



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

SAMMARA BESERRA NUNES

PETCARE:

UMA PLATAFORMA PARA O ACOMPANHAMENTO MÉDICO DE ANIMAIS DOMÉSTICOS

CAMPINA GRANDE - PB

2023

SAMMARA BESERRA NUNES

PETCARE:

UMA PLATAFORMA PARA O ACOMPANHAMENTO MÉDICO DE ANIMAIS DOMÉSTICOS

**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 Bacharel em
Ciência da Computação.**

Orientador : João Arthur Brunet Monteiro

CAMPINA GRANDE - PB

2023

SAMMARA BESERRA NUNES

PETCARE:

UMA PLATAFORMA PARA O ACOMPANHAMENTO MÉDICO DE ANIMAIS DOMÉSTICOS

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 Bacharel em Ciência da Computação.

BANCA EXAMINADORA:

Professor Dr. João Arthur Brunet Monteiro

Orientador – UASC/CEEI/UFCG

Professor Dr. Marcus Salerno de Aquino

Examinador – UASC/CEEI/UFCG

Melina Mongiovi

Professora da Disciplina TCC – UASC/CEEI/UFCG

Trabalho aprovado em: 17 de NOVEMBRO de 2023.

CAMPINA GRANDE - PB

RESUMO

Com o aumento de animais domésticos nos lares e ONGs de proteção aos animais, sobretudo durante a pandemia, também aumentou a busca por médicos veterinários e, conseqüentemente, exames clínicos para o cuidado deles. Porém, tais resultados laboratoriais são mantidos pelas clínicas e/ou profissionais que os realizam, causando enorme problema quando o tutor ou profissionais veterinários precisam do histórico médico do pet, embora devessem ser realizados orientados ao paciente, ou seja, o animal. Neste trabalho, buscamos desenvolver uma aplicação mobile que permite unificar o lugar em que estes dados são armazenados e facilitar a busca e a consulta de dados médicos de um pet. Nesse projeto foi produzido como resultado uma aplicação com NodeJS para Backend e React Native para Frontend, além de diagramas detalhados que foram usados para a construção da aplicação. Ao final do desenvolvimento foi realizada uma avaliação com 10 voluntários acerca da utilidade de suas funcionalidades, onde obteve-se resultados positivos da experiência do usuário, destacando-se a facilidade de uso do sistema e utilidade das funcionalidades implementadas. Também foi obtido feedback a respeito de melhorias no sistema que foram sugeridas ao final da avaliação, como refatorar as funcionalidades de registro de exame e cirurgias para facilitar seu uso e novas funcionalidades para o sistema.

PetCare: uma plataforma para o acompanhamento médico de animais domésticos

Sammará Beserra Nunes

Unidade Acadêmica de Sistemas e Computação
Universidade Federal de Campina Grande Campina
Grande, Paraíba, Brasil
sammará.nunes@ccc.ufcg.edu.br

João Arthur Brunet Monteiro

Unidade Acadêmica de Sistemas e Computação
Universidade Federal de Campina Grande Campina
Grande, Paraíba, Brasil
joao.arthur@computacao.ufcg.edu.br

Resumo

Com o aumento de animais domésticos nos lares e ONGs de proteção aos animais, sobretudo durante a pandemia, também aumentou a busca por médicos veterinários e, conseqüentemente, exames clínicos para o cuidado deles. Porém, tais resultados laboratoriais são mantidos pelas clínicas e/ou profissionais que os realizam, causando enorme problema quando o tutor ou profissionais veterinários precisam do histórico médico do pet, embora devesseser realizados orientados ao paciente, ou seja, o animal. Neste trabalho, buscamos desenvolver uma aplicação mobile que permite unificar o lugar em que estes dados são armazenados e facilitar a busca e a consulta de dados médicos de um pet. Nesse projeto foi produzido como resultado uma aplicação com NodeJS para Backend e React Native para Frontend, além de diagramas detalhados que foram usados para a construção da aplicação. Ao final do desenvolvimento foi realizada uma avaliação com 10 voluntários acerca da utilidade de suas funcionalidades, onde obteve-se resultados positivos da experiência do usuário, destacando-se a facilidade de uso do sistema e utilidade das funcionalidades implementadas. Também foi obtido feedback a respeito de melhorias no sistema que foram sugeridas ao final da avaliação, como refatorar as funcionalidades de registro de exame e cirurgias para facilitar seu uso e novas funcionalidades para o sistema.

Palavras-chave

mobile, medicina veterinária, desenvolvimento de software, react native

1. Introdução

O Instituto Pet Brasil estima que, no Brasil, haja cerca de 149,6 milhões de animais de estimação[1], fazendo o país ocupar a terceira posição mundial em quantidade de pets. Além disso, o setor de pets no Brasil faturou em 2021 cerca de R\$50 bilhões[2]. Com a pandemia, houve um aumento significativo de animais de estimação nos lares que ajudaram as pessoas a passar pelo isolamento social e também o crescimento número de ONGs protetoras dos animais que necessitam de cuidados especiais para os animais resgatados. Estima-se que haja cerca de 30 milhões de animais abandonados no Brasil[3]. O lucro desse mercado está concentrado em lojas de cuidado geral como pet shops de pequeno e médio porte, seguido pelas clínicas veterinárias, agrolojas, varejo alimentar, pet shops de grande porte e comércio eletrônico[4].

Em toda a fase da vida de um animal doméstico ele precisará de cuidados especiais, seja para vacinação, vermifugação, exames, acompanhamento médico de rotina ou procedimentos de tratamento de saúde. Embora esses cuidados sejam comuns, são mantidos pela clínica e/ou responsável que cuidou destes procedimentos, causando confusão caso o tutor ou futuro veterinário que precise cuidar destes animais precisem do histórico médico do pet caso ele necessite mudar de cidade ou prefira outro profissional. Seria de extrema importância que estes cuidados fossem orientados ao paciente, ou seja, o animal que está sendo tratado.

Além disso, tal histórico médico deveria ser concentrado em apenas um lugar que pudesse ser acessível para o tutor, impedindo que ele esqueça, perca e/ou tenha danos em documentos e dados sobre a saúde do animal, como datas de procedimentos médicos (exames, cirurgias, etc.), medicamentos e quaisquer condições que o animal possa vir a ter.

