

UNIVERSIDADE VIRTUAL DO ESTADO DE SÃO PAULO

Alexandre Junio dos Santos Vieira, RA 2010464

Fábio Borges Dias, RA 2007056

Fabio Rogerio Lins Pereira, RA 2003660

Gustavo Guerreiro Martinho da Cunha Sales, RA 2001642

Marcio Mesquita Marcelli, RA 2002829

Rodrigo Cristiano Ferreira Vieira, RA 2015474

Vandelei Cláudio, RA 1827341

Aplicativo Web para Agendamento de Consultas Veterinárias

Vídeo de apresentação do Projeto Integrador

<<https://youtu.be/i4fXOGrFEuI>>

2022

UNIVERSIDADE VIRTUAL DO ESTADO DE SÃO PAULO

Aplicativo Web para Agendamento de Consultas Veterinárias

Relatório Técnico-Científico apresentado na disciplina de Projeto Integrador II para os cursos de Ciência de Dados, Engenharia da Computação e Tecnologia da Informação, da Universidade Virtual do Estado de São Paulo (UNIVESP).

VIEIRA, Alexandre; DIAS, Fábio; SOUZA, Fabio; SALES, Gustavo; MARCELLI, Marcio; VIEIRA, Rodrigo; CLAUDIO, Vanderlei. **AgendaVet**. Relatório Técnico-Científico. Cursos de Ciência de Dados, Engenharia da Computação e Tecnologia da Informação – **Universidade Virtual do Estado de São Paulo**. Tutor: BISETTI, Denise. Polo Jaguaré e Ribeira, 2022.

RESUMO

Nesta era da informação, lidamos com dados a cada segundo de nossas vidas, e o cotidiano torna-se cada vez mais dinâmico, fazendo com que as soluções em TI estejam cada vez mais presentes atuando a fim de auxiliar no planejamento de nossas rotinas, gastos, desempenho profissional e economizar o recurso mais precioso que temos atualmente, “nosso tempo”.

O presente trabalho tem como objetivo o desenvolvimento de um software para agendamento de serviços tais como: consultas veterinárias, serviços de *pet shops*, e outros no âmbito do “*Pet Care*”. O AgendaVet é uma aplicação web que tem a pretensão de ser um agente facilitador na vida dos usuários. Lá, o usuário se cadastra, encontra o serviço procurado em seu banco de dados, verifica a agenda disponível e agenda o serviço conforme a necessidade, sem telefonemas, sem WhatsApp, sem perda de tempo, mas sem perder acesso a estes canais. O serviço utilizou um *framework* simples o FastAPI para criação da API e o Heroku para o *deploy* de maneira rápida e integrada com o sistema de versionamento de código do Github. Para criação do banco de dados foi utilizado o *framework* do PostgreSQL que é rápido e muito eficaz para a nossa solução. As linguagens utilizadas foram HTML, CSS e TypeScript com React, que é muitíssimo versátil e capaz de construir soluções rápidas e eficientes.

PALAVRAS-CHAVE: Software; Web; Agendamento; Agenda; Banco de Dados; Serviços, Veterinária, Pet Shop.

LISTA DE ILUSTRAÇÕES (opcional)

FIGURA 1 – MODELO ENTIDADE-RELACIONAMENTO INICIAL.....	13
FIGURA 2 – DIAGRAMA DE CLASSES.....	14
FIGURA 3 – FAST API.....	14
FIGURA 4 – INTERFACE DA APLICAÇÃO.....	15

Em Branco.

SUMÁRIO

1. INTRODUÇÃO.....	7
2. DESENVOLVIMENTO.....	8
2.1 OBJETIVOS.....	8
2.1.2 OBJETIVOS ESPECÍFICOS.....	8
2.2. JUSTIFICATIVA E DELIMITAÇÃO DO PROBLEMA.....	9
2. 3. FUNDAMENTAÇÃO TEÓRICA.....	9
2.4. APLICAÇÃO DAS DISCIPLINAS ESTUDADAS NO PROJETO INTEGRADOR.....	10
2.5. METODOLOGIA.....	11
3. RESULTADOS.....	12
3.1. SOLUÇÃO INICIAL.....	12
3.2. SOLUÇÃO FINAL.....	14
4. CONSIDERAÇÕES FINAIS.....	15
REFERÊNCIAS.....	16
ANEXO 1.....	17
ANEXO 2.....	20
ANEXO 3.....	22

1. INTRODUÇÃO

Segundo a Abinpet (Associação Brasileira da Indústria de Produtos para Animais de Estimação) o Brasil tem a segunda maior população de cães, gatos e aves cantoras e ornamentais do mundo sendo o terceiro maior país em população total de animais de estimação. Com um total de indivíduos que ultrapassam os 139 milhões de animais e expõe o potencial e tamanho do mercado consumidor envolvido.

