

**FACULDADE SENAC FLORIANÓPOLIS**  
**Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas**

**Guilherme Valmir de Andrade**

**PET LOVE:**  
**SOLUÇÃO DE SERVIÇOS PARA DONOS DE ANIMAIS DE ESTIMAÇÃO**

**Guilherme Valmir de Andrade**

**PET LOVE:**  
**SOLUÇÃO DE SERVIÇOS PARA DONOS DE ANIMAIS DE ESTIMAÇÃO**

Trabalho apresentado à Faculdade Senac Florianópolis  
como requisito parcial para obtenção do título de  
Tecnólogo em Análise e Desenvolvimento de Sistemas.

Florianópolis  
2021

## **RESUMO**

Os animais de estimação têm se tornado cada dia mais realidade na vida das pessoas. Tendo este fato em vista, criou-se um mercado enorme acerca desta área. Hoje podemos encontrar quase todos os serviços que antes eram destinados aos seres humanos – emissão de documentos, hotéis, hospitais, casamentos etc. – disponíveis para os animais de estimação. Tendo em vista o crescimento constante do mercado de serviços para animais de estimação, foi permitido desenvolver uma solução destinada aos donos de animais de estimação que conta basicamente com três produtos: uma API REST desenvolvida na linguagem de programação PHP que provê e gerencia todos os dados da aplicação; Um website responsivo que servirá como um layout de entrada e saída de dados para donos de animais de estimação e empresas que queiram oferecer seus serviços no sistema; além de um aplicativo onde os clientes poderão carregar consigo as informações dos seus animais de estimação.

Palavras-chave: Animais de estimação. Mercado. Crescimento. Clientes.

## 1 INTRODUÇÃO

Quando recorremos à história para compreender a relação entre seres humanos e animais de estimação, sempre nos esbarramos com o contexto da época. Nos primórdios da humanidade, os animais eram domesticados para serem utilizados como armas durante as caçadas. Quando o homem começou a formar suas primeiras civilizações, os animais de estimação ainda eram vistos em grande parte como uma ferramenta para proteger residências ou ajudar no trabalho. Hoje, embora a tecnologia deixe a experiência de interação entre pessoas cada vez mais facilitada, há cada vez menos afeto no mundo. As prioridades têm sido apontadas para outras facetas da vida como o trabalho ou o entretenimento em geral e a aproximação real entre pessoas tem sido deixada de lado. Dado esse distanciamento, os animais de estimação têm preenchido essa lacuna dando o carinho e a companhia que as pessoas necessitam.

Este protagonismo que os animais de estimação vêm ganhando – e, conseqüentemente o aumento da aquisição destes companheiros (o Brasil é o quarto país com maior população de animais de estimação no mundo segundo o IBGE em 2015) - gerou uma série de dores e necessidades aos seus donos que foram prontamente enxergadas pelo mercado.

Se olharmos para o passado, os animais de estimação ficavam geralmente presos no quintal de casa e se alimentavam apenas das sobras dos seus donos. Hoje temos alimentos destinados especificamente aos animais separados por idade e raça que oferecem todos os nutrientes necessários. Com o aumento considerável dos animais de estimação na casa das pessoas, aumentaram também os estudos acerca deles. Hoje os pets estão amplamente amparados na área médica e psicológica sendo facilmente diagnosticados dependendo do seu comportamento. Houve a necessidade de criar hospitais que tratam somente de animais, as chamadas clínicas veterinárias. Hoje o trabalho consome parte considerável do nosso dia e muitas vezes não sobra tempo para atividades básicas com os animais de estimação. Para sanar esta necessidade, foi criada uma profissão chamada “pet walker”, que é um profissional que cobra para levar animais de estimação para passear. Podemos citar muitos outros serviços que foram criados nos últimos anos devido ao aumento da presença de

animais de estimação nas famílias como: emissão de documentos para animais, casamento para animais, hotel para animais etc.

Com o aumento da aquisição de smartphones, o mercado de aplicativos também sofreu um crescimento e serviços para os pets têm sido oferecidos de forma abrangente através de apps. Após uma pesquisa de mercado, houve a percepção que muitos destes serviços são oferecidos de forma separada e algumas vezes insatisfatória. Desta maneira, há a intenção de oferecer os principais serviços que os donos de animais de estimação necessitam em um único lugar, sem precisar instalar diversos aplicativos ou se cadastrar em diversos serviços para isto. A ideia é oferecer um serviço onde o cliente possa carregar consigo informações importantes sobre os seus pets, além de poder encontrar serviços essenciais como clínicas, pet shops e dog walkers.

Para obter os dados necessários para justificar a existência da solução apresentada, será feita uma pesquisa que tem como propósito um método descritivo e uma abordagem quali-quantitativa. A pesquisa terá como cenário grupos de donos de animais de estimação da internet.

A seguir serão apresentadas as etapas através das quais foi desenvolvida esta aplicação.

## **2 OBJETIVOS**

### **2.1 Objetivo Geral**

Criar uma aplicação onde os donos de animais de estimação possam carregar consigo as principais informações dos seus pets, além de poderem encontrar serviços essenciais para eles como: clínica veterinária, pet shop e dog walkers.

### **2.2 Objetivos Específicos**

- Desenvolver um protótipo navegável da solução web e do aplicativo para demonstrar em apresentação para a banca.
- Levantar dados de donos de animais de estimação referentes a qualidade dos serviços oferecidos no mercado no que diz respeito a saúde e bem-estar do seu animal de estimação.
- Fazer uma pesquisa de mercado acerca das soluções atualmente oferecidas e como elas atendem os clientes.
- Através dos dados obtidos com a pesquisa de mercado e as informações dos clientes, levantar requisitos mínimos para o funcionamento da solução.
- Tendo como base os requisitos levantados, modelar os produtos oferecidos: API REST, website e aplicativo.
- Desenvolver os produtos tendo como base o modelo criado.
- Validar protótipo funcional.
- Testar versão 1.0 da aplicação.

