vários materiais educativos sobre como estimular as crianças, além de funcionar como um prontuário móvel de cada paciente, que é alimentado pelos pais da criança ao inserirem dados de como está a criança em relação a sintomas e ao tratamento. [11].

O Cuidar surgiu oriundo da necessidade do CER, visto o distanciamento social consequente da Pandemia do COVID-19 no Brasil, de melhor elucidar as atividades a serem realizadas no âmbito domiciliar pelos cuidadores, enquanto no ambiente domiciliar, além de monitorar se essas atividades estão sendo realizadas nos momentos corretos. Logo, o Cuidar se diferencia do Mobicare por ser uma aplicação especializada para o CER, como por contar com um site administrativo que permite a inserção rápida de novas atividades ao aplicativo, como também disponibiliza um histórico de atividades realizadas por cada paciente.

3 SOLUÇÃO

Cuidar é um sistema para duas plataformas distintas, sendo elas *web* e *mobile*. A aplicação *web*, tem como objetivo possibilitar aos profissionais do CER o cadastro de pacientes, de atividades e prescrevê-las aos seus pacientes, assim como a obtenção dos dados sobre as atividades realizadas. Já a parte *mobile* é direcionada aos cuidadores, a fim de obter melhores instruções quanto às atividades a serem executadas em ambiente domiciliar e o registro dessas atividades.

Assim, a aplicação *web*, nomeada como Cuidar Admin, está disponível de forma gratuita através de um site para os profissionais

do CER de Campina Grande. Quanto ao aplicativo *mobile*, nomeado apenas de Cuidar, é uma aplicação Android disponibilizada na Google Play, também gratuitamente.

3.1 Funcionalidades

Após conversas com os envolvidos na pesquisa de Design citada anteriormente, de estudo dos artefatos gerados por eles e das entrevistas realizadas com os profissionais do CER, foi possível fazer o levantamento das funcionalidades básicas para auxiliar na resolução do problema em mãos.

3.1.1 Paciente.

A partir do Cuidar Admin é possível o cadastro de novos pacientes, ou seja, a criança com microcefalia, como também ver os pacientes cadastrados e poder removê-los, como ilustrado na Figura 1 e Figura 2. Após realizado o registro, o cuidador da criança consegue efetuar *login* no aplicativo Android Cuidar ao fazer uso dos dados cadastrados, assim como possibilitar a recuperação de senha via email e a atualização de seus dados se desejar, conforme a Figura 3.

Figura 1: Registro de paciente.

Nome	Telefone	E-mail	Inscrição	Ações
Thaynara Gonçalves	081 336.144.10	thaynara.goncalves@ccc.ufcg.edu.br	02090700827	DETALHES REMOVER
Gabriel da Silva	081 323.504.85	cuidador_da_silva@gmail.com	02090884554	DETALHES REMOVER

Figura 2: Listagem dos pacientes e a opção de removê-los.

3.1.2 Categoria.

As categorias são usadas para melhorar a busca por certos tipos de atividades, logo, em uma categoria estão contidas várias atividades de um mesmo tipo. Sendo assim, no Cuidar Admin é possível o cadastro de novas categorias, atualização e remoção das mesmas e listagem das categorias existentes, como ilustrado nas Figuras 4 e 5. Já aos pacientes é possibilitado o acesso às categorias via aplicativo Cuidar, como ilustrado na Figura 6.

3.1.3 Atividade.

A partir do Cuidar Admin é possível o cadastro de novas atividades a serem realizadas pelos pacientes, atualização e remoção das mesmas e listagem das atividades de uma categoria, vide Figura 7 e 8. Dessa forma, ao paciente é possível acessar as atividades de uma categoria por meio da aplicação *mobile* Cuidar, como ilustrado na Figura 9.



Figura 3: Login no aplicativo móvel Cuidar.



Figura 6: Listagem das categorias no aplicativo móvel Cuidar.

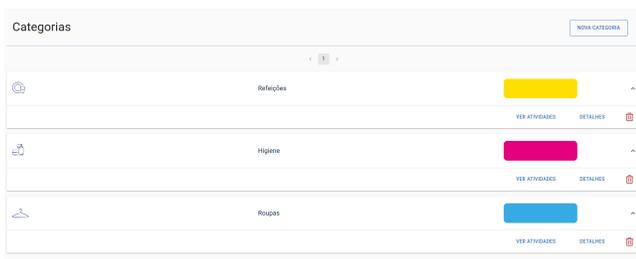


Figura 4: Listagem das categorias e a opção de removê-las.

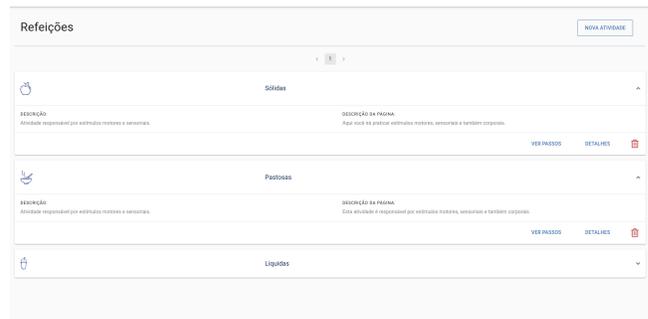


Figura 7: Listagem das atividades e a opção de removê-las.