Os setores de alimentação (Pet Food), medicamentos veterinários (Pet Vet) e cuidados com saúde e higiene (PetCare), representam um faturamento anual na casa dos R\$ 50 bilhões e somam 0,36% do PIB brasileiro, sendo um mercado que obteve uma taxa de crescimento de mais de 22% em 2021, em plena pandemia, sendo a classe média a grande consumidora desse tipo de produtos ou serviços com um gasto mensal médio de R\$ 342,20 segundo o jornal G1.

Ainda assim, os gastos com animais de estimação aumentam com a senioridade do animal, impactando o orçamento familiar com a compra de medicamentos e gastos com consultas veterinárias e exames, somando gastos mensais muito acima da média verificada, tornando por muitas vezes proibitivo financeiramente o acompanhamento e tratamento da saúde do pet.

Para ajudar a amenizar este problema e prover serviços de saúde necessários e de qualidade, existem hospitais veterinários públicos, geralmente ligados a instituições universitárias ou a prefeituras.

Entretanto, a alta demanda e o limitado número de vagas gera certa competição pelos atendimentos disponíveis e processos de triagem de pacientes devem ser seguidos.

A triagem dos pacientes geralmente ocorre com chegada do paciente ao local de triagem entre as 6 e 10 horas da manhã para que aguardem na fila, depois há uma entrevista em que são apresentados os sintomas dos pacientes assim como exames prévios. A partir daí as entrevistas são avaliadas e o animal pode ou não ser aceito pelo hospital.

Nesse sentido, o processo de triagem pode se tornar um inconveniente enorme tanto para o tutor quanto para o animal paciente e sua saúde, devido a isso acreditamos que automatizar o processo de triagem traria grandes vantagens a todas as partes envolvidas, ajudando no contato e comunicação entre os profissionais de saúde e os tutores que poderiam, através do envio digitalizado da história do paciente e seus exames ter um direcionamento mais eficiente para o início do tratamento do animal.

Portanto, para atender à essa demanda, idealizou-se o AgendaVet cuja a proposta é criar um ambiente web onde tutores e profissionais poderão trocar as informações necessárias para o entendimento do caso clínico e avançar no processo de tratamento através do agendamento de consultas e retornos, cadastro de atendimentos, pedidos de exames e prescrições veterinárias.

Com o objetivo de verificar a existência de trabalhos prévios no ambiente acadêmico, tais como softwares de agendamento, agendas, e similares, que poderiam ajudar a nortear as buscas e ações no que concerne à criação da solução, de maneira que fosse possível compreender possíveis caminhos, evitar falhas já encontradas e aproximar-se cada vez mais da melhor solução possível para o usuário e proprietários, utilizamos o Google Scholar para efetuar a busca.

Um formulário foi desenvolvido para verificar as necessidades dos clientes e proprietários, sendo os dados, parâmetros para construção da aplicação. Os estes foram coletados e analisados conforme é explicado na metodologia, seguindo as premissas do *design thinking*.

A solução, portanto, consiste na criação de uma aplicação web, com REACT, que utiliza TypeScript, HTML e CSS, Framework FastAPI, banco de dados PostgreSQL, com *deploy* executado em ambiente de nuvem com uso do Heroku.

O tema foi encontrado com base na experiência dos membros da equipe e seu contato com o mercado, analisando as diversas possibilidades dentro dos campos de conhecimento que estão ao alcance da equipe e da tecnologia disponível. A ideia foi buscar soluções para a área de serviços utilizando ferramentas aprendidas por meio das disciplinas da UNIVESP, lecionadas nos cursos do Eixo Computação. Assim, analisando-se a demanda e as necessidades de clientes e profissionais, procurou-se obter uma ideia que permitisse maior agilidade, praticidade e confiabilidade no contato entre profissionais e clientes, sendo a proposta resultado do uso de diversos recursos envolvendo tecnologia, computação, entre outros.

2. DESENVOLVIMENTO

2.1 Objetivos

O objetivo geral do projeto é o desenvolvimento de software web, para agendamento de serviços, utilizando as tecnologias aprendidas nas disciplinas de graduação da UNIVESP.

2.1.2 Objetivos Específicos

Os objetivos específicos são:

- Analisar a proposição com base na viabilidade e utilizar os métodos de *design thinking* para construir a arquitetura da solução, utilizando os *brainstormings*, protótipos e *feed-back* usuários;

- Analisar as melhores tecnologias para a construção da solução aplicando as melhores práticas com uso de ambientes virtuais ou contêineres e sistema de versionamento e compartilhamento de código a fim de garantir a participação da equipe e o bom andamento;
- Escolher a plataforma e colocar o software em produção de forma que esteja acessível e funcional com menor custo, garantindo sua disponibilidade.

2.2. Justificativa e delimitação do problema

Melhorar o processo de triagem de casos clínicos em hospitais veterinários públicos, conectando tutores e médicos veterinários, com organização de informações do paciente animal e agendas dos hospitais.