### 3 JUSTIFICATIVA

Segundo a Associação Brasileira da Indústria de Produtos para Animais de Estimação (Abinpet) em uma contagem realizada em 2020, há 144,3 milhões de animais de estimação nos lares brasileiros. Destes, 55,9 milhões são cães, 40,4 milhões são aves, 25,6 milhões são gatos, 19,9 milhões são peixes e há também 2,5 milhões de outros tipos de animais domesticáveis como coelhos e répteis. Se compararmos com 2019, houve 1,5% de aumento no número de cães no Brasil e 3,6% de aumento no número de gatos. Ainda segundo a Abinpet, houve um faturamento em 2020 de 27 bilhões de reais sendo, destes, 75% vindos da venda de comida para pets, 17% da área de saúde animal e 8% da área de serviços gerais para pets como dog walkers. O serviço de saúde animal cresceu 18% em relação a 2019 e os serviços gerais tiveram um crescimento de 9,5%. Se olharmos o mercado mundial em 2020, houve um faturamento de 145,8 bilhões de dólares e, deste valor, 3,9% correspondem ao mercado brasileiro, fazendo do Brasil o 7º colocado no mercado pet (Abinpet, 2021).

Em uma pesquisa realizada pela Folha de São Paulo, vemos que houve a criação de 31.639 novas lojas do segmento pet em 2018 sendo, destas, 31.066 lojas de pequeno e médio porte. Já em 2019, houve a criação de novas 32.675 lojas do segmento pet, sendo 32.082 lojas de pequeno e médio porte. Tivemos um crescimento de 3,27% na criação de lojas de bairro e 3,26% na criação de lojas de médio porte se compararmos 2018 e 2019. Quando falamos de faturamento no mercado nacional, tivemos 34,6 bilhões de reais em 2019 e 40,1 bilhões de reais em 2020, ou seja, um crescimento de 13,5%. Deste faturamento, 50% vieram de comida para pet, 12,1% da venda de animais por criadores, 11,8% de veterinárias, 10,4% de serviços gerais, 10,3% de serviços veterinários e 5,6% de pet care (Folha de São Paulo, 2021).

Estes dados nos mostram um claro crescimento do mercado pet nos últimos anos e uma projeção de crescimento para o futuro.

Com cada vez mais pets nos lares brasileiros e seus donos com cada vez menos tempo para cuidar deles, é necessário que haja serviços que agilizem e facilitem esse


processo. Há diversas soluções no mercado que visam auxiliar donos de animais de estimação. Temos serviços onde clientes podem encontrar produtos para os seus pets, serviços onde o cliente pode encontrar clínicas veterinárias próximas e marcar consultas nessas clínicas, serviços onde o cliente pode encontrar cuidadores e/ou hospedagem para os seus pets, entre muitos outros. Esse tipo de solução tem se tornado cada vez mais necessário.

Entre tantas soluções oferecidas no mercado que prometem sanar os mais variados problemas relacionados aos pets, foi percebida a falta de uma solução que abranja os principais serviços em um sistema só. Isso evita que o cliente instale diversos aplicativos ou se cadastre em diversos sistemas a fim de resolver um pequeno conjunto de soluções básicas que seu pet possa precisar. Além disso, nas soluções que oferecem uma lista de serviços (como a marcação de consultas em clínicas, por exemplo), não foi encontrado um tratamento satisfatório de feedbacks dos clientes em relação à empresa que ofereceu determinado serviço.

Através de pesquisas de mercado, chegamos a outros três serviços que oferecem serviços semelhantes ao que ofereceremos na solução proposta: 11 Pets, que é um aplicativo de lembretes para os animais de estimação. Através de cadastros prévios do dono, o aplicativo lembra o dono de tarefas básicas com alimentação até tarefas mais específicas como vacinação; Dog Hero, que é um aplicativo onde o dono pode encontrar serviços de hospedagem e cuidadores para os seus pets. Além disso, há um e-commerce parceiro integrado ao serviço que permite a compra de produtos pet; Pet Booking, que é um aplicativo onde o cliente pode encontrar clínicas veterinárias, pet shops e cuidadores para os seus pets. O diferencial deste aplicativo é que há a possibilidade de marcar consultas.



Figura 1 – Quadro comparativo entre as soluções encontradas

				
Oferece um histórico dos seus animais de estimação				
Oferece uma lista de serviços para o pet				
Possibilita a marcação de consulta para o pet				
Oferece serviço de dog walker para o pet				
Possibilidade de upload dos exames veterinários				

Fonte: elaboração do autor (2021).

Como mostra a Figura 1 acima, os três serviços citados oferecem soluções básicas de forma separada. A ideia da solução proposta é unir parte dos serviços que estas três soluções oferecem em uma única solução. Teremos um serviço onde o cliente poderá cadastrar informações dos seus pets e carregá-las consigo (além de poder compartilhá-las), encontrar clínicas, pet shops e cuidadores próximos além de haver a possibilidade de marcar consultas ou demais serviços com essas empresas. Além disso, os clientes terão acesso aos feedbacks dados por outros donos de pets nas clínicas cadastradas. A solução proposta oferece também a possibilidade de upload dos exames veterinários, diferente das demais soluções, que oferecem apenas a possibilidade de escrita dos exames.

## 4 FUNDAMENTAÇÃO TEÓRICA

Para melhor compreensão da solução que está sendo proposta, este capítulo se dedica exclusivamente a explicar os principais conceitos acerca do desenvolvimento da aplicação.

### 4.1 API

API (Application Programming Interface) ou "Interface de Programação de Aplicações" se trata de um conjunto de padrões pré-estabelecidos no desenvolvimento de um software de modo que ele possa compartilhar informações com outros softwares. A ideia principal de uma API é a integração de modo que sistemas possam se comunicar sem que haja um profundo conhecimento de como cada software foi desenvolvido. Uma API pode ser vista como um contrato. Para assinar este contrato, eu preciso obedecer às regras previstas em sua documentação. Se eu obedeco aos padrões que o contrato impôs, eu recebo o resultado esperado e o contrato é concluído. Um artigo do blog [take.net](https://take.net) utiliza uma metáfora para explicar, grosso modo, o funcionamento de uma API:

"... podemos pensar no que é API como um garçom. Quando estamos em um restaurante, buscamos o que desejamos no menu e solicitamos ao garçom. O garçom encaminha esse pedido à cozinha, que prepara o pedido. No fim, o garçom traz o prato pronto até a gente. Não temos detalhes de como esse prato foi preparado, apenas recebemos o que solicitamos" ([take.net](https://take.net), 2019).

