



INPI INSTITUTO
NACIONAL
DA PROPRIEDADE
INDUSTRIAL
Assinado
Digitalmente

REPÚBLICA FEDERATIVA DO BRASIL
MINISTÉRIO DA ECONOMIA
INSTITUTO NACIONAL DA PROPRIEDADE INDUSTRIAL
DIRETORIA DE PATENTES, PROGRAMAS DE COMPUTADOR E TOPOGRAFIAS DE CIRCUITOS INTEGRADOS

Certificado de Registro de Programa de Computador

Processo Nº: **BR512020000400-8**

O Instituto Nacional da Propriedade Industrial expede o presente certificado de registro de programa de computador, válido por 50 anos a partir de 1º de janeiro subsequente à data de 11/02/2020, em conformidade com o §2º, art. 2º da Lei 9.609, de 19 de Fevereiro de 1998.

Título: MiAu

Data de publicação: 11/02/2020

Data de criação: 30/06/2019

Titular(es): UNIVERSIDADE DO ESTADO DO RIO GRANDE DO NORTE - UERN

Autor(es): ÁLVARO GABRIEL GOMES DE OLIVEIRA; SEBASTIÃO EMÍDIO ALVES FILHO; JOÃO PAULO SILVA DE MOURA; DARAÉDNA FERREIRA MENEZES; ELISA DE FÁTIMA ANDRADE SOARES; ANDRÉ LUIZ LIMA QUEIROZ; EXLLEY CLEMENTE DOS SANTOS

Linguagem: JAVA SCRIPT

Campo de aplicação: IF-10; MA-01; SD-01

Tipo de programa: AP-01; FA-01; GI-01

Algoritmo hash: SHA-512

Resumo digital hash:

f97414818df0892fb7fd0b345576f117fd334c81a9103da60ac1e1fdebadbc71c276c5182b481d56992a8c2be8e3eb0d625
b79c01f3b9f528ac36bce84244926

Expedido em: 10/03/2020

Aprovado por:

Helmar Alvares

Chefe da DIPTO - Portaria/INPI/DIRPA Nº 09, de 01 de julho de 2019

UNIVERSIDADE DO ESTADO DO RIO GRANDE DO NORTE
FACULDADE DE CIÊNCIAS EXATAS E NATURAIS
DEPARTAMENTO DE INFORMÁTICA

API MIAU: API REST PARA UMA PLATAFORMA HÍBRIDA VOLTADA À
ADOÇÃO DE ANIMAIS

MOSSORÓ – RN
FEVEREIRO/2020

UNIVERSIDADE DO ESTADO DO RIO GRANDE DO NORTE
FACULDADE DE CIÊNCIAS EXATAS E NATURAIS
DEPARTAMENTO DE INFORMÁTICA

DARAÉDNA FERREIRA MENEZES

API MIAU: API REST PARA UMA PLATAFORMA HÍBRIDA VOLTADA À
ADOÇÃO DE ANIMAIS

Relatório apresentado ao curso de Ciência da Computação da Universidade do Estado do Rio Grande no norte como requisito da disciplina de Trabalho de Diplomação, sob a orientação do Prof. Dr. Sebastião Emidio Alves Filho e coorientação de Exlley Clemente dos Santos.

MOSSORÓ – RN
FEVEREIRO/2020

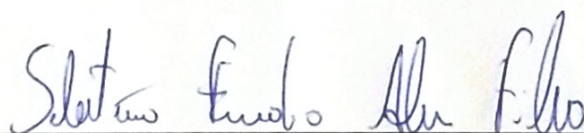
DARAÉDNA FERREIRA MENEZES

API MIAU: API REST PARA UMA PLATAFORMA HÍBRIDA VOLTADA À ADOÇÃO DE ANIMAIS

Monografia apresentada como pré-requisito para a obtenção do título de Bacharel em Ciência da Computação da Universidade do Estado do Rio Grande do Norte – UERN, submetida à aprovação da banca examinadora composta pelos seguintes membros:

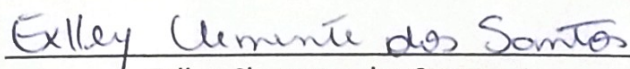
Aprovada em: 20/02/2020

Banca Examinadora




Prof. Dr. Sebastião Emídio Alves Filho

Universidade do Estado do Rio Grande do Norte - UERN



Exley Clemente dos Santos

Universidade do Estado do Rio Grande do Norte - UERN



Prof. Dr. Francisco Chagas de Lima Júnior

Universidade do Estado do Rio Grande do Norte - UERN



Prof. Dr. Rômulo Wladimir de Lima

Universidade do Estado do Rio Grande do Norte - UERN

SUMÁRIO

1. Introdução	4
2. Objetivos	4
3. Metodologia	4
4. Descrição do sistema	5
5. Resultados	9
6. Referências	9

1. INTRODUÇÃO

Conforme os últimos números divulgados pela Organização Mundial da Saúde (OMS) só no Brasil existiam mais de 30 milhões de animais abandonados, entre 10 milhões de gatos e 20 milhões de cães. Quando se trata de cidades de grande porte, estimava-se que para cada cinco habitantes há um cachorro, destes, 10% estão abandonados (ANDA, 2014). Diante esse cenário de abandono tem-se como consequência diversas situações que impactam diretamente a sociedade, como a disseminação de doenças, problemas no trânsito e maus tratos dos animais, riscos que atingem tanto os animais, quanto as pessoas.

Visando solucionar esse problema de abandono, foi proposto o desenvolvimento de uma plataforma para destinar os animais a um novo lar. O Sistema MiAu, é composto por um aplicativo *mobile* e uma aplicação Web. Devido a cada plataforma ter diferentes especificidades para comunicação, armazenamento e recuperação de informações, optou-se pela utilização de uma API (*Application Programming Interface*) REST (*Representational State Transfer*) para gerir as regras de negócio da aplicação.