Ao final do desenvolvimento deste projeto, se obteve um aplicativo Android onde os usuários poderiam realizar o cadastro e gerenciamento da vida hospitalar de seus animais. Para validação da aplicação foi realizada uma avaliação com 10 usuários, tutores de animais que já possuíam algum histórico médico de seus pets. Com esta pesquisa, obteve-se resultados positivos da experiência em relação a facilidade de uso do aplicativo e da utilidade das funcionalidades implementadas. No entanto, houve um feedback em relação à dificuldade de uso em duas funcionalidades específicas: registro de exames e o registro de cirurgias, onde relataram problemas em quais informações deveriam ser inseridas. Também houve um espaço para sugestão de funcionalidades que acreditam ser úteis para o aplicativo futuramente.

2. Metodologia

Para a construção da aplicação, foi preciso levar em conta as funcionalidades, a quantidade delas e o quanto seriam complexas. Além disso, para a escolha das tecnologias, foi considerado o critério de serem relevantes no mercado e o interesse pessoal em aprender as mesmas, visto que são pré-requisitos em grandes empresas.

Para a decisão de criação das *features*, foram utilizados diagramas para abstrair as informações que o banco de dados registraria e idealizar quais deles seriam importantes de manter na aplicação, tanto para consulta quanto para registro, além da experiência pessoal com clínicas veterinárias para animais domésticos.

Para a execução de atividades, foi utilizada a metodologia SCRUM [14], onde as atividades foram divididas em sprints com a estipulação de metas para cada uma delas. Cada funcionalidade foi estipulada de duas semanas (uma para desenvolvimento de funcionalidades de backend e outra para desenvolver o frontend), podendo ser adaptadas, em caso de adiantamento ou atraso de alguma funcionalidade. Para cada uma delas, era realizada uma validação para checar se estava nos conformes da especificação e disponível para a conexão com outras funções.

Para a elaboração dos diagramas, foi utilizada a ferramenta draw.io, onde dispõe de uma grande gama de templates e componentes para a construção de modelos, que representam a arquitetura da aplicação. Para a representação da arquitetura, foi utilizado o padrão C4 devido a um maior nível de detalhamento do serviço.

Para o versionamento de código, foi utilizado o GitHub por ser amplamente conhecido. Nele, foram criados dois repositórios: um para o front-end e outro para o backend.

3. Solução

Neste trabalho apresenta-se o PetCare, uma aplicação mobile que mostra o histórico médico de um animal de estimação, incluindo exames, remédios, vacinas, cirurgias e dados do animal. Sendo desenvolvida utilizando as tecnologias NodeJs e o framework express para o servidor backend e JavaScript contando com o framework React Native para o desenvolvimento da aplicação mobile em Android. Ambas foram escolhidas devido à vontade de ter um contato com o desenvolvimento de aplicações mobile. Principalmente React Native, devido a ser uma tecnologia que é muito utilizada comercialmente e conta com uma vasta gama de recursos e bibliotecas disponíveis.

O público alvo são pessoas que possuem animais domésticos e têm o interesse em manter os dados médicos do seu animal em um lugar seguro e unificado, facilitando a consulta e auxiliando-os na administração da saúde dos seus animais.

Através da aplicação desenvolvida, o usuário poderá se cadastrar e registrar seus animais, registrar o histórico médico deles, características e registro de remédios, vacinas e vermífugos. É necessária uma autenticação para registrá-los e editar informações acerca deles.

Nas próximas seções serão apresentados e detalhados os requisitos funcionais e não-funcionais do sistema, tanto de como eles se comunicam, assim como também foram garantidos na implementação.

3.1 Requisitos funcionais

Nesta seção será descrito as funcionalidades do sistema desenvolvido durante este projeto. Elas foram descritas utilizando o modelo de user story, sendo escritas da perspectiva de um usuário, o fluxo que ele deveria seguir para alcançar o objetivo daquela funcionalidade e o que é esperado que aconteça no objetivo final.

Chegou-se aos requisitos funcionais do sistema através de um levantamento que considerou as principais funções que donos de animais julgam importante para administrar a saúde do seu pet, assim como também de características que são consideradas comuns para o tratamento de todo animal doméstico, como por exemplo a realização de vacinas anuais.

3.1.1 Realizar login

Como usuário cadastrado, devo conseguir realizar acesso ao sistema através do meu e-mail e senha cadastrados, com o fim de autenticar a aplicação para garantir que meus dados sejam consultados de forma correta e impeça que pessoas não-autorizadas de visualizar minhas informações (Figura 1).

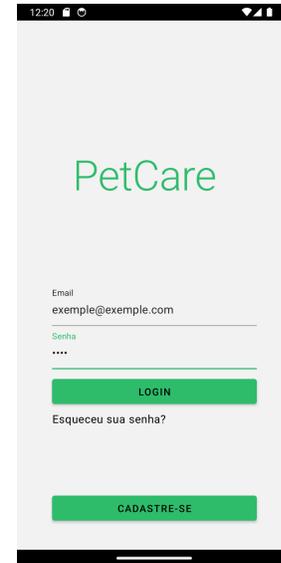


Figura 1: Login de um usuário

3.1.2 Cadastrar um usuário

Como usuário, devo realizar meu cadastro incluindo dados pessoais (nome, sobrenome, e-mail, telefone e a criação de uma senha) e o cadastro de pelo menos um animal inserindo seus dados gerais como nome, peso, data de nascimento, sexo e se é castrado ou não (Figura 2).