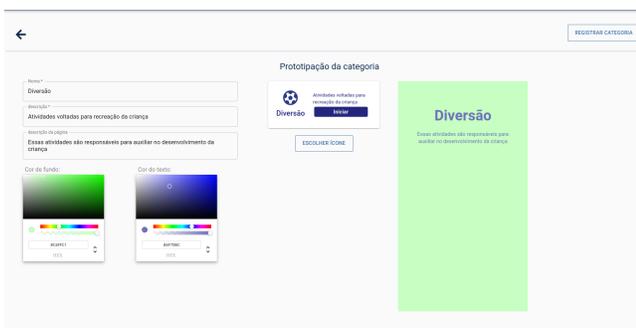


Figura 5: Criação de uma categoria.

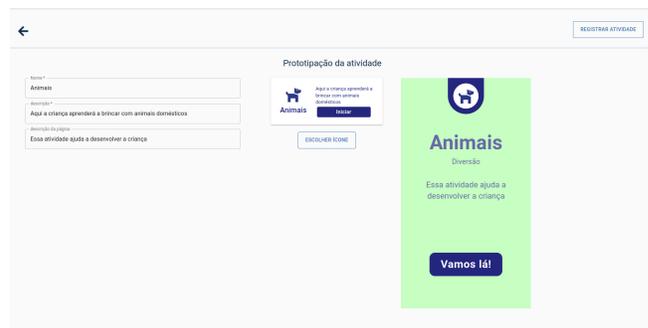


Figura 8: Criação de uma atividade.

3.1.4 Criar etapa.

As etapas de uma atividade equivalem ao passo-a-passo que deve ser seguido para a conclusão da atividade. Logo, é possível também, através do Cuidar Admin, a adição de novas etapas à uma atividade, atualização e remoção das mesmas e a listagem das etapas de uma atividade, como ilustrado nas Figuras 10 e 11. Dessa forma, ao

paciente é possível acessar as etapas de uma atividade por meio do Cuidar, conforme Figura 12.



Figura 9: Listagem de atividades no aplicativo móvel Cuidar.



Figura 12: Visualização de uma etapa no aplicativo móvel Cuidar.

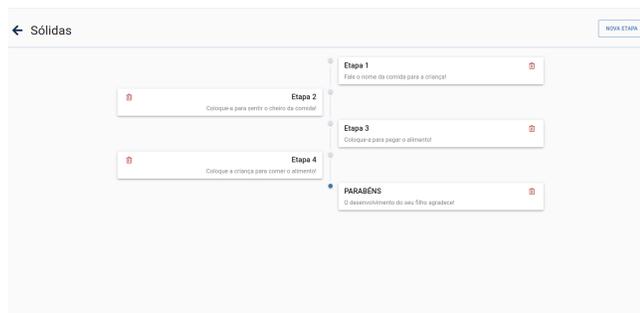


Figura 10: Listagem das etapas e a opção de removê-las.

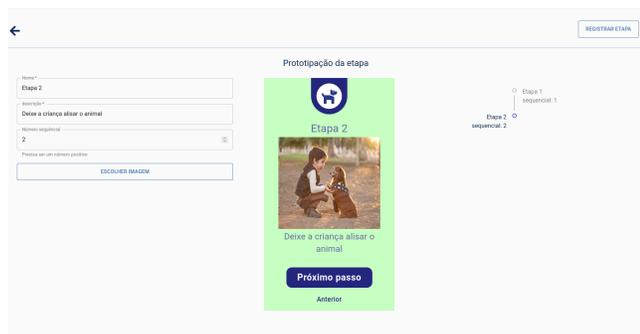


Figura 11: Criação de uma etapa.

3.1.5 Histórico do paciente.

Aos profissionais do CER é possível acompanhar as atividades realizadas por seus pacientes. Sendo assim, o Cuidar Admin possibilita

a visualização de um histórico das atividades realizadas pelos pacientes. Funcionalidade ilustrada na Figura 13.

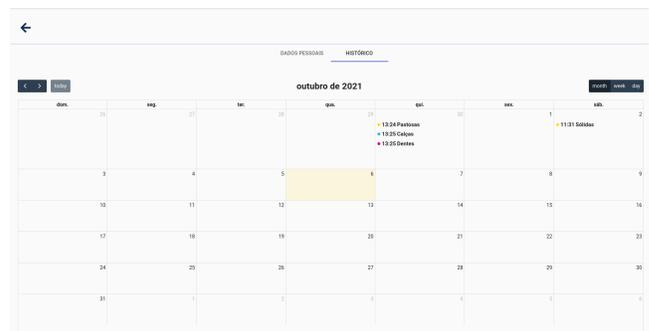


Figura 13: Histórico de um paciente.

3.1.6 Criar usuário.

Por fim, apenas usuários cadastrados no Cuidar Admin podem acessá-lo, vide Figura 14. Dessa forma, é possível o registro de novos usuários e a atribuição de permissões a esses, de forma a barrar ou liberar o acesso de algumas funcionalidades aos usuários. Além disso, é possível a remoção de usuários, para aqueles que têm permissão para isso, e a listagem dos usuários existentes, conforme ilustrado nas Figuras 15, 16 e 17.