A triagem de atendimentos é uma necessidade do serviço público de saúde animal e, para agregar eficiência e modernidade ao processo, é preciso desenvolver uma aplicação que permita aos tutores usuários terem controle de um sistema de agendamento e compartilhamento de informação adequado.

Atender a necessidade de uma agenda com os tutores, apresentando os serviços, sua disponibilidade, atendendo às preferências por soluções de tecnologia, e fornecendo um meio rápido e efetivo de confirmação, de consulta e retorno.

A relevância do projeto vem abarcada neste contexto, uma vez que a solução se propõe a reservar o tempo necessário para o compromisso, sincronizando agendas de clientes e negócios, sem custos, com o mínimo tempo e uma interface amigável (encontrando o serviço desejado mais rápido possível, dentro do raio de ação), e atendendo as preferências dos clientes mais conectados, fornecendo informações essenciais e podendo servir como meio de divulgação de promoções.

2.3. Fundamentação teórica

Conforme mencionado anteriormente, foi executada uma pesquisa no ambiente do Google Scholar, onde procurou-se por artigos, dissertações, teses dentre outros documentos considerados durante a pesquisa, a fim de angariar conhecimento relativo à solução proposta e recolher dados pertinentes para sua construção.

Tal pesquisa, como podemos identificar, foram executadas levando em consideração os campos: títulos, *abstracts* e palavras-chaves, que chamaremos aqui de metadados. Isso porque a preocupação era buscar por tópicos relacionados, excluindo-se palavras perdidas no meio dos textos, que viessem a confundir e busca, uma vez que termos como software e web, são muito comuns nos dias atuais.

Nestas buscas foram encontrados 9 documentos pertinentes que foram revisados e estão entre artigos, trabalhos de conclusão de curso, relatórios de estágio e sítios de informações correlacionadas. Nos dando subsídios como os que seguem, para a escolha dos recursos de linguagem, além de servirem de guia para a construção da aplicação, nos quesitos estruturais e procedurais.

Como podemos ver em MACEDO (2013), o uso de linguagem de script como JavaScript (JS) é vastamente utilizado em aplicações web, visto que é uma linguagem leve, interpretada e reconhecida pela vasta maioria dos navegadores, conferindo dinamismo à aplicação e usando recursos do cliente. Mas o React, que é uma biblioteca para uso de linguagem de script, utiliza o TypeScript (TS) que é uma versão mais moderna do JS e resolve um problema da fraca tipagem do JS, com a mesma curva de aprendizagem e desenvoltura para aplicações no “front-end”. O React possui uma coleção de funcionalidades que reduz a necessidade de codificação, facilitando a sua implementação.

No caso dos SGBDs, ou sistemas gerenciadores de banco de dados, foi escolhido o PostgreSQL, que é open source, possui uma ferramenta de administração muito intuitiva e prática, além de ser compatível com a maioria das linguagens, *frameworks*, e recursos de computação em nuvem, para bancos de dados relacionais, cabendo perfeitamente na necessidade da aplicação.

Por fim, para produção da(s) APIs, foi utilizado o framework FastAPI, para assim garantir a comunicação do back-end com o front-end, dividindo os recursos chamados no lado do servidor e no lado do cliente da aplicação.

Partindo deste princípio o material quintessencial para o processo de desenvolvimento será aqueles apresentados durante nas disciplinas e como subsídio ao projeto integrador que serão as questões de desenvolvimento web e banco de dados.

2.4. Aplicação das disciplinas estudadas no Projeto Integrador

Um dos objetivos fundamentais deste trabalho, como método avaliativo da disciplina Projeto Integrador I, é conduzir os integrantes ao uso de ferramentas aprendidas durante os cursos de graduação da Universidade Virtual do Estado de São Paulo UNIVESP, o que se mostrou verdadeiro conforme descrito a seguir:

- Durante o curso da disciplina Formação Profissional em Computação - COM200, foram mostrados conceitos de projeto de um software, como desenvolver e como implementar os algoritmos, foram estudados conceitos de padrões utilizados em desenvolvimento web como HTML e Python além da introdução do conceito de REST, todos recursos empregados na produção deste projeto;
- Durante o curso da disciplina Gestão da Inovação e Desenvolvimento de Produtos - ADM200, foram mostrados os conceitos de desenvolvimento de novos produtos, com o emprego de técnicas de ideação como design thinking, o brainstorming, o plano de negócios e a prototipagem, todos conceitos muito utilizados e fundamentais para condução do projeto;

- Durante o curso da disciplina Banco de Dados – COM300, foram apresentados os conceitos de modelo entidade relacionamento, bancos de dados relacionais e uso de sistemas SGBD que foram cruciais para a produção do resultado final;
- Durante o curso da disciplina Desenvolvimento web - COM320, foram ensinados os conceitos de métodos HTTP, Tecnologias web para front-end e back-end, bancos de dados em aplicações web, e metodologias de desenvolvimento de aplicações web, que foram o cerne para a construção da aplicação.