Entendemos, então, que uma API é uma interface de aplicação que segue um conjunto de rotinas que, quando obedecidas por outros softwares, possibilita a comunicação entre eles. As APIs utilizam formatos de dados que podem ser lidos e tratados por qualquer linguagem de programação de alto nível, geralmente estes formatos de dados são JSON e XML. Independente da tecnologia que eu utilizo para desenvolver uma API, eu terei como entrada e saída um dado que é compreendido em qualquer linguagem de programação. Isso faz com que a tecnologia utilizada entre os sistemas que estão se comunicando seja irrelevante.

As vias pelas quais as APIs se comunicam são as requisições HTTP. As requisições HTTP são mensagens que são enviadas do cliente (máquina do usuário)

para o servidor. Entre os principais elementos que uma requisição HTTP possui estão o cabeçalho, o método, o alvo e o corpo. No cabeçalho da requisição estão definidas as suas regras gerais como o status da mensagem, o tipo de mensagem retornada, o tipo de conteúdo que está sendo transmitido, entre outras. O método da requisição indica a ação que está sendo esperada nessa requisição. Temos quatro métodos principais: GET, POST, PUT e DELETE. O alvo é o endereço (endpoint) para o qual se está enviando a requisição. Já no corpo da requisição estão os dados que estamos enviando para ela. Quando se trata de uma API, na maioria dos casos a estrutura de dados enviada no corpo da requisição é do tipo JSON.

Contudo, uma API é uma interface de aplicação desenvolvida utilizando qualquer linguagem de programação de alto nível que, através de requisições HTTP, transmite dados que são compreendidos independente da linguagem de programação utilizada na aplicação que está acessando os dados da API, geralmente JSON.

Para que houvesse organização no desenvolvimento de APIs, assim como ocorre com linguagens de programação, foram implementadas uma série de convenções para que a manutenção em uma API se torne viável. Suponhamos que um time de desenvolvimento crie uma API e utilize seus próprios padrões de código neste desenvolvimento. Se os padrões de desenvolvimento utilizados nessa API são conhecidos somente pelo time que a desenvolveu, a manutenção por outra equipe de desenvolvedores será extremamente trabalhosa. Será necessário muito tempo para que uma outra equipe entenda os padrões de desenvolvimento utilizados na API de modo que eles possam inserir melhorias ou atualizações. Tendo este cenário em vista, se faz necessário o uso de convenções universalmente conhecidas para que a manutenção seja viável por qualquer desenvolvedor ou time de desenvolvimento que conheça os padrões utilizados, pois, sendo esses padrões universais, a chance de familiaridade por desenvolvedores alheios é consideravelmente maior. O padrão de desenvolvimento de API que iremos nos ocupar será o padrão REST.

Para que uma API seja considerada como padrão REST, temos alguns critérios principais que precisam ser seguidos:

- Ter uma arquitetura cliente/servidor. Temos um cliente (usuário) que faz solicitações onde pode enviar ou não dados ao servidor (API) por meio de solicitações que são enviadas via protocolo HTTP.
- A comunicação entre cliente e servidor precisa ser "stateless", ou seja, a solicitação não pode armazenar informações e utilizá-las em solicitações futuras. Cada solicitação tem que ser feita de maneira isolada.
- Ter métodos bem estabelecidos. Os métodos das minhas requisições HTTP precisam seguir o intuito para o qual foram criados. Quando queremos uma lista de dados como resposta do servidor, temos o método GET. Quando queremos inserir um novo recurso no banco de dados, temos o método POST e no corpo da requisição HTTP temos um objeto que queremos inserir. Já quando queremos atualizar um recurso, temos o método PUT, e no corpo da requisição HTTP temos o objeto que queremos atualizar. Já quando queremos excluir um recurso, usamos o método DELETE.
- O desenvolvimento precisa ser distribuído em camadas para que o acesso a certos recursos por determinados tipos de usuários possa ser controlado.

Nesta solução teremos uma API REST. Uma interface de desenvolvimento onde um cliente (usuário) faz requisições ao servidor por meio de requisições HTTP seguindo os padrões impostos pela API, obtendo uma resposta.

## **4.2 SISTEMA WEB**

Primeiramente vamos falar sobre sistemas em si. Um sistema é um software que oferece solução para automatizar ou eliminar processos que antes tomavam mais tempo e dinheiro, tornando a entrega do produto mais ágil, barata e eficaz. Podemos tomar como exemplo um mercado de bairro onde o dono guarda todas as transações realizadas em um caderno. Se ele precisar saber quanto ele gastou com o fornecimento de produtos nos últimos três meses, precisará pegar o seu caderno e uma calculadora. Outro fato a se considerar é que o caderno é um material físico, ou seja, não pode estar em dois lugares ao mesmo tempo e só pode ser acessado

quando o dono vai até o mercado e abre a gaveta onde o guarda. O caderno, sendo um material físico, está sujeito às condições do tempo e do acaso, podendo envelhecer, queimar ou ser roubado. Para este cenário, um sistema é muito bem-vindo. Com um sistema instalado e configurado em seu mercado, o dono poderia ter acesso aos seus gastos em tempo real, podendo tomar decisões com muito mais facilidade e precisão.

Quando adicionamos a palavra "web" a sistema, estamos apenas nos referindo ao ambiente onde este sistema está funcionando. Um sistema web é um software que fica hospedado em um servidor web, podendo ser acessado por meio de um navegador. Um sistema web é diferente de um sistema desktop, por exemplo, onde o sistema é instalado na máquina do cliente. Uma grande vantagem de um sistema web em relação a sistemas desktop é o seu processo de atualização. Digamos que eu tenha um sistema web e um sistema desktop em sua versão 1.0 e quero lançar uma versão 2.0 com diversas melhorias críticas. Num sistema desktop, eu precisaria criar um executável com a atualização e pedir para o usuário do sistema baixar e instalar em sua máquina. Já em um sistema web, depois que a atualização é feita pelos desenvolvedores, basta que o usuário abra novamente o navegador ou aperte a tecla F5 com o navegador já aberto e as melhorias já estarão em funcionamento.

Na solução proposta, será disponibilizado aos usuários um sistema web onde as empresas poderão gerenciar suas agendas e os clientes poderão marcar serviços para os seus animais de estimação. Este sistema acessará a API mencionada no tópico anterior.