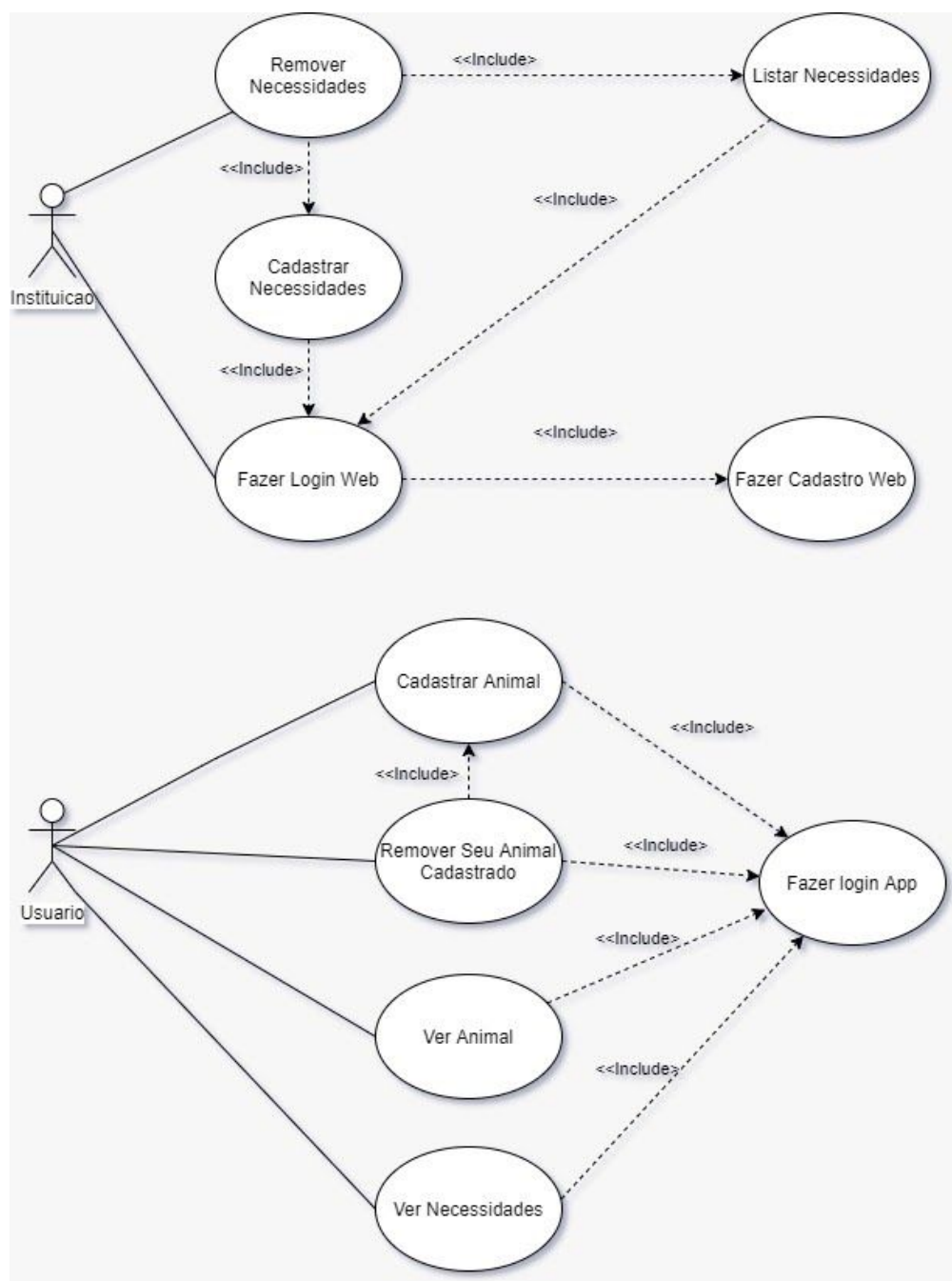
2. OBJETIVO

O objetivo deste trabalho é apresentar uma API REST, denominada de API MIAU, que viabiliza o desenvolvimento de aplicações para adoções de animais, além de realizar a integração Web e *Mobile*, bem como gerenciar as informações geradas no banco de dados. A API MIAU segue os princípios do estilo arquitetural REST e faz uso de *Web Services*, o que permite que aplicações implementadas em diferentes plataformas possam acessá-la.

3. METODOLOGIA

Foi empregada a metodologia de desenvolvimento ágil Scrum (SOMMERVILLE, 2011), que é eficiente na gestão e planejamento de projetos de software. Inicialmente foi feito o *backlog* do sistema, que se trata de uma lista de funcionalidades desejadas na aplicação, no qual a proposta foi dividida em ciclos, conhecidos como *Sprints*, nas quais foram trabalhadas as atividades a serem executadas ao longo do projeto. As principais funcionalidades da API MIAU foram baseadas em outras APIs REST e de acordo com os requisitos da aplicação, descritas no diagrama de caso de uso da **Figura 1**.

Figura 1: Diagrama de caso de uso.



Fonte: Autoria própria (2020).

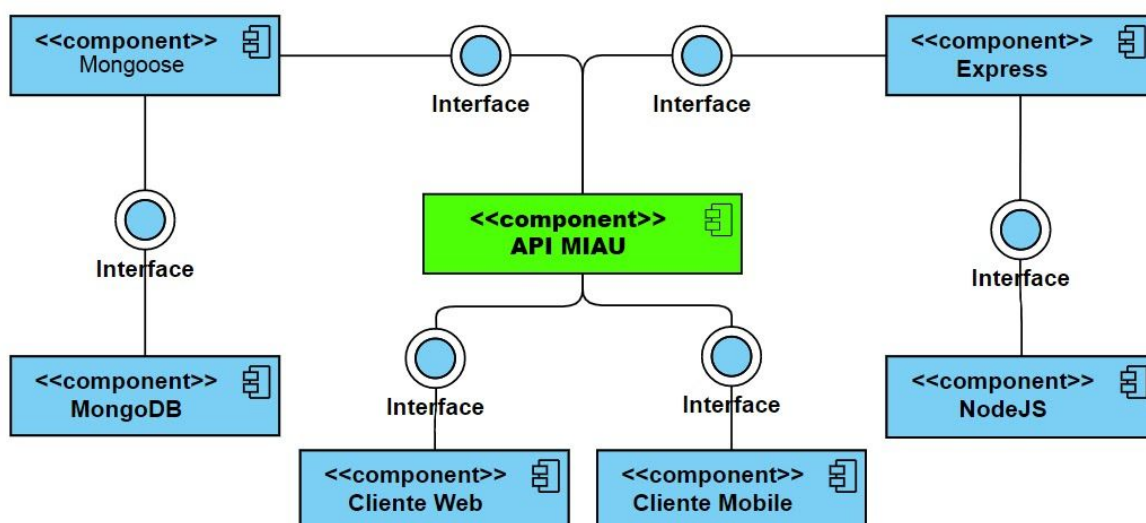
4. DESCRIÇÃO DO SISTEMA

A API MIAU foi desenvolvida com o emprego do estilo arquitetural REST propiciando integrar diferentes sistemas e garantindo o intercâmbio das informações com diferentes linguagens de programação e plataformas de implantação. Assim, é possível

realizar a conexão com as aplicações Web e *mobile*, por exemplo, através das especificações e padrões definidos pelas APIs.