2.5. Metodologia

Na etapa inicial do projeto foram adotados métodos de ideação como *design thinking*, *brainstormings* em ocasiões com todos os componentes da equipe, partindo inicialmente da enumeração do arcabouço técnico que seria quais os elementos que deveriam ser empregados dentro do contexto que viria a ser escolhido, tendo em vista as questões de estudo quintessenciais como as seguintes:

- O projeto deve ter pertinência e ser uma hipótese para solução de necessidades ou problemas de alguma parcela da sociedade, levando em conta a exequibilidade e a possibilidade de aplicação dos conceitos estudados;
- Deve ser uma proposta de uma solução web, que serão acessadas pelo navegador, mas não naturalmente restritas a isso, tendo em vista as boas práticas de desenvolvimento de softwares;
- Deve fazer uso de banco de dados, relacional no caso, e para tanto deve-se escolher a melhor ferramenta e o melhor SGBD, que tenha aplicação fácil e a possibilidade de ser utilizada dentro de ambientes de produção;
- Deve-se produzir um formulário para avaliação por parte dos usuários, com a possibilidade de retroalimentação do projeto e evolução da proposta inicial;

Para a escolha do tema, o grupo reuniu-se por algumas vezes, sempre com novas propostas, até que se encontrasse uma proposta de valor coerente e viável para as necessidades da sociedade, dentro do microambiente escolhido. Nesta fase, após relatos, decidiu-se pelo software de agendamento, que fosse capaz de flexibilizar, facilitar e acelerar o processo de escolha e agendamento de serviços, inicialmente na área de estética e similares, cujas empresas de menor porte necessitavam de soluções para facilitar o contato com o cliente, automatizando o processo de contratação, sem o custo de horas de trabalho dos profissionais para receptação dos clientes, e sem falhas humanas de informação. Isso tudo com a reserva automática de agenda entre cliente e fornecedor.

Foi criado um formulário para avaliação da solução por parte dos clientes, e esse formulário foi encaminhado para avaliação da proposta de valor e expectativas gerais. Será incorporado no relatório, **Anexo 1**.

Seguindo para uma fase de estudo de material técnico como artigos, dissertações, teses e outras publicações dentro da área, a fim de que se fosse possível avaliar as possibilidades de trabalhos já realizados, e possíveis pontes para aceleração do processo de criação do software.

Será criado um MVP para análise da aceitação dos clientes, sendo uma etapa crucial para retroalimentação do projeto e reestruturação da solução.

Os testes de código serão executados pela equipe de desenvolvimento, e eventuais testes de usabilidade e aceitação por parte de clientes contratantes dos serviços, com o apoio de clientes fornecedores que serão a outra ponta do projeto

O processo de desenvolvimento será com React e elementos de web como HTML e CSS, usando *framework* FastAPI e o banco PostgreSQL, já em vista da execução da aplicação em servidor remoto, em ambiente de nuvem.

Após os testes, os dados serão retroalimentados no projeto para que o MVP dê lugar à primeira versão da solução que será apresentada ao final do projeto.

3. RESULTADOS

Serão identificados aqui os passos iniciais e finais da solução.

3.1. Solução inicial

Após os testes, os dados serão retroalimentados no projeto para que o MVP dê lugar à primeira versão da solução que será apresentada ao final do projeto.

Como explicado anteriormente, a proposta da aplicação é de melhorar os processos administrativos em hospitais, que culmina em uma persistência de informações de maneira clara, disponível e de fácil acesso, para os tutores pacientes e hospitais.

Para avaliar a adesão dos possíveis clientes e a aderência da proposta de solução com os problemas e necessidades dentro destes, dentro da proposta, foi elaborado um questionário, onde estavam as perguntas:

1. Quantos pets existem em sua residência?
2. Quais?
3. Aproximadamente, com qual frequência você busca serviços de atendimento médico-veterinário para seu(s) pet(s)?
4. Qual é o principal problema encontrado no atendimento no veterinário?

5. Qual o tempo médio que seu pet leva para ser atendido?

O modelo do formulário encontra-se no Anexo 1 deste relatório, e os resultados estão descritos no **Anexo 2**. Onde os resultados demonstram a aderência e viabilidade do aplicativo.

Os primeiros passos se deram com o levantamento do esquema do banco de dados que tem integração com a linguagem escolhida para criação da API, Python em nosso caso, e a possibilidade de execução do ambiente em nuvem. Foi criado o modelo entidade-relacionamento do BD e ambos, em sua versão inicial, apresentado na figura 1 a seguir.