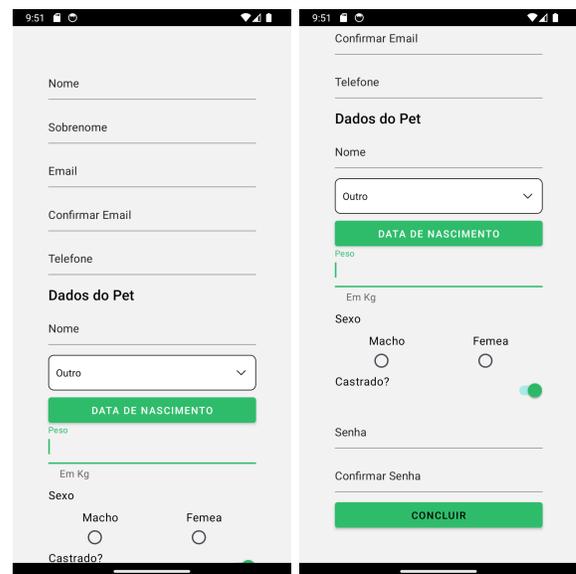


Figura 2: Tela de cadastro

3.1.3 Visualização de animais cadastrados

Após o login bem-sucedido, como usuário cadastrado devo visualizar uma lista com todos os animais que eu cadastrei

informando o nome e o tipo dele (canino, felino ou outro). Abaixo, temos um botão que direciona para a janela de cadastro de um novo pet detalhado na seção 3.1.4. Também posso clicar em itens da lista para ver detalhes sobre meu animal (funcionalidade será melhor explorada na seção 3.1.5) (Figura 3).

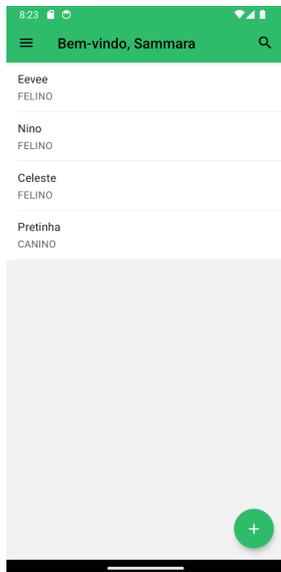


Figura 3: Tela de exibição de Animais

3.1.4 Cadastrar um novo pet

Como usuário, devo poder cadastrar um outro animal, inserindo suas informações básicas como nome, peso, data de nascimento, sexo e se é castrado ou não. Após a inserção das informações, pode salvá-las ao toque do botão “Concluir” (Figura 4).



Figura 4: Tela de cadastro de novo animal

3.1.5 Visualização de dados gerais do animal

Como usuário, devo visualizar informações detalhadas do meu animal como nome, peso, data de nascimento, sexo e se é castrado ou não. A tela também possui um conjunto inferior de cinco abas onde cada uma detalha uma questão do pet, sendo elas: Sobre (a

inicial), Cirurgias anteriores, Condições, Medicamentos e exames. Estas serão melhor exploradas nas seções seguintes. (Figura 5).

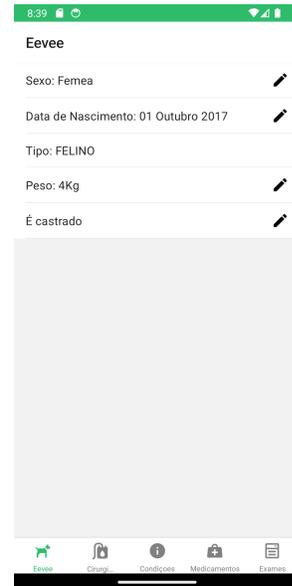


Figura 5: Tela de exibição de dados gerais de um animal

3.1.6 Visualização de Cirurgias anteriores

Como usuário, devo visualizar as cirurgias feitas anteriormente pelo meu animal. Cada item acompanha um ícone para editar ou deletar o item. Também acompanha um botão para inserir uma nova cirurgia, onde irá disponibilizar uma janela com uma linha para a inserção das informações de nome e data do procedimento (Figura 6).

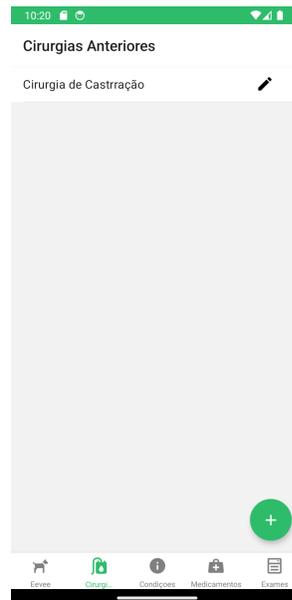


Figura 6: Exibição de Cirurgias Anteriores

3.1.7 Visualização das condições do animal

Como usuário, devo visualizar as condições do meu animal, como deficiência física, visual, auditiva, problemas cardíacos, entre

outros. Cada item acompanha um ícone para editar ou deletar a condição. Também acompanha um botão para inserir uma nova condição, onde irá disponibilizar uma janela com uma linha para a inserção das informações a respeito da condição (Figura 7).

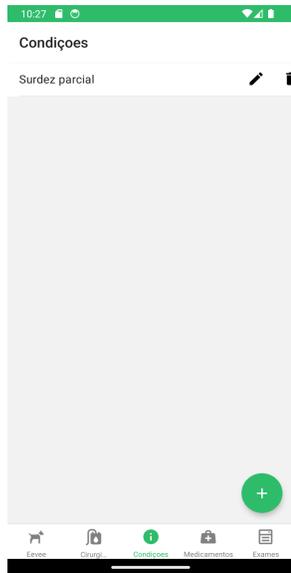


Figura 7: Exibição de condições do animal

3.1.8 Visualização de medicamentos

Como usuário, devo visualizar os fármacos que meu animal tomou ou toma. Os medicamentos podem ser classificados como vacinas, vermífugos ou remédios. Nesta tela, será visualizada a lista de medicamentos, acompanhados de seu tipo e ícones para deletar ou editar o item. Ao clicá-los, poderei visualizar melhor sobre o remédio, trazendo informações como nome, tipo, data de início da primeira dose e da última dose, se é uma medicação recorrente e a próxima dose (caso se aplique, como nas vacinas e vermífugos) (Figura 8 e 9).

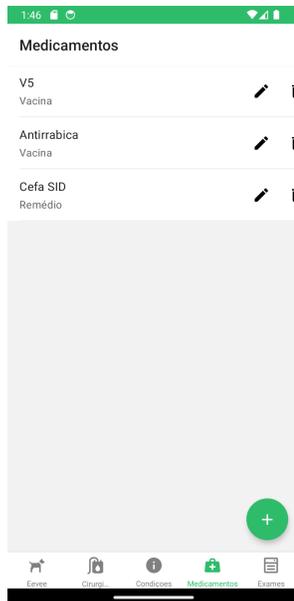


Figura 8: Exibição de medicamentos do animal.