3.2 Arquitetura

Uma das arquiteturas para sistemas webs mais comumente usada na atualidade é a de Cliente-Servidor. Esta é definida por duas entidades, o Cliente, que é aquele que está em busca de alguma informação,



Figura 14: Login no sistema

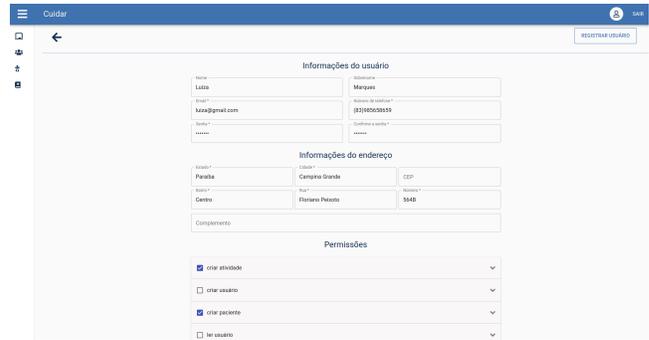


Figura 17: Criação de um usuário.

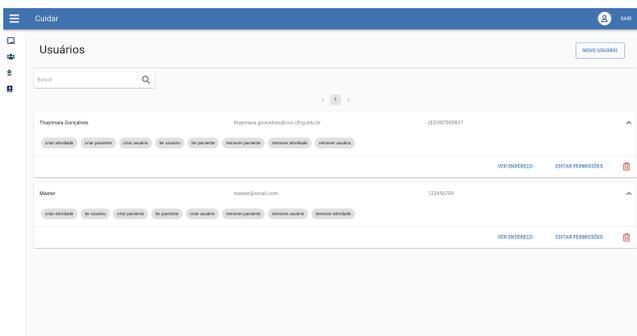


Figura 15: Listagem dos usuários e a opção de removê-los.

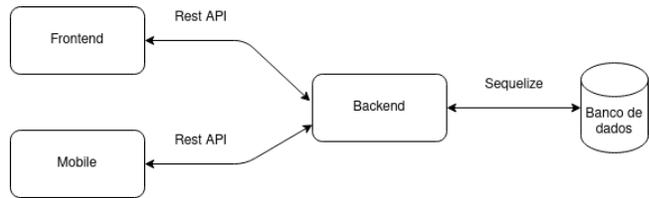


Figura 18: Arquitetura da solução.

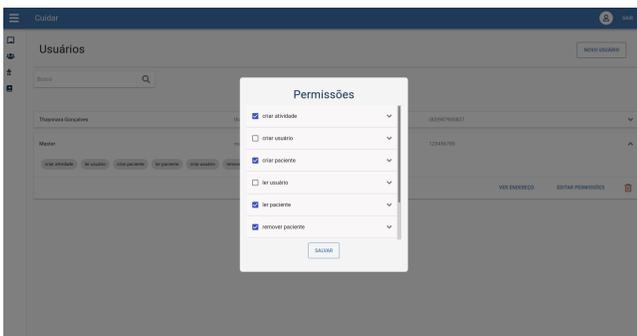


Figura 16: Opção de editar as permissões de um usuário.

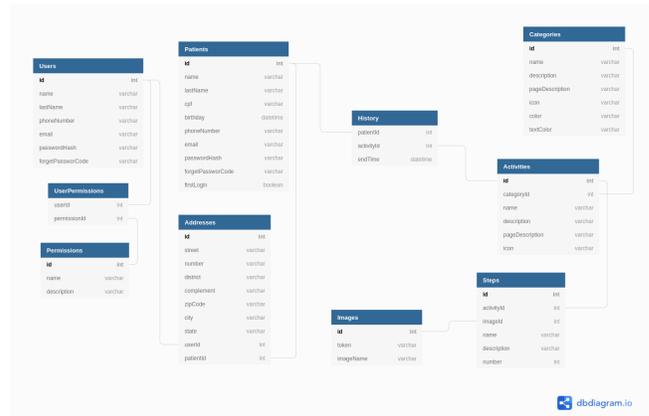


Figura 19: Modelo do banco de dados.

e o Servidor, aquele que provê informação. Dessa forma, quando o Cliente quer alguma informação do Servidor é então estabelecida uma conexão usando o protocolo HTTP, assim o Cliente manda um requisição para o Servidor, em uma linguagem que seja compreendida pelas duas entidades, que então retorna a resposta apropriada [25].

Logo, essa foi a arquitetura usada, com dois clientes, a aplicação *mobile* e o *front-end*, e o servidor, o *back-end*, conforme ilustrado na Figura 18. Para a linguagem de comunicação entre as entidades foi usado o JSON(JavaScript Object Notation) [18].

3.2.2 Back-end.

O servidor foi utilizado como um intermediário entre o cliente e o banco de dados. Dessa forma, o *backend* é responsável pelas interações com o banco de dados, cabendo a ele a adição, remoção ou atualização dos dados, como também selecionar e tratar quais informações serão repassadas ao cliente. Nesse contexto, para comunicação entre essas entidades, o servidor foi construído usando

o padrão REST API [16], por proporcionar um design moderno e simples, como também pela experiência prévia com esse tipo de aplicação.