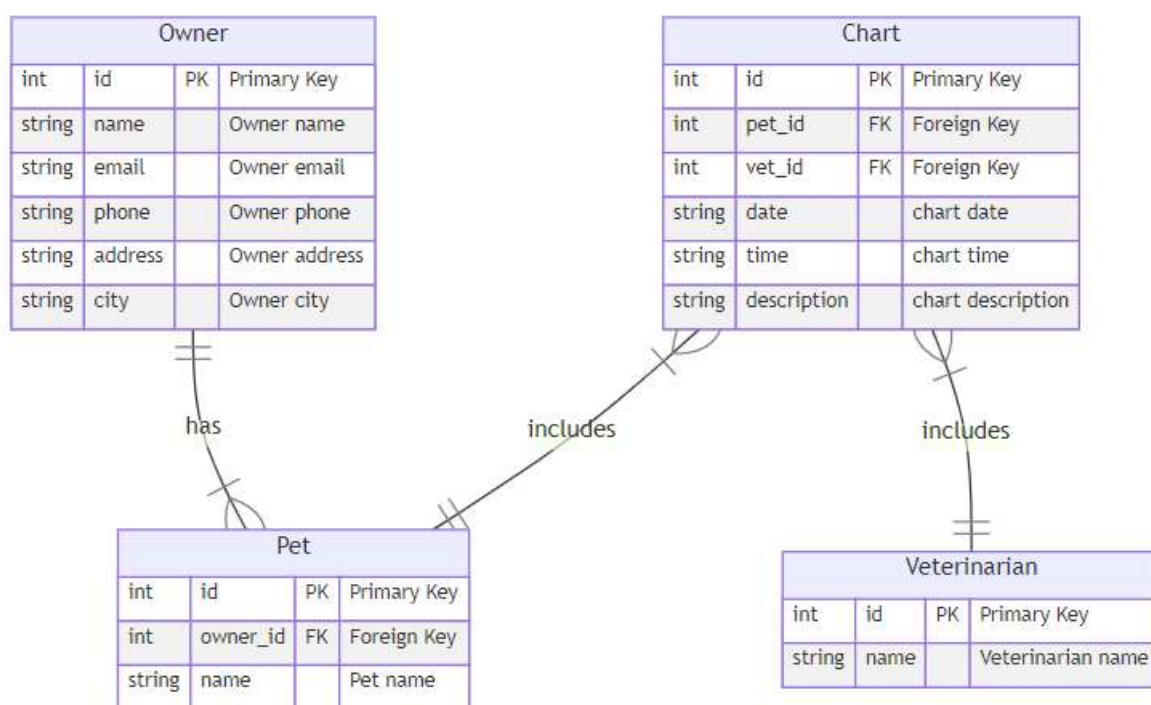


Figura1 – Modelo Entidade-Relacionamento Inicial

Simultaneamente com processo de criação do banco de dados, foi o processo de codificação da API para consumo no front-end. Conforme o código apresentado no **Anexo 2**.

No versionamento do código foi escolhido o Github, que já foi objeto de estudo durante os cursos da UNIVESP e pela sua imensa adesão de toda a comunidade de desenvolvimento, sendo uma opção segura (do ponto de vista de confiabilidade) e fácil de se empregar. <https://github.com/Rcristiano/api-agenda>.

O IDE usado para construção da solução foi o Visual Studio Code, que é uma IDE simples, mas extremamente poderosa, e com as devidas extensões é capaz de se integrar com recursos de containers, bancos de dados, repositórios e mais. Assim foi escolhido um SGBD que foi o PostgreSQL.

3.2. Solução Final

A partir dos feedbacks recebidos o grupo fez uma pesquisa dentro dos limites das tecnologias conhecidas para soluções previamente definidas, com o intuito de trazer rapidez e facilidade ao projeto, empregando soluções com reutilização de códigos com templates disponibilizados na rede, mais especificamente nos repositórios abertos.

O banco de dados foi fechado com o retorno dos usuários, após avaliação dos resultados dos formulários, e a sua forma final é apresentada na figura 2, a seguir:

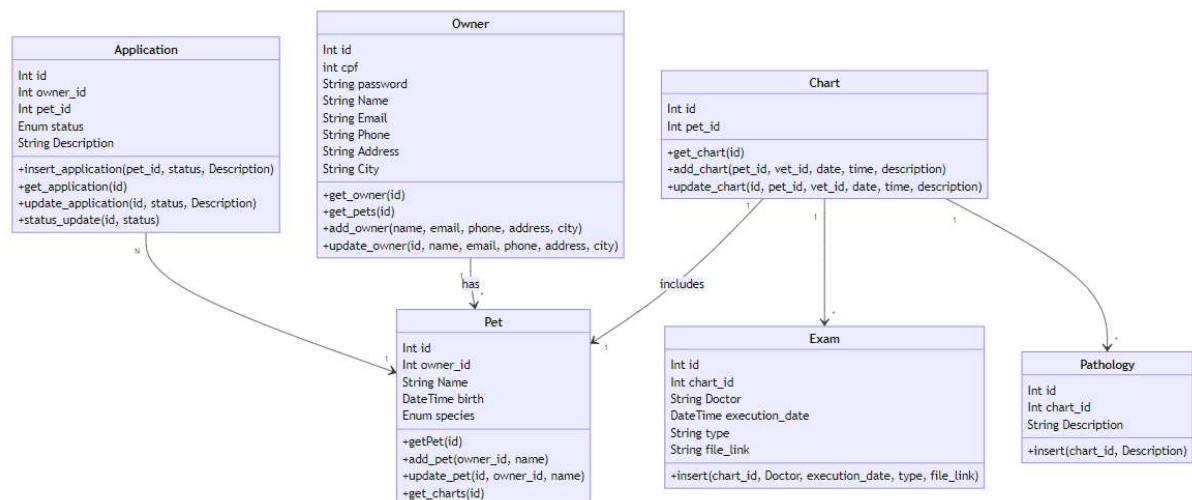


Figura 2 – Diagrama de Classes.

A partir dos resultados destes feedbacks, apresentados no **Anexo 3** as respostas mais importantes foram:

1. Qual é o principal problema encontrado no atendimento no veterinário?

- Demora no atendimento;
- dificuldade de agendamento.

2. Qual o tempo médio que seu pet leva para ser atendido?

- Mais de 1 hora;
- Cerca de 1 hora.

No que concerne à API, foi feita com FastAPI e o design foi incrementado para o layout da interface conforme apresentado na figura 3, a seguir:

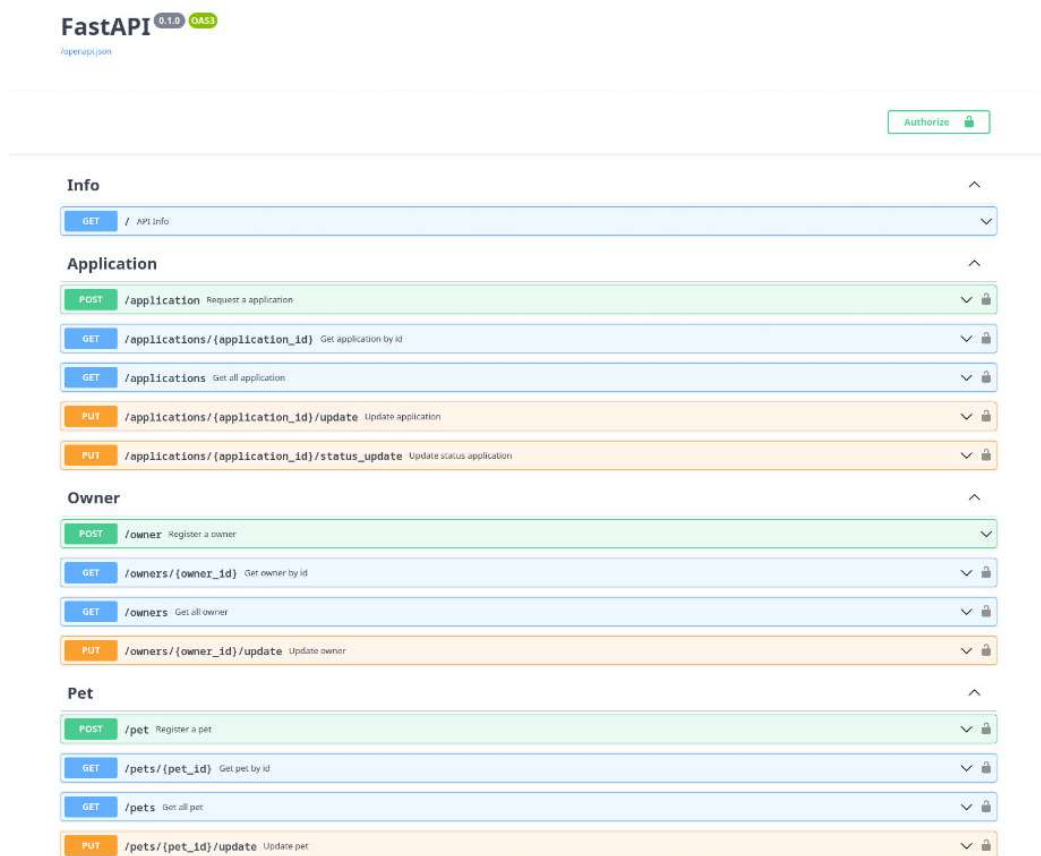


Figura 3 – FastAPI e teste de API.

Os testes foram executados com uso da API via PostMan e seus resultados foram satisfatórios, as persistências não produziram erro, as regras do banco funcionaram perfeitamente e não houve problemas de integração.

A aplicação produzida tem o *login* como mostrado na figura 4, a *home* na figura 5 e o formulário na figura 6, a seguir:




Figura 4 – Login.



Figura 5 – Home Page.