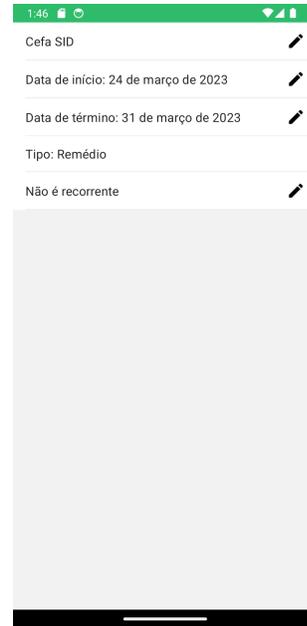


Figura 9: Exibição de detalhes do medicamento.

3.1.9 Exames

Como usuário, deve-se visualizar os exames realizados anteriormente pelo meu animal. Assim como na tela de Cirurgias anteriores, será possível ver cada item com ícones para editar e deletar informações, além de um botão para a inserção de um novo exame com a disponibilização de uma janela com campos para a inserção de nome e data do procedimento (Figura 10).

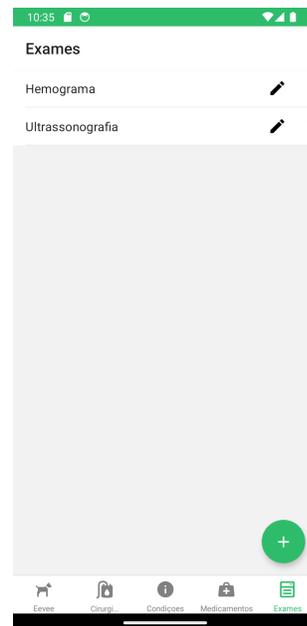


Figura 10: Exibição de exames anteriores

3.2 Requisitos não-funcionais

Para o levantamento dos requisitos não-funcionais, foram elencados quatro pontos importantes considerados vitais para a qualidade de um bom sistema: usabilidade, portabilidade,

disponibilidade e a segurança, visto que estes são requisitos suficientes para um desempenho satisfatório da aplicação no momento. A seguir, são descritos cada um deles e como foram garantidos na implementação do projeto.

A usabilidade foi um dos pontos em questão de importância da plataforma, para garantir que os usuários tenham uma boa experiência ao utilizar a aplicação e que seja de fácil uso. Para garantir isso, foi utilizado o React Native Material, uma biblioteca open-source que fornece diversos componentes pré-prontos, com documentação para cada um deles e de como utilizá-los, que agilizou o desenvolvimento do aplicativo. Ele também garante uma boa experiência para o usuário, uma vez que ele foi feito com base no Google Material[5], que são uma série de diretrizes para o desenvolvimento de plataformas que focam na melhor experiência do usuário.

Através do React Native é possível desenvolver aplicações que podem ser portadas tanto para Android como para iOS com um único código-fonte, garantindo assim uma portabilidade para o sistema desenvolvido do PetCare. Como há a necessidade de um aparelho Apple para gerar o arquivo de instalação nos aparelhos iOS, a versão MVP do sistema foi apenas testada em Android, sendo que futuramente será gerada para ambos os sistemas.

A implantação do servidor se deu através de um container que foi levantado em uma máquina virtual em uma instância EC2 na AWS da Amazon, assim garantindo que ele esteja sempre disponível para requisições. Quanto ao aplicativo mobile, será disponibilizado tanto na Google Play Store quanto na App Store do iOS futuramente. Em um primeiro momento o MVP está sendo testado apenas por usuários selecionados distribuindo manualmente o arquivo .apk para android.

Com a utilização dos tokens JWT é possível realizar requisições com segurança no aplicativo, onde as informações sensíveis, como nome e e-mail do usuário, estão inseridas no payload do token, incapazes de serem decodificadas sem a palavra-chave que se encontra apenas no servidor. O servidor confere se cada requisição possui o Header de autenticação contendo o token. As informações pessoais também são codificadas no banco de dados da aplicação, através de um algoritmo Hash, que é feito utilizando a biblioteca Bcrypt.

Na próxima seção será detalhado a arquitetura do projeto e seus diversos componentes.

4. Arquitetura

Para descrever a arquitetura do PetCare, será usada a documentação C4 [6], onde é constituída pelo contexto, container, componentes e código. Cada elemento citado é responsável por ilustrar um nível de abstração que parte de um nível mais superior, que mostra como o usuário interage com sistema e possíveis outros sistemas, até um nível mais detalhado, seguindo a ideia de quanto menor o nível, mais detalhada é a solução.

4.1 Contexto

O PetCare possui uma API própria onde os usuários podem inserir informações, consultá-las e manipulá-las sobre os dados médicos de seus animais de estimação. No momento, não utiliza comunicação com sistemas externos para seu funcionamento (Figura 11).



Figura 11: Contexto da aplicação

4.2 Containers

A aplicação possui dois containers: A interface mobile e o servidor (Figura 12).

A aplicação móvel serve como interface para interação do usuário que através de ações na aplicação faz requisições para o servidor que permite que ele insira, consulte, atualize e delete informações suas e de seu animal de estimação. Devido a flexibilidade da interface é possível adicionar diversos tipos de funcionalidades futuramente. Toda a comunicação realizada com o servidor é utilizando requisições com o protocolo *http*.

O servidor tem a regra de negócios da aplicação, provendo os dados para o aplicativo móvel através de requisições, requerendo autenticação para o acesso das rotas do servidor, através do módulo de requisição que será detalhado na próxima seção. Também realiza a criptografia de senha para acesso da aplicação através de uma função *hash* realizada pela biblioteca *bcrypt*, tanto no ato de cadastro quanto no ato de *login*. Com isso, o servidor consegue acesso ao banco de dados, onde recupera informações do usuário, garantindo que eles terão acesso apenas aos seus próprios dados. O servidor estará disponível 24 horas através de uma instância EC2 da AWS (Amazon Web Service).