Para o seu desenvolvimento foi utilizada a linguagem Javascript (JAVASCRIPT, 2020). A arquitetura da API pode ser melhor visualizada no diagrama de componentes na **Figura 2**. Os clientes Web e *Mobile* são os requisitantes dos serviços. Na API os serviços são providos junto ao framework Express (STRONGLOOP et. al, 2020) e a plataforma Node.js (OPENJS, 2020), utilizados para o desenvolvimento das funcionalidades. E, por último, encontra-se a parte do armazenamento das informações através do banco de dados MongoDB (MONGODB, 2020), utilizando a biblioteca *Mongoose* do Node.js desenvolvida para conectar-se ao MongoDB.

Figura 2: Diagrama de componentes.



Fonte: Autoria própria (2020).

O funcionamento da API MIAU ocorre com uma requisição sendo enviada da aplicação para o servidor com uma mensagem específica qual o método será utilizado para o envio. Os métodos HTTP (GET, POST e DELETE) indicam qual ação deve ser realizada no servidor. O método GET, recupera dados dos recursos. Não é utilizado para executar ações, mas apenas para retornar dados. O método POST, utilizado para criar um novo recurso através de uma representação como resposta. O servidor envia ao cliente informação a respeito do sucesso ou falha da operação. O método DELETE, tem como finalidade a remoção de um recurso.

A API é destinada ao uso de aplicações Web e *mobile* para realizar adoções de animais. Além disso, a API MIAU permite realizar o cadastro de instituições responsáveis em cuidar de animais. Assim, através de requisições é possível que os doadores/adotantes sejam cadastrados, onde através de um perfil poderão realizar adoções de pets, doações para instituições que cuidam dos animais ou a divulgação a fim de encontrar um dono responsável.

As rotas são os pontos de acesso disponibilizados por uma API para que a aplicação cliente possa consumir os serviços providos, sendo classificadas em recursos como os principais tópicos disponibilizados (SOARES, 2019). Foram desenvolvidas 12 rotas para acessar os dados da API, divididos em 4 categorias, sendo elas: usuário, instituição, animal e necessidade, na qual podem ser visualizados abaixo:

Usuário

- [POST] - '/register' => Cria um usuário;
- [POST] - '/authenticate' => Faz login do usuário.

Instituição

- [POST] - '/registerInst' => Cria uma Instituição;
- [POST] - '/authenticateInst' => Faz login da Instituição;
- [GET] - '/listInst' => Retorna todas as Instituições cadastradas.

Animal

- [POST] - '/animal' => Cadastrar animal;
- [GET] - '/animals' => Retorna todos os animais cadastrados;
- [GET] - '/myAnimals' => Retorna todos os animais cadastrado pelo usuário logado (id no header);
- [DELETE] - '/animals/:id' => Excluir um animal de acordo com o id.

Necessidades

- [POST] - '/necessitie' => Cadastrar necessidade em uma instituição(id da instituição no header);
- [GET] - '/necessities' => Retorna todas as necessidades cadastradas;
- [DELETE] - '/necessities/:id' => Excluir necessidade de acordo com o id.

A **Figura 3** ilustra o processo para criar um novo usuário. É enviado ao servidor uma requisição utilizando o método POST, composta pelos dados necessários para registro de um usuário, que são eles: nome, e-mail, senha, telefone, cidade, estado e uma foto.

Figura 3: Rota para registrar usuário

```
26
27   const handleRegister = () => {
28     try {
29       const data = {
30         name, email, password, phone, city, state, img_user
31       }
32       api.post('/register', data).then(resp => {
33         console.log('OK');
34         console.log(resp.data);
35       }).catch(err => {
36         console.log('error');
37         console.log(err);
38       })
39       navigation.navigate('Login');
40     } catch (error) {
41       console.log(error);
42     }
43   }
44
```

Fonte: Autoria própria (2020).