Para sua construção foi utilizado o Node.js [21] como ambiente de execução junto do *framework* Express [9] para estruturar os recursos servidos no padrão REST. Além disso, foi utilizado o ORM Sequelize [24] para modelagem dos dados e comunicação com o PostgreSQL. Para o armazenamento de imagens foi utilizado o Cloud Storage do Firebase [15]. Por fim, para o processo de desenvolvimento e eventual deploy no Heroku [17], fez-se uso do Docker [5]. Todas as tecnologias foram escolhidas por terem uma versão gratuita ou pela experiência prévia com o uso das ferramentas.

Por se tratar de uma REST API, o *backend* foi estruturado em um padrão comumente seguido pela comunidade de desenvolvedores. Logo, a aplicação pode ser dividida em sete diretórios principais: *controllers*, *middlewares*, *models*, *routes*, *services*, *util* e *email*. Por conta da quantidade de arquivos necessários ao desenvolvimento, na Figura 20 é exposto os diretórios omitindo os arquivos de código JavaScript, porém a arquitetura não é comprometida.

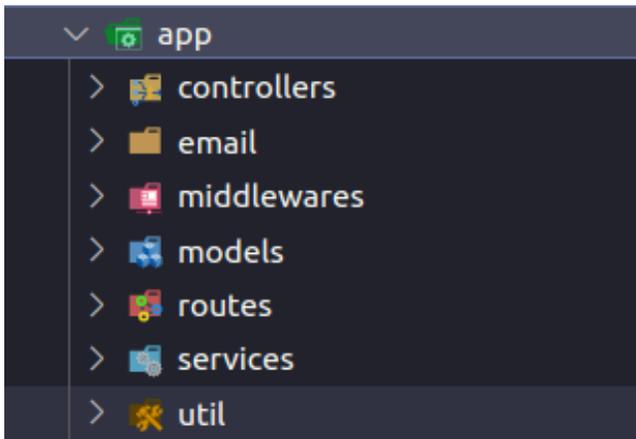


Figura 20: Estrutura dos diretórios do backend.

Os *routes* indicam os *endpoints* da API, definindo os métodos HTTP [6] em cada um, os *middlewares* a serem utilizados e qual *controller* utilizar para o respectivo *endpoint*.

Os *middlewares* são um conjunto de métodos utilizados para interceptar as requisições. Dessa forma, quando chega uma requisição e esta é casada com um *endpoint*, se este fizer uso de *middlewares*, estes são executados antes da requisição ser passada ao respectivo *controller*.

Os *controllers*, por sua vez, são um conjunto de módulos que contém as funções que solucionam as chamadas feitas a API. Nesse contexto, o *controller* é responsável por tratar os dados que vem nas requisições, delegar algum processamento de dados necessário ou alguma busca ao *database* ao seu *service*, e então tratar os dados recebidos por esse e retornar as informações ao cliente.

Sendo assim, os *services* são um conjunto de métodos utilizados para o processamento mais específico de alguma informação relativa a uma entidade do banco de dados ou para fazer requisições ao mesmo via *models*.

Dessa forma, os *models* são um conjunto de modelos que definem as entidades do banco de dados, sendo a partir dos *models* feita a conexão com as tabelas, no *database*, que os mesmos representam.

Por fim, no *util* estão um conjunto de funções genéricas e utilitárias usadas em mais de um arquivo, com intuito de melhorar a legibilidade e reuso de código, já no diretório *email* é encontrado os templates de emails utilizados. O fluxo de uma requisição dentro do *backend* é melhor ilustrado na Figura 21.

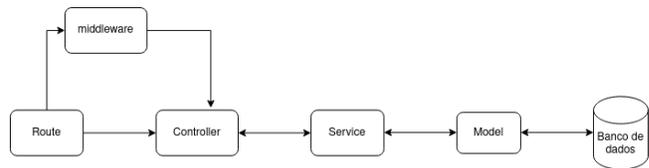


Figura 21: Fluxo da requisição.

3.2.3 Front-end.

O *frontend* foi feito para ser acessível e usado pelos profissionais do CER, logo, para esse cliente foi escolhido uma interface *web*, de cunho administrativo, sendo a partir dele que os profissionais interagem com o *backend*.

O *frontend* foi desenvolvido usando React [10], uma biblioteca JavaScript para criação de interfaces de usuário. Esta tecnologia foi escolhida por permitir um encapsulação em componentes de forma intuitiva e mais próxima do JavaScript, sendo outro fator a experiência prévia com a ferramenta. Além disso, para facilitar no desenvolvimento foi utilizado uma biblioteca de componentes, o Material-UI [19], assim como uma biblioteca de ícones, o FontAwesome [2]. Por fim, para facilitar na visualização do histórico de um paciente, foi também incorporado o FullCalendar [3], uma biblioteca para uso de calendários.

Para o *deploy* da aplicação foi utilizado o Docker, para a contêinerização, em conjunto com o Nginx [20], para o balanço de carga e o Heroku para a hospedagem.

O *frontend* pode ser dividido em quatro diretórios principais: *api*, *components*, *pages* e *utils*, como ilustra a Figura 22, porém, por conta da quantidade de arquivos necessários ao desenvolvimento, é omitindo os arquivos de código JavaScript, mas sem comprometer a arquitetura.