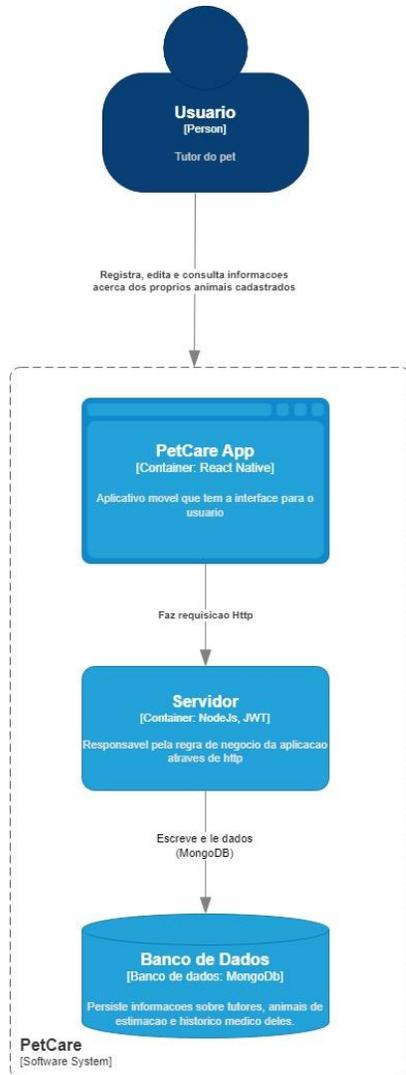


Figura 12: Containers da aplicação

4.3 Componentes

Nesta seção serão detalhados os componentes da aplicação que foram citados anteriormente e seus diversos módulos, descrevendo suas partes e comportamento esperados.

4.3.1 Servidor

Este componente possui quatro módulos que são responsáveis pela autenticação da aplicação, comunicação através das rotas http, regra de negócio e entidades do banco de dados (Figura 13). A seguir ela serão detalhadas:

1. Autenticação: Verifica se o usuário tem permissão para acessar as rotas do sistema e garante a segurança de que apenas o autor da informação tenha acesso a tais dados através do JWT, onde este guarda as informações de forma segura em seu payload. A palavra-chave utilizada para realizar a codificação do token se encontra apenas no servidor, garantindo que apenas esse tenha acesso à informação passada. Além disso, faz a criptografia e descriptografia da senha utilizando uma função de Hash do bcrypt.

2. Comunicação: Realiza a interação entre a interface do usuário através de rotas que são encarregadas de receber as requisições contendo informações necessárias para a camada de negócio da aplicação. Ao final do processamento é feita a comunicação de resposta para o usuário que fez a solicitação. Toda a comunicação utiliza o protocolo http.
3. Serviço: Responsável pela regra de negócio da aplicação. Aqui os dados são manipulados ou inseridos de acordo com as funcionalidades e dados passados pelo usuário. Realiza comunicação e persistência no banco de dados, assim como validação das informações passadas nas requisições. Nele está também a representação das entidades do banco de dados, representadas por schemas, sendo elas de Tutor e Pet.
4. Repositório: Responsável por conter e persistir as informações da aplicação a respeito das entidades de Tutor e Pet, onde essas podem sofrer alterações através de *queries* executadas pelo servidor, sejam elas de criação, atualização, consulta ou exclusão de documentos.

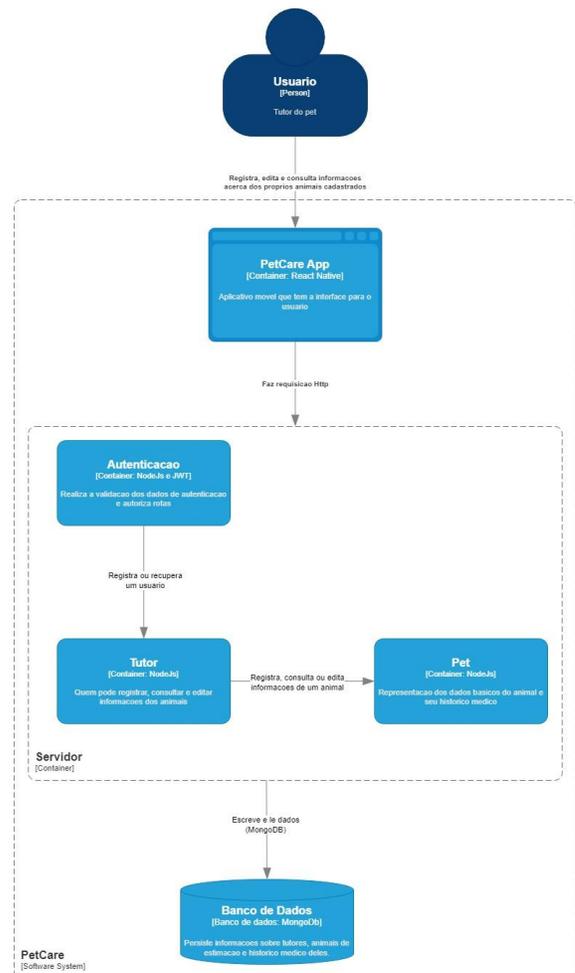


Figura 13: Componentes do servidor-

4.3.2 Cliente

O cliente está estruturado em três partes (Figura 14):

1. Componentes: Estruturas básicas da aplicação que definem o modelo e funções da tela como um template. São também as estruturas que baseiam o framework escolhido, React Native. Cada um deles foi idealizado para representar partes menores da parte visual da aplicação e melhorar o reuso de código.
2. Telas (Screens): Carregam funcionalidades da aplicação como cadastro, visualização, edição e login. Também são responsáveis pelo roteamento entre elas.
3. Compartilhados:
 - a. Store: Gerencia o armazenamento do estado de token, armazenado localmente por um tempo.
 - b. Service: Gerencia a comunicação com o servidor através da biblioteca Axios.