### **4.3 APLICATIVO MOBILE**

Um aplicativo mobile é um software desenvolvido para funcionar em smartphones e tablets. O desenvolvimento de aplicativos mobile está geralmente associado a serviços que precisam oferecer mobilidade ao cliente. Um bom exemplo de um serviço onde a mobilidade é essencial é o serviço de contrato de motoristas. Se o cliente está em algum ponto da cidade e precisa de um motorista para levá-lo a algum lugar, é extremamente inviável que ele precise de um computador desktop para acessar o

serviço onde ele irá contratar um motorista. Neste caso, o acesso ao serviço através do seu smartphone é essencial.

Na solução proposta, será disponibilizado aos usuários donos de animais de estimação um aplicativo onde eles poderão guardar consigo as informações básicas e médicas principais do seu pet.

## **4.4 TECNOLOGIAS**

### *4.4.1 Programação Orientada a Objetos (POO)*

Por muitos anos o desenvolvimento era realizado sob o paradigma de programação estruturada. Na programação estruturada nós temos basicamente uma rotina (ou função) principal que se divide em sub-rotinas. Essas sub-rotinas são gerenciadas através de condições (if, else, switch) e repetições (for, while, do-while, foreach). O resultado deste tipo de paradigma são arquivos extensos e diversos trechos de código repetidos que foram copiados e colados de uma rotina para a outra. Este tipo de cenário dificulta bastante a manutenção do projeto, pois se um trecho de código dentro de uma rotina mudar, tenho que me certificar linha a linha de efetuar a mudança nos trechos que foram colados nas sub-rotinas. Além disso, a proteção de acesso a variáveis na programação estruturada é limitado. Não conseguimos dizer de forma nativa que uma determinada variável é acessada somente por determinada função.

Como solução e alternativa à programação estruturada, surgiu a programação orientada a objetos. A ideia principal da programação orientada a objetos (ou POO) é relacionar o ambiente de desenvolvimento com o mundo real. Nosso principal componente na POO é o objeto, nome genérico para representar algo do mundo real, alguma coisa tangível. Na POO temos dois conceitos chave: classes e objetos. Uma classe funciona como um molde de algo que eu estou abstraindo do mundo real, já um objeto é um modelo criado (instanciado) a partir deste molde. Vamos imaginar um software de gerenciamento de veículos. Um veículo é algo do mundo real que eu preciso abstrair no meu software, então eu crio uma classe chamada veículo. Em uma classe, eu tenho atributos e métodos. Se pensarmos em um veículo, temos

características como modelo, ano de fabricação e tipo. Estes são atributos da minha classe veículo. O veículo realiza ações como se locomover e mudar a velocidade. Estes são métodos da minha classe veículo. Com a classe veículo em mãos, eu posso criar (instanciar) diversos objetos veículos em qualquer parte do meu software. Subdividir o software em classes permite que cada classe seja implementada de maneira individual e suas regras de negócio sejam tratadas individualmente e eu possa simplesmente criar um objeto seguindo todas as regras implementadas na classe em qualquer parte do meu software, isso evita a cópia do mesmo código em diversas partes do programa, além de deixar o desenvolvimento mais organizado.

Na POO nós temos alguns pilares que precisam ser explicados: encapsulamento, herança e polimorfismo. Vamos imaginar a nossa classe veículo e nosso método para mudar de velocidade. Quando um veículo acelera ou diminui sua velocidade, há um mecanismo de aceleração dentro dele que faz esta tarefa. Não sabemos como esta tarefa é feita, tudo o que sabemos é que se pisarmos no acelerador, a velocidade irá mudar. Neste caso, criamos uma classe para o mecanismo de aceleração do veículo e encapsulamos esta classe dentro da classe veículo. Como no mundo real, quando estamos em um veículo nós sabemos que ele tem um mecanismo de aceleração, mas não sabemos em detalhes como ele funciona, sabemos apenas que ele é ativado quando pressionamos o acelerador do veículo. Na POO, esta é a ideia de encapsulamento. Já quando falamos de herança, estamos nos referindo a subclasses que herdam propriedades e métodos de classes principais (ou superclasses). Carro, motocicleta e barco são tipos de veículos. Podemos criar diversas subclasses de veículos em um programa com suas próprias características que as diferenciam de outras subclasses, porém todas herdam características em comum da superclasse veículo. Se considerarmos a classe barco, ela pode ter um atributo chamado marina. Este é um atributo que pertence somente a esta classe, diferenciando-a das outras subclasses de veículo. Quando falamos de polimorfismo em POO, estamos falando de formas diferentes que subclasses realizam um mesmo método. Por exemplo, tanto a classe barco quanto a classe carro possuem o método acelerar, porém a forma com que a aceleração se dá é tratada de maneira diferente em cada classe, esta é a ideia do polimorfismo em POO.

Como paradigma de programação no desenvolvimento da API proposta, será utilizada a programação orientada a objetos.

#### *4.4.2 ARQUITETURA MVC*

MVC (ou model-view-controller) é uma arquitetura de projeto no desenvolvimento de aplicações. A ideia principal do MVC é separar o projeto em camadas com responsabilidades claras e bem definidas, isso traz organização ao projeto e facilita sua manutenção e escalabilidade. De maneira geral, o software é dividido em três camadas. A primeira delas é o model (ou modelo). Nesta camada estão todas as nossas entidades (classes) de banco de dados, bem como as regras de negócio e classes responsáveis diretamente pelas transações e pesquisas no banco de dados. A view (ou camada de visão) é a única camada que o cliente pode interagir, ou seja, é a camada responsável pelo layout, pelo que é apresentado na tela. Já o controller (ou controlador) é a camada que faz a comunicação entre a camada de modelo e a camada de visão. Vamos imaginar um processo de login em um sistema que utiliza a arquitetura MVC. O usuário acessa o sistema através de um navegador e é direcionado à tela de login. Neste momento, estamos na camada de visão. Ele digita o seu login, a sua senha e clica no botão para efetuar a ação de entrada no sistema. Neste momento, as informações digitadas na camada de visão são interceptadas pelo controlador, que pega os dados digitados na camada de visão (login e senha) e os envia à camada de modelo, que verifica a integridade dos dados digitados e vai ao banco de dados verificar se há um usuário com o login e a senha informados pelo usuário. Caso haja um usuário com o login e senha informados, a camada de modelo retorna uma resposta positiva à camada controladora, que envia a mensagem de sucesso ao cliente e o direciona para o painel de usuário.