AgendaVet



[Home](#)[Solicitar atendimento](#)[Status](#)

Formulário de Cadastro

Nome do Proprietário	Nome do Pet
<input type="text"/>	<input type="text"/>
Data de nascimento	Data de nascimento do Pet
<input type="text" value="dd / mm / aaaa"/>	<input type="text" value="dd / mm / aaaa"/>
CPF	Especie do PET
<input type="text"/>	<div>Cachorro ▾</div>
Email	Senha
<input type="text"/>	<input type="text"/>
Telefone	Confirme a senha
<input type="text"/>	<input type="text"/>
CEP	
<input type="text"/>	
Endereço	
<input type="text"/>	
Complemento	
<input type="text"/>	
Cidade	
<input type="text"/>	
Estado	
<div>Acre ▾</div>	

Cadastrar

Figura 6 – Formulário

4. CONSIDERAÇÕES FINAIS

Este trabalho, que teve como objetivo introduzir os integrantes a um processo de desenvolvimento de um software web, cujo estudo de caso foi o desenvolvimento de uma aplicação web, atendendo aos estudos já conduzidos, ao processo de criação baseado no design thinking e nas necessidades de clientes reais, pode tratado como atendido, uma vez que a aplicação foi desenvolvida, e os processos e métodos foram aprendidos e bem utilizados.

Como mencionado pela Abinpet (Associação Brasileira da Indústria de Produtos para Animais de Estimação) o Brasil tem uma enorme população de animais de estimação, como o 3º maior mercado do mundo, e por isso é um setor em pleno crescimento e pronto para absorver as inovações da área de tecnologia e aderir a 4ª revolução industrial.

Mesmo com o aumento da importância destes companheiros no ambiente familiar, a rotina estressante as longas esperas por triagem e avaliação dos maiores inconvenientes nos ambientes hospitalares.

Com isso o AgendaVet aparece com intenção de aprimorar a experiência de usuário, aliviando o estresse e a espera nestes ambientes.

O uso de frameworks e bibliotecas conhecidas aumenta a rapidez no desenvolvimento, aumenta a possibilidade de escalar a solução e também o suporte as tecnologias, tanto no deploy da aplicação, quanto na sua manutenção e teste.

O impacto para os consumidores espera-se que seja muito positivo e de grande valor para aliviar as dificuldades de atendimento em uma hora tão preocupante quanto esta que é a de cuidar. Apesar de ser uma aplicação simples e protótipo, ela contribui muito para o bem estar, agilidade e confiança no processo de atendimento.

REFERÊNCIAS

SEBRAE, **Loja de Animais – Pet Shop**, Ideias de Negócios. São Paulo, 2020, 33 páginas, disponível em: <<https://www.sebrae.com.br/sites/PortalSebrae/ideias/como-montar-uma-loja-de-animais-ou-pet-shop,88187a51b9105410VgnVCM1000003b74010aRCRD#apresentacao-de-negocio>>, Acesso em Maio de 2022;

Macedo, F. D., **Gerenciamento de Hospitais Veterinários UNIVET**. Trabalho de Conclusão de Curso – Centro Universitário UNIFACVEST, Curso de Ciência da Computação, 2013;

Martins, T. F., Junior, G. T., **Hipet – Management Systems of Pets**. 10ª Jornada Acadêmica da Jornada da UEG “Integrando saberes e construindo conhecimento” - Campus Santa Helena de Goiás, GO, Novembro de 2016;

Santiago, B. L., Nogueira, L. M., **Vet Treck: “Amar é Cuidar”**. Trabalho de Conclusão de Curso – ETEC Poá, Curso de Informática para Internet integrado ao ensino médio, 2021;

Fernandes, E. I., **Plano de Negócios: Cat Shop – Mundo Felino**. Trabalho de Conclusão de Curso – Universidade do Sul de Santa Catarina UNISUL, Curso de Administração, 2018;

Agostinelli, G. A., **Relatório de Estágio Supervisionado I**. Relatório de Estágio Supervisionado – Universidade Federal do ABC, Curso de Bacharelado em Ciência da Computação, 2018;

Junior, C. L. S., Ferreira, L. N., **Tecnologia da Informação para Apoio a Gestão de Clínicas Veterinárias**. Trabalho de Graduação – Faculdade de Tecnologia de Franca FATEC, Curso de Tecnologia em Análise e Desenvolvimento de Sistemas, 2020;

Mendes, J. V., **Plano de Negócios para o “Agenda XPET APP”: Uma Proposta para um Aplicativo de Agendamento de Serviços de Pet Shop**. Trabalho Conclusão de Curso – Universidade Federal de Minas Gerais, Curso de Pós-Graduação de Gestão Estratégica, 2021.

ANEXO 1

Formulário de Pesquisa.

Consultas em Clínicas Veterinárias e Pet Shops

[Seu e-mail](#)(não compartilhado) [Alternar conta](#)

Quantos pets vivem na sua casa?