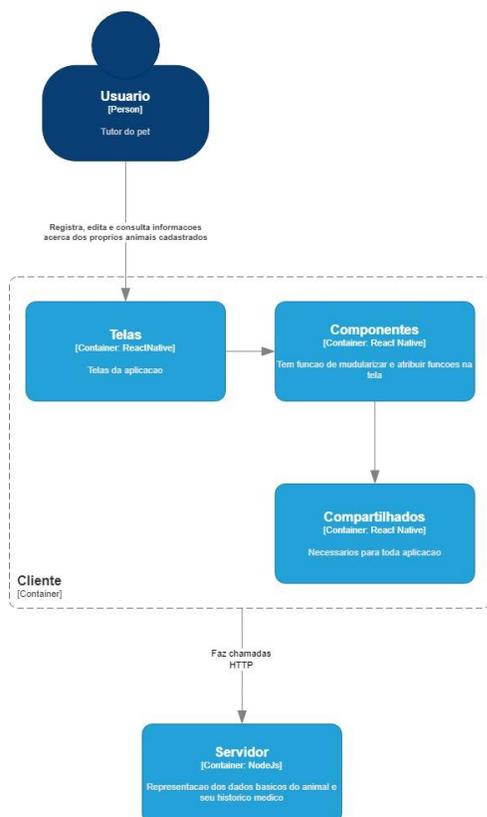


Figura 14: Componentes do cliente

4.4 Implementação

4.4.1 Tecnologias front-end

Para construir o frontend, foi escolhido o framework React Native criado pela Meta Platforms, que permite a criação de telas voltadas para aplicativos móveis através de componentes. Assim como React JS, permite a reutilização de componentes para criar novas interfaces de forma limpa e que permite o reaproveitamento de código através dos componentes. Grandes empresas como Instagram, Facebook, Discord, AirBnb e UberEats[7], são usuários da ferramenta.

Para acelerar o processo de design das telas, foi escolhida a ferramenta React Native Material que possui uma vasta quantidade de componentes usados frequentemente, poupando

tempo de desenvolvimento em componentes frequentemente usados, assim como facilitando a personalização de cores e a alteração de temas como Modo Claro e Modo Escuro.

Para a navegação entre as telas, foi utilizada biblioteca open-source React Navigation [8], permitindo a integração de telas em React Native, além de fornecer algumas funções que não foram fornecidas pelas bibliotecas de design escolhidas.

Além disso, foi utilizada a biblioteca Axios [9], uma ferramenta open-source que visa facilitar a comunicação com a API de forma rápida e simplificada, poupando código e tempo de desenvolvimento.

4.4.2 Tecnologias back-end

Como citado anteriormente, as tecnologias escolhidas para o Backend foram o framework NodeJS desenvolvido pela OpenJS Foundation [13], utilizando a linguagem JavaScript, que permite uma integração mais simples com o MongoDB através da biblioteca Mongoose.

O MongoDB, banco de dados não relacional, foi escolhido devido ao seu caráter simples e suficiente para o tamanho da aplicação. É amplamente utilizado em várias empresas como a Globo.com, MTV, LinkedIn e a SAP [12]. O MongoDB representa suas entidades através de coleções de documentos, onde estes apresentam uma estrutura semelhante a de um objeto JSON, possuindo uma chave e um valor.

4.4.3 Estrutura back-end

O projeto em NodeJs foi dividido em duas pastas: schemas e controllers.

Na pasta Schemas encontram-se os modelos utilizados para representar as entidades do banco, sendo elas Tutor e Pet. Esta pasta também configura os modelos em sua forma de como serão armazenados no banco, definindo atributos pertencentes a cada coleção.

Já na pasta Controllers, lida com toda a gerência de rotas que serão utilizadas para a comunicação com o cliente através do protocolo HTTP, assim como a regra de negócio da aplicação e comunicação com o banco de dados.

Ainda na estrutura de backend, fora destas pastas, encontra-se o arquivo de autenticação, que serve como um filtro, autorizando rotas e validando informações sobre usuários.

4.4.4 Módulo de autenticação

Na autenticação, utilizou-se o JWT (JSON Web Token) onde através de dados como login e senha, são enviados ao servidor e então é gerado um token, com base em uma palavra-chave e é retornado para o usuário um token com as informações, caso em sucesso da identificação. Com ele, o usuário consegue acesso às rotas, fornecendo o token no *header* da requisição. A biblioteca jsonwebtoken do NodeJs foi a responsável por lidar com esta parte.

Visando proteger a senha no banco de dados, utilizou-se também a criptografia através da biblioteca bcrypt js onde se utiliza uma chave para mapear e salvar senhas de forma criptografada. Apenas o servidor tem acesso a ela.

5. Avaliação

Após finalizado, o aplicativo foi avaliado por 10 usuários que são tutores de animais e que de preferência, possuíam um histórico

médico registrado ou alguma condição para que fosse possível o teste de todos os fluxos do sistema. O aplicativo foi distribuído através de um arquivo .apk para os voluntários. Eles foram instruídos a realizar os fluxos de cadastro, login, registro e consulta de animais, registro e consulta de cirurgias, medicamentos e exames.

Foi elaborado um formulário digital sobre as funcionalidades que os voluntários tiveram ao usar o aplicativo e nela foram elaboradas 13 afirmações a respeito da utilidade das funcionalidades, desempenho da aplicação e seu design onde foi utilizada a escala Likert de 1 a 5 para as afirmações sendo 1 para “discordo completamente” e 5 para “concordo completamente”. Além disso, na mesma seção, foi disponibilizada uma pergunta que buscava avaliar a experiência geral que os usuários tiveram ao utilizar o aplicativo em uma escala de 1 a 10, sendo 1 uma experiência desagradável e 10 para uma experiência agradável e um espaço para que estes escrevessem sugestões de funcionalidades que auxiliassem no cuidado de seus animais ou alterações que venham a melhorar a experiência do sistema como um todo.

Ao se analisar os gráficos que foram relacionados ao design e experiência da aplicação para o usuário, como mostra nas figuras 15 e 16, podemos ver que foi atingido o objetivo de proporcionar uma boa experiência do usuário no aplicativo. Em questão de performance também se obteve bons resultados, de acordo com a figura 17, onde os usuários não apresentaram queixas com travamentos ou demora no aplicativo. No geral, o aplicativo teve bons resultados na administração dos dados do animal e na experiência do usuário, como mostrado nas figuras 18 a 21.

No que tange às funcionalidades da aplicação, foram feitas afirmações em relação a sua utilidade e relevância para o dia a dia dos usuários no cuidado de seus pets. Ao olharmos os gráficos resultantes nas figuras 22 até a 28, podemos concluir que a maioria das funcionalidades tiveram boas avaliações pelos usuários, porém há pontos a serem melhorados na visualização de exames e cirurgia do pet.