Na solução proposta, será utilizada a arquitetura MVC no desenvolvimento da API.

#### *4.4.3 PHP*

PHP (acrônimo recursivo para PHP: Hypertext Preprocessor) é uma linguagem de programação server-side (que funciona do lado do servidor, ou seja, utilizada na



grande maioria das vezes no back-end) de alto nível e código aberto. Sua primeira versão foi disponibilizada em 1995 por Rasmus Lerdorf. O PHP foi uma linguagem projetada para funcionar no ambiente web, sendo extremamente competitiva em relação a outras linguagens e runtimes que trabalham no mesmo ambiente como Java e NodeJS, principalmente a partir da sua versão 7.0, quando passou a ser extremamente mais segura. Outras grandes vantagens do PHP podem ser mencionadas, como sua facilidade de implementação e sua curva de aprendizado baixa (site oficial do PHP, 2021).

Na solução proposta, será utilizada como linguagem de programação no desenvolvimento da API o PHP.

#### *4.4.4 MYSQL*

O MySQL é uma ferramenta de banco de dados relacional. Com uma boa facilidade de implantação, é um dos servidores de banco de dados mais utilizados do planeta. A licença de uso do MySQL é oferecida de duas formas: gratuita ou sob a compra de uma licença de uso. Caso o seu projeto esteja totalmente sob a licença GPL (licença utilizada para softwares públicos e gratuitos), o uso do MySQL é gratuito, caso contrário, deverá ser comprada uma licença para poder utilizá-lo em seu projeto (site oficial do MySQL, 2021).

A ferramenta de banco de dados utilizada na solução proposta será o MySQL.

## 5 METODOLOGIA

Para que qualquer processo de desenvolvimento tenha mais chances de ser bem-sucedido, é necessário planejamento. Antes de começar a escrever linhas de código, foi feita uma entrevista com donos de animais de estimação. Nesta entrevista, foi possível levantar dados que deixaram o processo de desenvolvimento mais direcionado. Quando o projeto foi iniciado, havia uma ideia generalizada de solução. A análise do resultado desta entrevista trouxe clareza quanto às reais necessidades dos clientes na primeira versão da solução. Esta entrevista tem método descritivo, uma vez que foi proposta com intuito de observar de forma analítica a opinião de um nicho de mercado em relação às suas características e dores acerca do assunto proposto na solução. Além disso, o questionário tem um método de apresentação do tipo quali-quantitativo, ou seja, há tanto perguntas quantitativas (que apresentam números, porcentagem em relação às alternativas escolhidas pelos entrevistados) quanto perguntas qualitativas (que requerem a opinião de cada entrevistado e são analisadas caso a caso).

Foi realizada também uma pesquisa de mercado para ver que tipo de serviços semelhantes aos propostos nesta solução estão sendo atualmente oferecidos aos clientes. Dessa maneira, podemos observar softwares que já passaram por processos de validação em funcionamento. Utilizando softwares concorrentes, podemos fazer uma análise crítica das suas principais funcionalidades, bem como propor melhorias para elas ou funcionalidades novas em nossa solução.

Com os resultados da entrevista e da pesquisa de mercado em mãos já temos uma ideia de viabilidade da solução, bem como condições para escrever seus requisitos. Nesta etapa do desenvolvimento foram levantados os requisitos funcionais e não funcionais, bem como diagramas UML de casos de uso e classes para compreender a estrutura dos dados e das responsabilidades dos usuários como um todo. Foi levantado também o modelo lógico do banco de dados.

Com a parte de análise devidamente desenvolvida, podemos iniciar o desenvolvimento da aplicação. Durante todo o projeto, foi utilizada a arquitetura MVC (model-view-controller) por um motivo de organização de código e clareza das

responsabilidades de cada parte do software. Para melhor distribuição e organização do projeto, a solução foi dividida em partes. Foi desenvolvida uma API responsável pelo gerenciamento e provisionamento dos dados, bem como toda a parte de regra de negócio e validação de dados de entrada. Para consumir os dados da API, foram desenvolvidos um sistema web que trabalha somente com a camada de visão e um aplicativo mobile que o cliente utilizará para carregar os dados de exames, consultas e vacinação dos seus animais de estimação. Como ferramenta de banco de dados, foi utilizado o MySQL. Além disso, para um controle de versionamento e atualização de código durante o processo de desenvolvimento, todo o projeto foi colocado num repositório em uma página na internet.

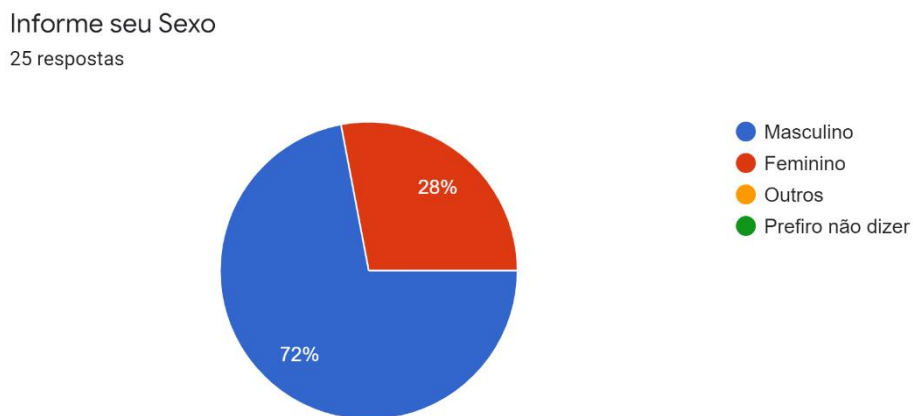
## 6 DESENVOLVIMENTO

Nesta seção serão apresentadas em detalhes todas as atividades realizadas durante o processo de desenvolvimento de software, desde as pesquisas realizadas, passando pelos diagramas UML e requisitos até a codificação em si e seus respectivos testes.