No pacote *api* contém uma série de métodos responsáveis por fazer as requisições ao *backend*, tornando mais fácil a manutenção caso alguma dessas requisições mudem de formato. Já nos *components* estão vários componentes menores e reusáveis que são utilizados em uma ou mais páginas. Sendo assim, em *pages* é onde estão as páginas que serão renderizadas, e por fim, em *utils* estão um conjunto de funções genéricas e utilitárias usadas em mais de um arquivo, com intuito de melhorar a legibilidade e reuso de código.

3.2.4 Mobile.

O aplicativo *mobile* foi feito para ser acessível e usado pelos cuidadores de crianças com microcefalia que fazem tratamento junto ao CER. Logo, para esse cliente foi escolhido uma interface mais amigável e intuitiva, disponível pelo celular, para facilitar o uso no âmbito domiciliar.

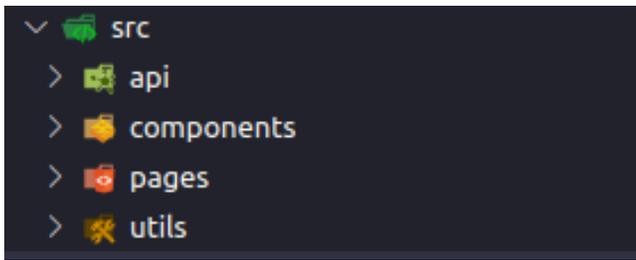


Figura 22: Estrutura dos diretórios do frontend.

A aplicação móvel teve seu foco de desenvolvimento voltado para o Android, por se tratar de um sistema operacional que está presente em cerca de 95% dos *smartphones* do Brasil, ser de fácil uso e estar disponível em diferentes modelos e preços, sendo possível atingir um público maior [28].

Sendo assim, o aplicativo foi desenvolvido usando a IDE Android Studio [14], por ser de fácil configuração e facilitar muito o processo de instalação dos SDKs do Android e no de desenvolvimento. Somado a isso foi utilizado o Genymotion [12], um simulador virtual de dispositivos Android, permitindo o uso da aplicação durante o processo. Por fim, foi utilizado também a biblioteca Retrofit [27], para fazer as requisições ao *backend*, e Picasso [26], para lidar com o *download* de imagens.

A estrutura do aplicativo pode ser dividida em dois diretórios principais: *java* e *res*. Em *java* é onde ficam os arquivos referentes a código Java, que implementam uma série de classes e interfaces que definem o comportamento da aplicação, conforme ilustrado na Figura 23. Já o diretório *res* é conjunto de arquivos, geralmente XML, que define a parte visual do aplicativo, conforme ilustrado na Figura 24. Por conta da quantidade de arquivos necessários ao desenvolvimento, é omitindo alguns arquivos de código Java e XML, mas sem comprometer a estrutura.

Ao final do desenvolvimento o aplicativo Cuidar foi disponibilizado no Google Play, para *download* gratuito.

4 SISTEMA EM USO

Antes do sistema ser utilizado por usuários reais, primeiramente o mesmo foi testado manualmente a cada tarefa feita e comportamento adicionado, removido ou atualizado. Além disso, depois de pronto todos os fluxos da aplicação foram testados novamente, de forma manual.

4.1 Metodologia

Foi realizada uma apresentação, no CER, para os designers e alguns dos profissionais, mostrando o resultado do desenvolvimento, tanto a parte *mobile* quanto a *web*. Depois, foi feito o *download* do aplicativo, por parte dos designers e profissionais do CER, e disponibilizado o *link* para o site do Cuidar Admin e uma conta para acesso.

Sendo assim, foi então pedido aos designers e aos profissionais que respondessem um formulário, criado usando o Google Forms, após o uso da aplicação. Esse formulário consiste de perguntas abertas e fechadas, estas podem variar em respostas de sim ou não,

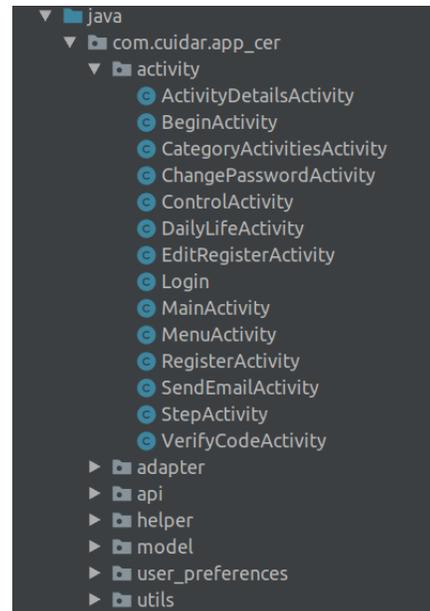


Figura 23: Estrutura dos diretórios do aplicativo mobile, diretório java.