Para essas funcionalidades foi coletado um feedback a respeito das dificuldades encontradas e os usuários indicaram complexidade ao preencherem informações a respeito de exames e cirurgias, devido a quais informações deveriam ser inseridas para serem salvas. Uma melhoria indicada seria a de pré-definições que indicariam quais campos devem ser preenchidos para cada procedimento.

Ainda no campo onde indicavam estas melhorias, foram apontadas sugestões de melhorias que poderiam ser feitas no aplicativo para o usuário em questão. Com essas sugestões, em conjunto com o feedback dado das funcionalidades, é possível destacar quais pontos de evolução da aplicação, como por exemplo refatorar o método de registro de exame e cirurgia para incluir predefinições com o auxílio de um profissional veterinário que ajudaria na validação das informações. Quanto às novas funcionalidades sugeridas, estas serão descritas na seção de trabalhos futuros.



Figura 15: A usabilidade foi satisfatória para mim

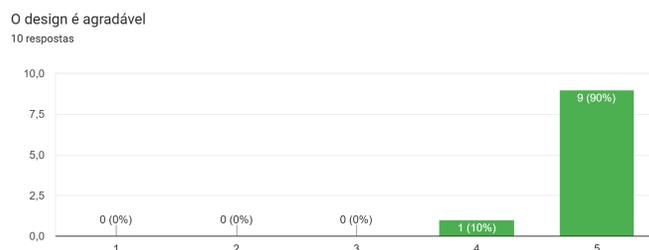


Figura 16: O design é agradável



Figura 17: A performance foi satisfatória para mim

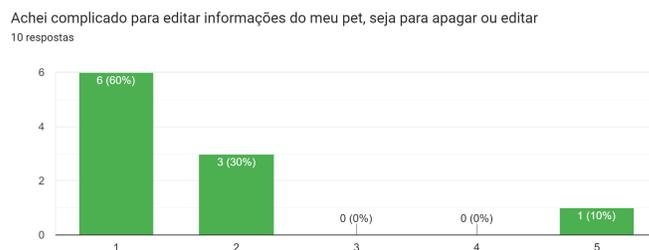


Figura 18: Achei complicado para editar informações do meu pet, seja para apagar ou editar

Ajudará na organização de dados médicos dos meus animais domésticos e trará mais facilidade em seu acesso
10 respostas

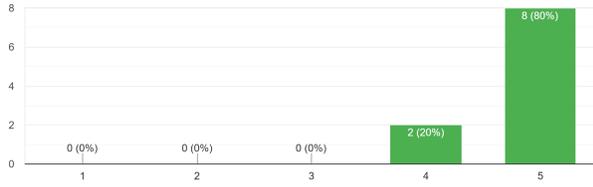


Figura 19: Ajudará na organização de dados médicos dos meus animais domésticos e trará mais facilidade em seu acesso

A visualização de meus animais foi útil
10 respostas

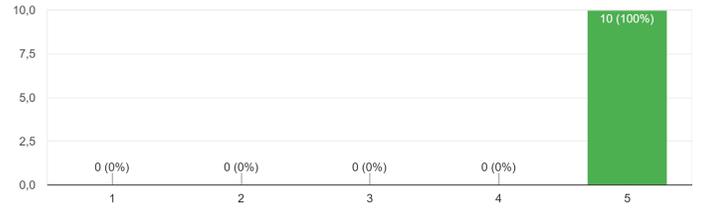


Figura 23: A visualização de meus animais foi útil

Concordo que o PetCare possui funcionalidades essenciais
10 respostas

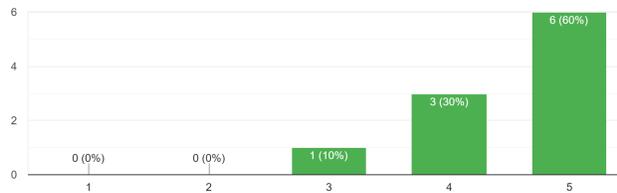


Figura 20: Concordo que o PetCare possui funcionalidades essenciais

A visualização de informações gerais do pet foi útil
10 respostas

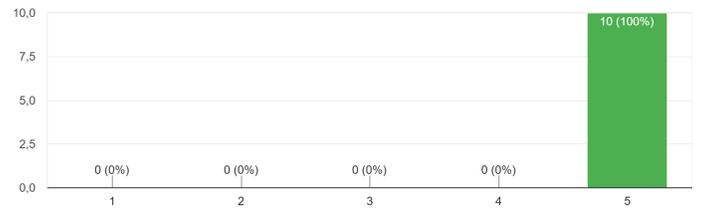


Figura 24: A visualização de informações gerais do pet foi útil

Sobre seu uso e avaliando as funcionalidades, que nota daria para o que você usou? Considere 1 como experiência desagradável e 10 para uma experiência agradável e que recomendaria para alguém.
10 respostas