### 6.1 Entrevista

A primeira atividade durante o desenvolvimento desta solução foi a elaboração de um questionário que foi respondido por 25 donos de animais de estimação. Este questionário foi distribuído através da ferramenta Google Forms. Esta entrevista teve como objetivo analisar o perfil do público alvo, além de coletar suas principais necessidades e dores com os serviços atuais. Vamos analisar as respostas de cada pergunta.

Figura 2.1: Entrevista (questão 1)

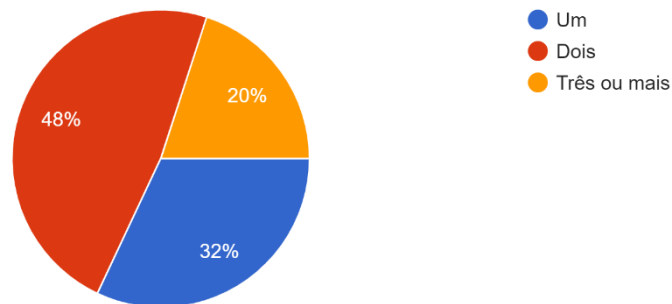


Fonte: elaboração do autor (2021).

Na primeira pergunta, pedi aos entrevistados que informassem seu sexo. Nesta entrevista, 28% dos entrevistados são mulheres e 72% são homens.

Figura 2.2: Entrevista (questão 2)

Quantos animais de estimação você possui?  
25 respostas

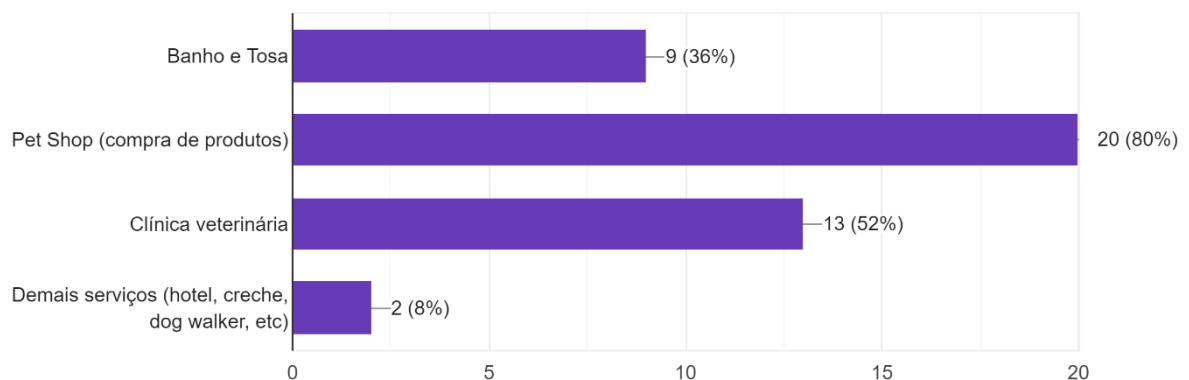


Fonte: elaboração do autor (2021).

A segunda pergunta foi em relação a quantidade de animais de estimação presentes nas casas dos entrevistados. 48% dos entrevistados possuem dois animais de estimação, 32% possuem um animal de estimação e 20% possuem dois ou mais.

Figura 2.3: Entrevista (questão 3)

Quais destes serviços você mais utiliza para o seu animal de estimação?  
25 respostas



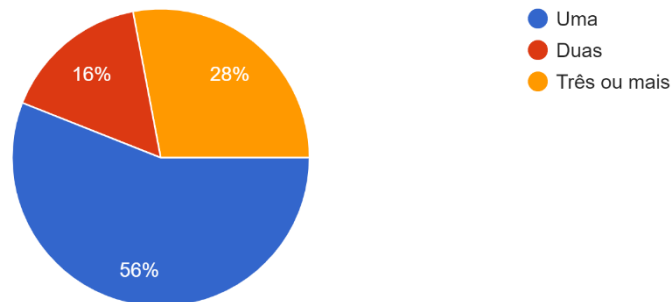
Fonte: elaboração do autor (2021).

A terceira pergunta foi em relação ao tipo de serviço que os donos de animais de estimação entrevistados mais utilizam para os seus pets. Nesta questão, era possível

responder mais de uma opção. Dos 25 entrevistados, 80% utilizam pet shop, 52% utilizam clínica veterinária, 36% utilizam banho e tosa e 8% utilizam demais serviços.

Figura 2.4: Entrevista (questão 4)

Quantas vezes por mês em média você utiliza serviços para o seu animal de estimação?  
25 respostas

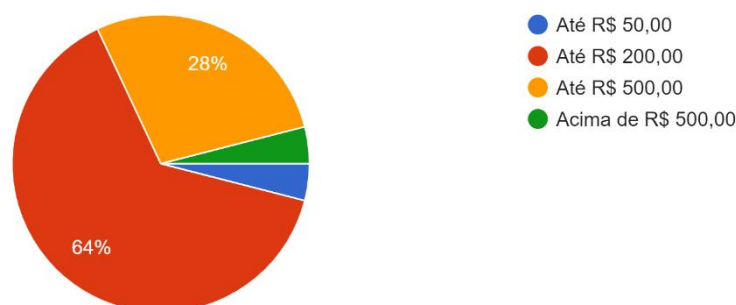


Fonte: elaboração do autor (2021).

A quarta pergunta foi quantas vezes por mês os entrevistados costumam utilizar serviços para os seus animais de estimação. 56% dos entrevistados utilizam uma vez apenas, 28% dos entrevistados utilizam serviços três vezes ou mais ao mês e 16% dos entrevistados utilizam duas vezes ao mês.

Figura 2.5: Entrevista (questão 5)

Qual a sua média de gastos mensais com o seu animal de estimação?  
25 respostas



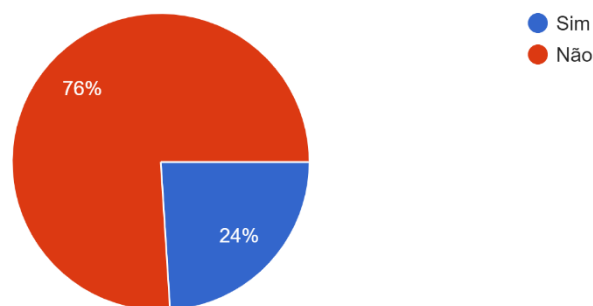
Fonte: elaboração do autor (2021).