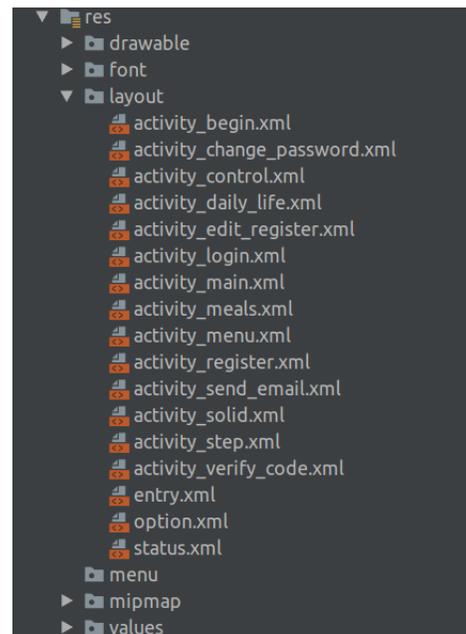


Figura 24: Estrutura dos diretórios do aplicativo mobile, diretório res.

ou com opções que variam em uma escala de 1 a 5. Logo, as questões foram:

4.1.1 Questões referentes ao aplicativo móvel Cuidar.

- (1) “O quão você avalia a dificuldade de usar o aplicativo?": onde 1 é muito difícil e 5 muito fácil.
- (2) “O quão você concorda que as atividades estão estruturadas de forma intuitiva?": onde 1 é discordo completamente e 5 concordo completamente.
- (3) “O quão você considera que as etapas de realização de uma atividade estão bem estruturadas?": onde 1 é muito ruim e 5 muito bem.
- (4) “Qual sua satisfação com o aplicativo?": onde 1 é pouco satisfeito e 5 muito satisfeito.
- (5) “Acredita que está faltando algo no aplicativo?": sendo as respostas possíveis 'Sim' ou 'Não'.
- (6) “Em caso afirmativo na pergunta anterior, o que está faltando?": pergunta de resposta aberta.

4.1.2 Questões referentes ao site Cuidar Admin.

- (1) “O quão você avalia a dificuldade de usar o site?": onde 1 é muito difícil e 5 muito fácil.
- (2) “O quão você avalia a dificuldade para criar ou atualizar uma atividade já existente?": onde 1 é muito difícil e 5 muito fácil.
- (3) “O quão você avalia a forma de visualizar o histórico de um paciente?": onde 1 é muito ruim e 5 muito bom.
- (4) “Qual sua satisfação com o site?": onde 1 é pouco satisfeito e 5 muito satisfeito.
- (5) “Acredita que está faltando algo no aplicativo?": sendo as respostas possíveis 'Sim' ou 'Não'.
- (6) “Em caso afirmativo na pergunta anterior, o que está faltando?": pergunta de resposta aberta.

4.1.3 Questões referentes ao uso em conjunto.

- (1) “Em relação ao uso em conjunto do Aplicativo Android Cuidar e do site Cuidar Admin, o quão você acredita que esses sistemas ajudam a resolver o problema da continuidade do tratamento em âmbito domiciliar e o acompanhamento do mesmo?": onde 1 é não ajuda e 5 resolve o problema.
- (2) “Gostaria de deixar algum comentário?": pergunta de resposta aberta.

4.2 Resultados

Ao total, foram coletadas 9 respostas. Após a análise destas é possível dizer, quanto ao aplicativo móvel Cuidar e o site Cuidar Admin, que os mesmo tem uma usabilidade e estruturação satisfatória, conforme ilustra a Figura 25, que é uma média das respostas às perguntas 1,2,3 e 4.

No entanto, quando perguntado se está faltando algo nas aplicações, o aplicativo móvel Cuidar teve 33% das respostas como 'Sim' e o site Cuidar Admin 22%. Logo, também é possível concluir que as aplicações não estão completas e são passíveis de melhorias.

Nesse contexto, ao analisar as respostas relativas a pergunta 6, é visível as melhorias desejadas, sendo elas:

- A possibilidade de adicionar *gifs*, para que melhor simulam as atividades.
- Possibilitar um contato mais direto com o usuário a partir de *feedbacks*
- tornar a interface mais intuitiva de forma gráfica.

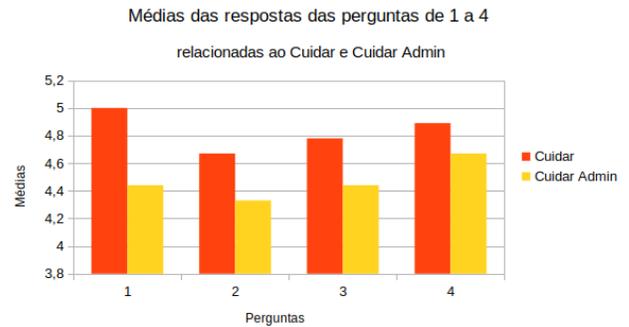


Figura 25: Gráfico vertical de barras que mostra as médias entre as repostas das perguntas 1 a 4 para cada aplicação.

Quanto ao quesito de ajudar na resolução do problema, para a pergunta que isso questiona, obteve-se resultados positivos, logo, conclui-se que este trabalho não resolve o problema completamente, porém auxilia na sua solução, como ilustra o gráfico da Figura 26.

Por fim, seguem os comentários considerados o mais positivo e o mais negativo, dentro os deixado no formulário.

- “Gosto da forma como o site complementa o aplicativo no seu contexto geral, tanto em relação aos cuidadores como em relação aos profissionais, que terão mais autonomia com o uso do site para administração das atividades existentes e das que ainda serão adicionadas.”
- “Parabéns! Está muito bem estruturado, apenas melhorar a questão da interface gráfica no futuro, a fim de aperfeiçoá-lo mais ainda. ”

Em relação ao uso em conjunto do Aplicativo Android Cuidar e do site Cuidar Admin, o quão você acredita que esses sistemas ajudam a resolver o problema da continuidade do tratamento em âmbito domiciliar e o acompanhamento do mesmo?