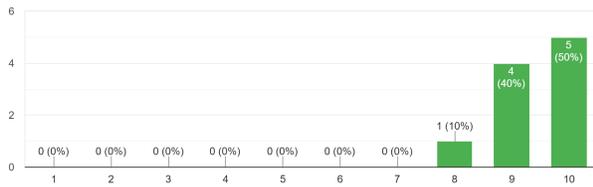


Figura 21: Nota acerca da experiência geral da aplicação

A visualização de Condições do Pet foi útil
10 respostas

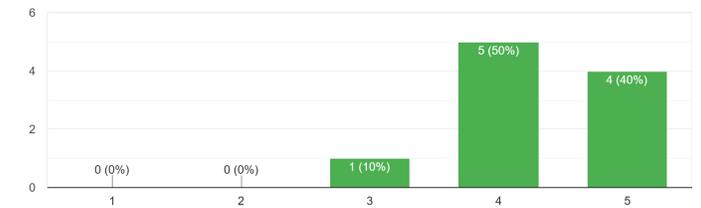


Figura 25: A visualização de condições do pet foi útil

O cadastro de novos pets foi útil
10 respostas

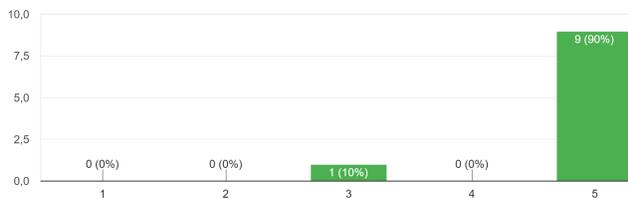


Figura 22: O cadastro de novos pets foi útil

A visualização de Cirurgias Anteriores foi útil
10 respostas

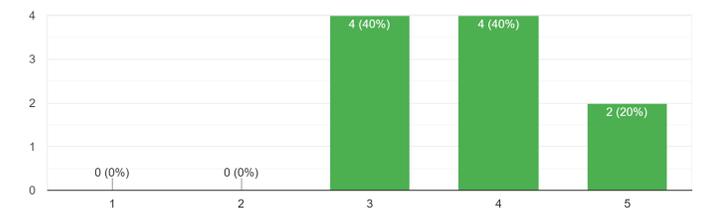


Figura 26: A visualização de cirurgias anteriores foi útil

A visualização de Medicações foi útil
10 respostas

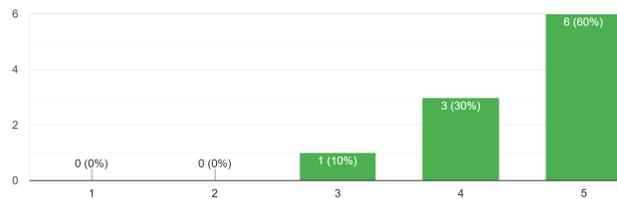


Figura 27: A visualização de medicações foi útil

A visualização de exames foi útil
10 respostas

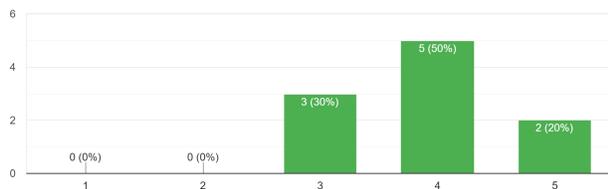


Figura 28: A visualização de exames foi útil

6. Desafios

Desenvolver uma aplicação em um tempo reduzido e com poucas pessoas já é um desafio. Além disso, encontrar ferramentas seguras para a hospedagem da aplicação com uma receita reduzida implicou na escolha de infraestrutura.

Outro ponto importante que se mostrou delicado na questão de aplicações mobile são as burocracias necessárias para a implantação de aplicativo em suas lojas, haja vista que é necessário a transparência e a assinatura de termos para que o futuro usuário possa consentir que em seu aparelho sejam registradas informações ou permitir o uso de funcionalidades de seu celular (como câmera, localização, etc.).

Outra dificuldade relacionada às funcionalidades da aplicação foi a inexperiência com a autenticação Google, pois inicialmente, tinha-se o interesse em que uma opção de autenticação fosse através do Google OAuth, mas não foi possível obter sucesso na implementação.

7. Trabalhos futuros

Futuramente, planeja-se aprimorar efeitos visuais da aplicação, assim como fornecer autenticação com Google (OAuth), viabilizando uma autenticação mais segura e mais rápida, além de aproveitar a conta Google que pode estar atrelada ao dispositivo do usuário, sobretudo aqueles com sistema operacional Android.

Quanto às funcionalidades, pretende-se implementar campos editáveis para datas de procedimentos como exames e cirurgias e uma agenda que notifique ao usuário futuros procedimentos médicos do animal. Também pretende-se tornar a aplicação mais funcional para tutores que possuem animais de outros tipos, já que no momento encontra-se mais focada em cães e gatos.

Ainda quanto às funcionalidades, seria interessante inserir opções de procedimentos médicos para registro pré-definidos, como

exames e cirurgias. Esta modificação seria de extrema importância o auxílio de um profissional veterinário para a melhor validação das informações médicas registradas como opção no aplicativo.

Quanto à portabilidade da aplicação, pretende-se desenvolver a aplicação para uma plataforma Web e outra para iOS, visando uma maior gama de possíveis tutores interessados na aplicação.

Quanto a publicação em lojas de aplicativo móvel, é preciso melhorar a transparência dos dados que serão armazenados para a elaboração de termos que informam ao usuário o que será requisitado e o porquê, além de solicitar a permissão, caso o aplicativo necessite utilizar algumas funcionalidades do *smartphone*.

8. Referências

- [1] Censo Pet IPB: com alta recorde de 6% em um ano, gatos lideram crescimento de animais de estimação no Brasil. <https://institutopetbrasil.com/fique-por-dentro/amor-pelos-animais-impulsiona-os-negocios-2-2/>
- [2] Brasil é o terceiro país com mais pets; setor fatura R\$ 52 bilhões. <https://forbes.com.br/forbes-money/2022/10/brasil-e-o-terceiro-pais-com-mais-pets-setor-fatura-r-52-bilhoes/>
- [3] Abando de animais é crime. <https://noticias.unb.br/artigos-main/6573-abandono-de-animal-is-e-crime>
- [4] Pequenos negócios dominam mercado pet, que está em expansão e fatura bilhões. <https://www1.folha.uol.com.br/mpme/2023/07/pequenos-negocios-dominam-mercado-pet-que-esta-em-expansao-e-fatura-bilhoes.shtml>
- [5] Introduction to Material Design. <https://espeo.eu/blog/the-evolution-of-material-design/>
- [6] The C4 model for visualising software architecture. <https://c4model.com/>
- [7] React Native no desenvolvimento de apps. <https://uds.com.br/blog/react-native-no-desenvolvimento-de-apps/>
- [8] Getting started. <https://reactnavigation.org/docs/getting-started>
- [9] Getting Started. <https://axios-http.com/docs/intro>
- [10] Building a RESTful CRUD API with Node.js, JWT, Bcrypt, Express and MongoDB. <https://medium.com/zero-equals-false/building-a-restful-crud-api-with-node-js-jwt-bcrypt-express-and-mongodb-4e1fb20b7f3d>
- [11] Brasil é um dos países com a maior taxa de celulares Android frente ao iOS. <https://istoedinheiro.com.br/brasil-e-um-dos-paises-com-a-maior-taxa-de-celulares-android-frente-ao-ios/>
- [12] O que é o banco de dados MongoDB? <https://www.mestredaweb.com.br/tecnologias/o-que-e-o-banco-de-dados-mongodb>
- [13] Hosted Projects. <https://openjfs.org/projects/>
- [14] What is Scrum? <https://www.scrum.org/learning-series/what-is-scrum>