A quinta pergunta foi em relação aos gastos mensais que os entrevistados têm com os seus animais de estimação. 64% dos entrevistados gastam até R\$ 200,00, 28% dos entrevistados gastam até R\$ 500,00, 4% dos entrevistados gastam até R\$ 50,00 e 4% dos entrevistados gastam R\$ 500,00 ou mais.

Figura 2.6: Entrevista (questão 6)

Você utiliza algum aplicativo para auxiliar no cuidado e gerenciamento do seu animal de estimação?

25 respostas



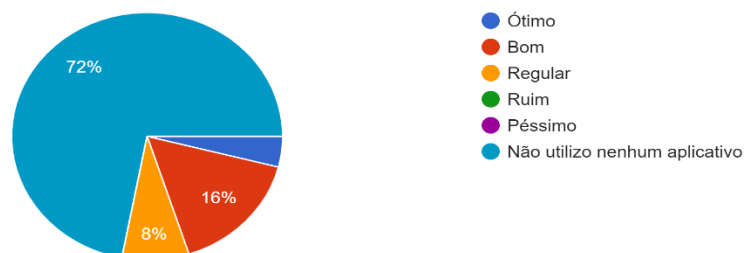
Fonte: elaboração do autor (2021).

A sexta pergunta indaga os entrevistados questionando se eles possuem ou não algum aplicativo que ofereça algum tipo de serviço para os seus animais de estimação. 78% dos entrevistados não utilizam qualquer tipo de aplicativo para os seus pets, já 24% dos entrevistados utiliza aplicativos.

Figura 2.7: Entrevista (questão 7)

De maneira geral, como você avalia o(s) aplicativo(s) que você utiliza para auxiliar no cuidado do seu animal de estimação?

25 respostas



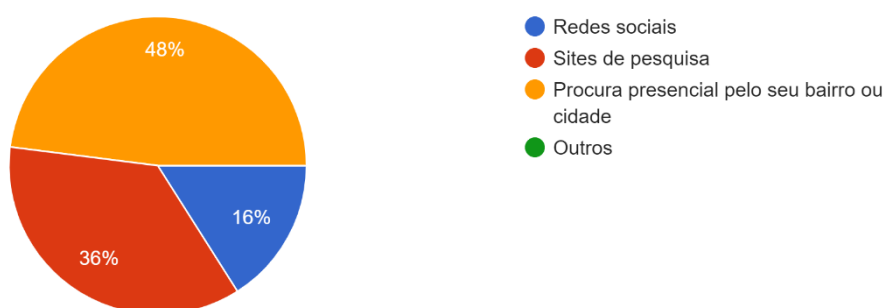
Fonte: elaboração do autor (2021);

A sétima pergunta foi para os entrevistados que utilizam aplicativos para os seus pets. Dos entrevistados que utilizam aplicativos (sete, o que equivale a 28% do total de entrevistados), 57% avaliam os aplicativos que utilizam como bons, 29% como regulares e 14% como ótimos.

Figura 2.8: Entrevista (questão 8)

Qual meio de comunicação você mais utiliza para procurar serviços para o seu animal de estimação?

25 respostas



Fonte: elaboração do autor (2021).

A oitava pergunta (e última quantitativa) foi em relação ao meio de comunicação que os entrevistados mais utilizam para procurar serviços para os seus pets. 48% dos entrevistados fazem uma busca física por pet shops e clínicas veterinárias pelo próprio bairro onde moram, já 36% utilizam sites de pesquisa e 16% utilizam redes sociais.

A última pergunta do questionário (qualitativa) foi para que os entrevistados dissertassem sobre as funcionalidades que os aplicativos que eles utilizam não suprem ou que dessem ideias de funcionalidades que aplicativos poderiam oferecer para que aqueles que não utilizam aplicativo (a pesquisa nos mostrou que foi a maioria) possam vir a utilizar. Muitos usuários sugeriram controle de saúde (clínicas, medicamentos, etc) e a busca por serviços próximos.

Com a entrevista devidamente respondida e as respostas analisadas, foram tiradas algumas conclusões que auxiliaram a qualificar o desenvolvimento, deixando