- ☐ Nenhum
- ☐ 1
- ☐ 2
- ☐ Outro:

Que tipo de animal é(são) seu(s) pet(s)?

- ☐ Gato
- ☐ Cachorro
- ☐ Pássaro
- ☐ Peixe
- ☐ Tartaruga
- ☐ Coelho
- ☐ Outro:



Aproximadamente, com qual frequência você busca serviços de atendimento médico-veterinário para seu(s) pet(s)?

- ☐ Menos de uma vez a cada 2 anos
- ☐ Uma vez a cada 2 anos
- ☐ 1 vez por ano
- ☐ 2 vezes por ano
- ☐ 3 vezes por ano
- ☐ Uma vez a cada 2 meses
- ☐ Uma vez por mês
- ☐ Mais de uma vez por mês
- ☐ Opção 5

Qual é o principal problema encontrado no atendimento no veterinário?

Sua resposta

Qual o tempo médio que seu pet leva para ser atendido?

Sua resposta

Enviar

[Limpar formulário](#)

Nunca envie senhas pelo Formulários Google.

Este formulário foi criado em Universidade Virtual do Estado de São Paulo. [Denunciar abuso](#)

Google Formulários



ANEXO 2

Código Inicial da API.

```

import os
import mimesis
from fastapi import FastAPI, Request, HTTPException, status
from pydantic import BaseModel
from fastapi import Depends, Security
from fastapi.security import HTTPAuthorizationCredentials
from fastapi.security import HTTPBearer
from jose import JWTError, jwt

SECRET_KEY = os.getenv("SECRET_KEY", "secret")
ALGORITHM = "HS256"
def validate_jwt(
    token: HTTPAuthorizationCredentials = Depends(HTTPBearer()),
) -> dict:
    try:
        return jwt.decode(token.credentials, SECRET_KEY,
                           algorithms=[ALGORITHM])
    except JWTError as e:
        raise HTTPException(
            status_code=status.HTTP_401_UNAUTHORIZED,
            detail="Could not validate credentials",
            headers={"WWW-Authenticate": "Bearer"},
        ) from e

app = FastAPI()
@app.get("/", summary="API Info", tags=["Info"])
def read_root(request: Request):
    return {
        "doc": f"{request.url}docs",
        "redoc": f"{request.url}redoc",
        "json": f"{request.url}json",
    }

class Owner(BaseModel):
    id: int
    name: str
    age: int

@app.get("/owners", dependencies=[Security(validate_jwt)], summary="Get all owners",
tags=["Owners"])
def list_owners():
    owner = mimesis.Person()
    return [Owner(id=i, name=owner.full_name(), age=owner.age()) for i in range(1, 5)]

@app.get("/owners/{owner_id}", dependencies=[Security(validate_jwt)], summary="Get Owner",
tags=["Owner"])
def get_owner(owner_id: int):
    fake = mimesis.Person()
    return Owner(id=owner_id, name=fake.full_name(), age=fake.age())

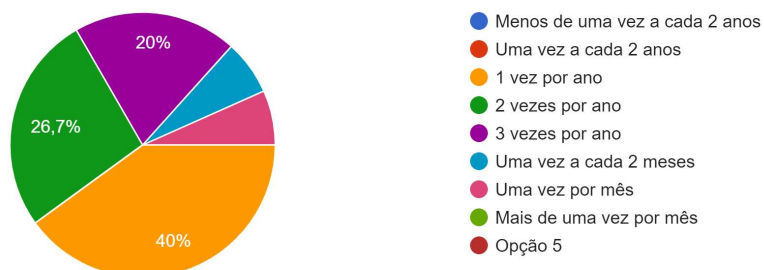
```


ANEXO 3

Resultados da Pesquisa.

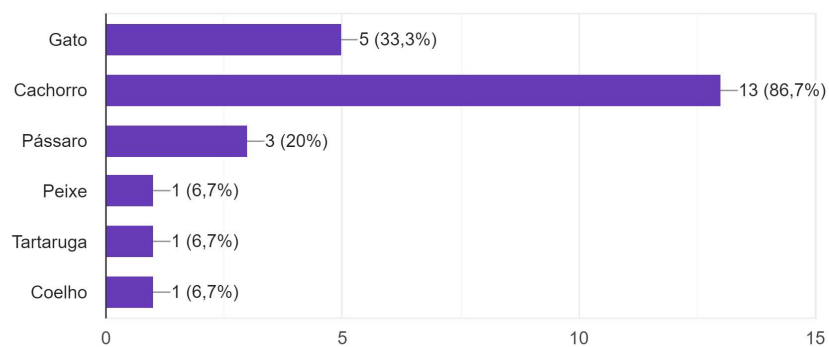
Aproximadamente, com qual frequência você busca serviços de atendimento médico-veterinário para seu(s) pet(s)?

15 respostas



Que tipo de animal é(são) seu(s) pet(s)?

15 respostas



Quantos pets vivem na sua casa?

15 respostas