A **Figura 4**, ilustra o processo para listar os animais cadastrados pelo próprio usuário logado na aplicação. É enviado ao servidor uma requisição utilizando o método GET, e no cabeçalho é passado o ID do usuário logado.

Figura 4: Rota para carregar animais

```
14
15   useEffect(() => {
16     async function loadMyAnimals() {
17       const user_id = JSON.parse(await AsyncStorage.getItem('user')).user_id;
18       const { data } = await api.get('/myAnimals', { headers: { user_id } });
19       const { animals } = data;
20       setMyAnimals(animals);
21     };
22     loadMyAnimals();
23   });
24
```

Fonte: Autoria própria (2020).

A **Figura 5**, ilustra o processo para apagar um animal cadastrado pelo próprio usuário logado na aplicação. É enviado ao servidor uma requisição utilizando o método GET, no cabeçalho é passado o ID do usuário logado, e na url é passado o id do animal que deseja apagar. Essa requisição retorna a uma lista com os animais cadastrados pelo usuário atualizada, ou seja, sem o animal que ele escolheu apagar.

Figura 5: Rota para deletar animal

```
34     };  
35  
36     const deleteAnimal = async (id) => {  
37         const user_id = JSON.parse(await AsyncStorage.getItem('user')).user_id;  
38         const { data } = await api.delete(`/animals/${id}`, {  
39             headers: { user_id },  
40         });  
41         const { animals } = data;  
42         setMyAnimals(animals);  
43     };  
44
```

Fonte: Autoria própria (2020).

5. RESULTADOS

Foram realizados os testes com os métodos da API através da ferramenta Insomnia (KONG, 2020), e conseqüentemente, a validação dos mesmos. O principal objetivo foi testar o comportamento dos métodos da API, por meio da validação com a inserção de valores.

Diante do exposto, como plataforma distribuída, a API MIAU para este caso de testes proporcionou e disponibilizou os recursos necessários para a criação de aplicações para adoções de animais. Um exemplo de sua utilização como fornecedora de serviços pode ser vista na aplicação MIAU: Aplicação Híbrida Para Adoções de Animais, que se baseia no consumo dos recursos disponibilizados pelas rotas da API proposta.

Como perspectivas futuras, tem-se: site de divulgação da API, para que assim outros desenvolvedores possam usá-la em suas aplicações. Além disso, planeja-se o desenvolvimento de novas rotas para atualizar o perfil do usuário e filtro de pesquisa para os animais a serem adotados de acordo com suas características, por exemplo, porte(grande ou pequeno), espécie(gato ou cachorro), etc.

6. REFERÊNCIAS

ANANDA. **Brasil tem 30 milhões de animais abandonados.** 2014. Disponível em: <https://goo.gl/sjojsT>. Acesso em: Fevereiro de 2020.

JAVASCRIPT. **About.** 2020. Disponível em: <https://www.javascript.com/about>. Acesso em: Fevereiro de 2020.

MONGODB, Inc. **What Is MongoDB?**. 2020. Disponível em: <https://www.mongodb.com/what-is-mongodb>. Acesso em: Fevereiro de 2020.

OPENJS, Foundation. **About Node.js**. 2020. Disponível em: <https://nodejs.org/en/about/>. Acesso em: Fevereiro de 2020.

SOARES, Elisa de Fátima Andrade. **API URB: UMA API REST PARA GERENCIAMENTO DE PROBLEMAS URBANOS**. 2019. 88 f. TCC (Graduação) - Curso de Curso de Ciência da Computação, Universidade do Estado do Rio Grande do Norte - Uern, Mossoró - Rn, 2019.

SOMMERVILLE, I. **Engenharia de Software**. Ed. 9. São Paulo, Brasil: Pearson, 2011.

STRONGLOOP, et. al. **Fast, unopinionated, minimalist web framework for Node.js**. 2020. Disponível em: <https://expressjs.com>. Acesso em: Março de 2020.

KONG, Inc.. **Debug APIs like a human, not a robot**. 2020. Disponível em: <https://insomnia.rest>. Acesso em: Março de 2020.