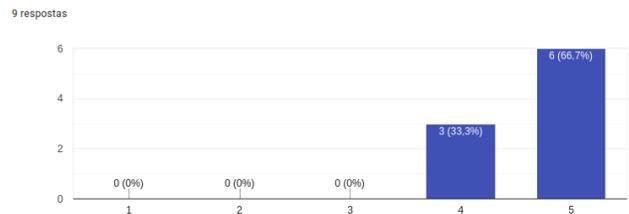


Figura 26: Gráfico vertical de barras que mostra as respostas para a pergunta relativa a resolução do problema, pelas aplicações.

5 EXPERIÊNCIA

Para o desenvolvimento deste trabalho foi essencial o conhecimento de algumas tecnologias, conceitos e habilidades adquiridas durante a graduação. Por tanto, algumas disciplinas foram importantes para esse processo, sendo elas: Programação II, Laboratório de Programação II, Estrutura de Dados, Banco de Dados I, Engenharia de Software, Metodologia Científica e Princípios de Desenvolvimento Web.

5.1 Processo de desenvolvimento

O processo de desenvolvimento se baseou nas etapas do Scrum, com sprints de uma semana. Inicialmente foram levantadas as funcionalidades necessárias ao sistema e adicionadas ao *backlog*. Em seguida foi atribuída uma prioridade (Baixa, Média ou Alta) para cada uma e então foi ordenado o *backlog* de forma que as funcionalidades de prioridade mais alta fossem feitas primeiro. Nesse contexto, essas foram: cadastrar usuário, cadastrar paciente, autenticação, cadastro de categoria, cadastro de atividade, cadastro de etapa e as respectivas visualizações, no Cuidar e no Cuidar Admin, para cada uma das entidades criadas nas funcionalidades listadas. Estas foram selecionadas como de alta prioridade por ser o necessário para o aplicativo mobile idealizado pela pesquisa dos alunos de Design o qual esse trabalho contribui, como também para o funcionamento em conjunto com a aplicação web Cuidar Admin, proposto por esse trabalho.

Com isso, foi decidido as tecnologias a serem utilizadas e iniciado um repositório no Github [13] para cada uma das aplicações: *backend*, *frontend* e *mobile*. Em seguida, para os dois primeiros repositórios foi feita a configuração de um analisador estático de código, o Eslint [8], com intuito de manter a qualidade do código e seguir determinados padrões e diretrizes que são considerados boas práticas. Após isso, foi então criado um Dockerfile para esses repositórios, para facilitar no eventual deploy das aplicações.

Dessa forma, teve início o processo de desenvolvimento, de forma que para cada tarefa sendo realizada fosse feita primeiro no *backend*, depois no *frontend* e em seguida no *mobile*, levando em consideração que as telas para esta última foram feitas seguindo a prototipagem realizada pela pesquisa de Design já mencionada. Cada tarefa era codificada em uma *branch* e após sua finalização era então feito um *pull request* para a *master*.

Por fim, ao final do desenvolvimento foi então feito o deploy do *backend* e *frontend* no Heroku, usando containers. Já para a parte *mobile* foi gerado uma APK assinada e então disponibilizada no Google Play.

5.2 Desafios

O primeiro grande desafio para a conclusão deste trabalho foi o tempo, por se tratar de um sistema que conta com uma versão mobile e outra web, e pela grande quantidade de funcionalidades básicas para seu funcionamento.

O segundo desafio foi o aprendizado de como desenvolver para Android em tempo de projeto, o que pesou no quesito tempo.

Por fim, o terceiro desafio foi a validação do sistema, por requerer o uso por usuários muito específicos (os profissionais do CER de Campina Grande e os designers envolvidos na pesquisa) com pouca disponibilidade.

5.3 Trabalhos futuros

Apesar do Cuidar permitir que os profissionais do Centro Especializado em Reabilitação de Campina Grande acompanhe o progresso das atividades realizadas em âmbito domiciliar por seus pacientes, ainda não há uma forma de contato mais direta entre cuidador e profissional, de forma que permita um cuidador enviar um dúvida ao CER e ser respondido, por meio da aplicação móvel Cuidar. Dessa

forma, há a intenção e planejamento para o incremento mencionado. Além de fazer um teste mais amplo da aplicação, com mais profissionais do CER junto aos seus pacientes.

6 AGRADECIMENTOS

Meu principal agradecimento vai para minha mãe que fez de tudo para que eu pudesse focar só nesse trabalho e não gastasse tempo com outras coisas. O próximo agradecimento vai para meu namorado e amigos que tiveram muita paciência comigo durante o desenvolvimento deste trabalho.

REFERÊNCIAS

- [1] TEIXEIRA GA Enders B.C. Carvalho ALB Dantas DNA Carvalho GAFIL Silva AN. 2020. Análise do conceito síndrome congênita pelo Zika vírus. *SeiELO Brazil* 25, 2 (Fev 2020). <https://doi.org/10.1590/1413-81232020252.30002017>
- [2] Font Awesome. [n. d.]. Get vector icons and social logos on your website with Font Awesome, the web's most popular icon set and toolkit. Retrieved Outubro 5, 2021 from <https://fontawesome.com/>
- [3] Full Calendar. [n. d.]. The Most Popular JavaScript Calendar. Retrieved Outubro 5, 2021 from <https://fullcalendar.io/>
- [4] Ministério da Saúde. 2017. *Vírus Zika no Brasil: a resposta do SUS*. Secretaria de Vigilância em Saúde, esplanada dos Ministérios, bloco G, edifício Sede, sala 156. 9–18 pages. http://biblioteca.cofen.gov.br/wp-content/uploads/2017/08/virus_zika_brasil_resposta_sus.pdf
- [5] Docker. [n. d.]. What is a container? Retrieved Outubro 5, 2021 from <https://www.docker.com/resources/what-container>
- [6] MDN Web Docs. [n. d.]. Uma visão geral do HTTP. Retrieved Outubro 5, 2021 from <https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Overview>
- [7] A S Oliveira Melo G Malinger R Ximenes P O Szejnfeld S Alves Sampaio e A M Bispo de Filippis. 2016. Zika virus intrauterine infection causes fetal brain abnormality and microcephaly: tip of the iceberg? *Ultrasound in Obstetrics & Gynecology* 47, 1 (Jan. 2016), 6–7. <https://doi.org/10.1002/uog.15831>
- [8] Eslint. [n. d.]. Find and fix problems in your JavaScript code. Retrieved Outubro 7, 2021 from <https://eslint.org/>
- [9] Express. [n. d.]. Express: framework web rápido, flexível e minimalista para Node.js. Retrieved Outubro 5, 2021 from <https://expressjs.com/pt-br/>
- [10] Facebook. [n. d.]. React: uma biblioteca JavaScript para criar interfaces de usuário. Retrieved Outubro 5, 2021 from <https://pt-br.reactjs.org/>
- [11] G1. 2019. Aplicativo monitora crianças com microcefalia para facilitar tratamento em Pernambuco. Retrieved Outubro 8, 2021 from <https://g1.globo.com/pe/pernambuco/noticia/2019/03/17/aplicativo-monitora-criancas-com-microcefalia-para-facilitar-atendimento-em-pernambuco.ghtml>
- [12] Genymotion. [n. d.]. Android as a Service. Retrieved Outubro 6, 2021 from <https://www.genymotion.com/>
- [13] Github. [n. d.]. Github. Retrieved Outubro 7, 2021 from <https://github.com/>
- [14] Google. [n. d.]. Android Studio provides the fastest tools for building apps on every type of Android device. Retrieved Outubro 6, 2021 from <https://developer.android.com/studio>
- [15] Google. [n. d.]. Cloud Storage para Firebase. Retrieved Outubro 5, 2021 from <https://firebase.google.com/docs/storage>
- [16] Red Hat. 2020. O que é API REST? Retrieved Outubro 5, 2021 from <https://www.redhat.com/pt-br/topics/api/what-is-a-rest-api>
- [17] Heroku. [n. d.]. What is Heroku? Retrieved Outubro 5, 2021 from <https://www.heroku.com/about>
- [18] JSON. [n. d.]. Introducing JSON. Retrieved Outubro 3, 2021 from <https://www.json.org/json-en.html>
- [19] Mui. [n. d.]. MATERIAL-UI: react components for faster and easier web development. Retrieved Outubro 5, 2021 from <https://v4.mui.com/>
- [20] NGINX. [n. d.]. NGINX Service Mesh. Retrieved Outubro 5, 2021 from <https://www.nginx.com/>
- [21] Node.js. [n. d.]. About Node.js. Retrieved Outubro 5, 2021 from <https://nodejs.org/en/about/>
- [22] PostgreSQL. [n. d.]. PostgreSQL: The World's Most Advanced Open Source Relational Database. Retrieved Outubro 3, 2021 from <https://www.postgresql.org/>
- [23] E.F. Cardoso L.C.C. Oliveira e C.C.C. Santos D.B.C. Prado L.O.M. Silva R.S. Silva. 2019. Sensibilização das mães de crianças com microcefalia na promoção da saúde de seus filhos. *SeiELO Brazil* (2019). <https://doi.org/10.1590/S1980-220X2018022903491>
- [24] Sequelize. [n. d.]. Sequelize. Retrieved Outubro 5, 2021 from <https://sequelize.org/master/>
- [25] Virender Singh. 2021. Client Server Architecture and HTTP Protocol. Retrieved Outubro 3, 2021 from <https://toolsqa.com/client-server/client-server-architecture-and-http-protocol/>

- [26] Square. [n. d.]. A powerful image downloading and caching library for Android. Retrieved Outubro 6, 2021 from <https://square.github.io/picasso/>
- [27] Square. [n. d.]. A type-safe HTTP client for Android and Java. Retrieved Outubro 6, 2021 from <https://square.github.io/retrofit/>
- [28] G1 Thiago Lavado. 2019. Em 10 anos no Brasil, Android foi de 2 smartphones para sistema operacional dominante do mercado. Retrieved Outubro 6, 2021 from <https://g1.globo.com/economia/tecnologia/noticia/2019/11/26/ha-10-anos-no-brasil-android-foi-de-2-smartphones-para-sistema-operacional-dominante-do-mercado.ghtml>