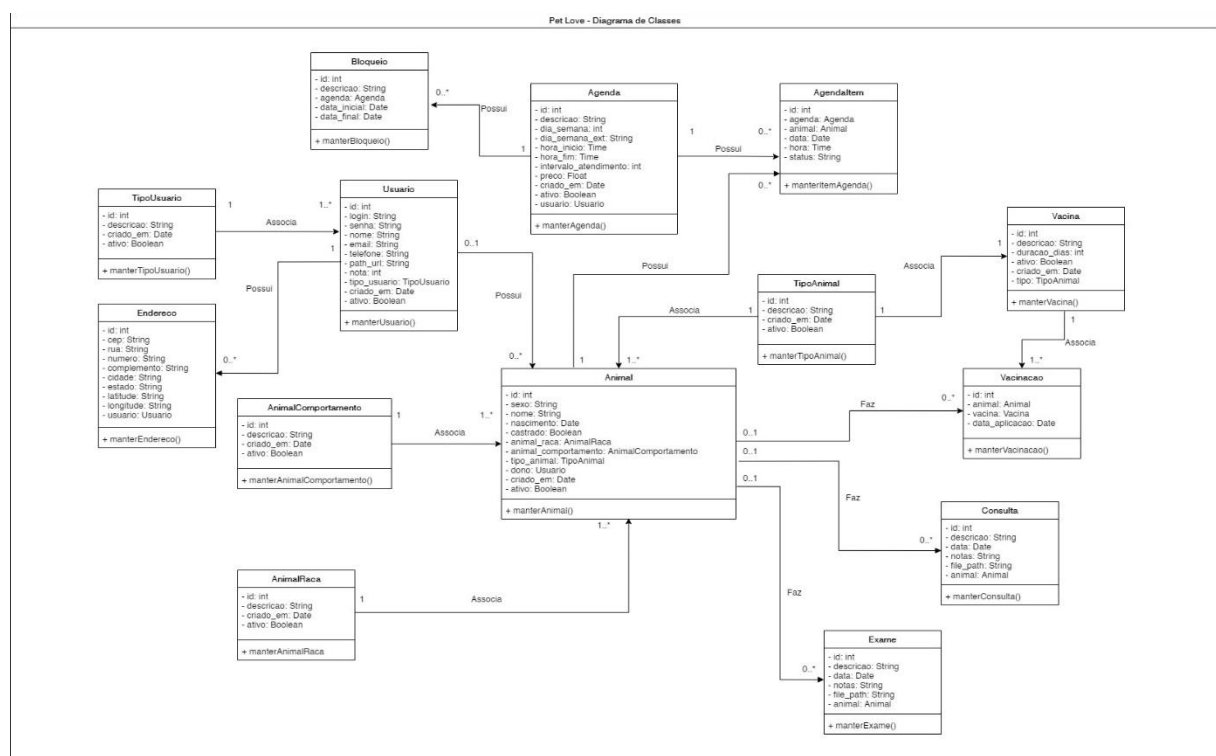


a responsabilidade de cada serviço mais clara. A ideia inicial era oferecer um aplicativo onde os clientes, além de carregarem consigo as informações dos seus pets e buscarem serviços próximos, também pudessem marcar consultas. Como a maioria dos entrevistados não utiliza aplicativo, a funcionalidade de marcação de consultas foi retirada da versão 1.0 da solução. Além disso, o aplicativo mobile será desenvolvido exclusivamente para o cliente. Já a versão web será destinada somente às empresas que desejam se associar e associar seus serviços à solução.

## 6.2 Diagramas UML e Modelo Lógico

Para um melhor entendimento da estrutura da aplicação, foram desenvolvidos dois diagramas UML: um diagrama de classes e um diagrama de casos de uso.

Figura 3: Diagrama de classes geral

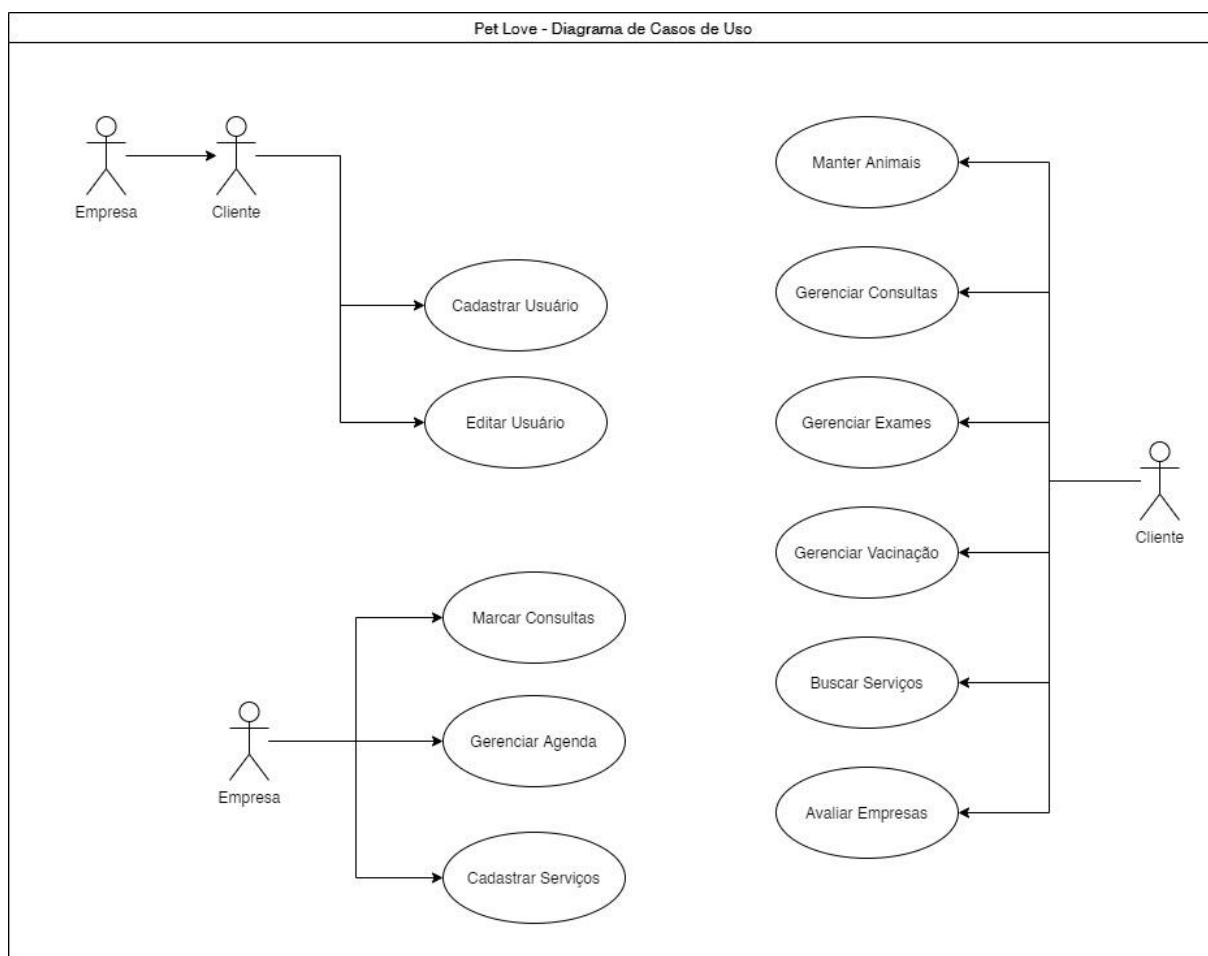


Fonte: elaboração do autor (2021).

No diagrama de classes temos uma noção da estrutura de entidades da aplicação e dos seus respectivos relacionamentos. As principais entidades do sistema são as entidades Animal, Usuario e Agenda. As outras entidades são entidades de apoio ou entidades que servem para armazenar itens das entidades principais. A entidade

Usuario irá comportar tanto clientes quanto empresas e a entidade TipoUsuario definirá o acesso de cada um deles. A entidade Usuario do tipo empresa terá uma ou mais agendas associada a ela, já a entidade Usuario do tipo cliente terá um ou mais animais associados a ela. A entidade Animal terá entidades de Vacinacao, Exames e Consultas associadas a ela para armazenar os serviços utilizados.

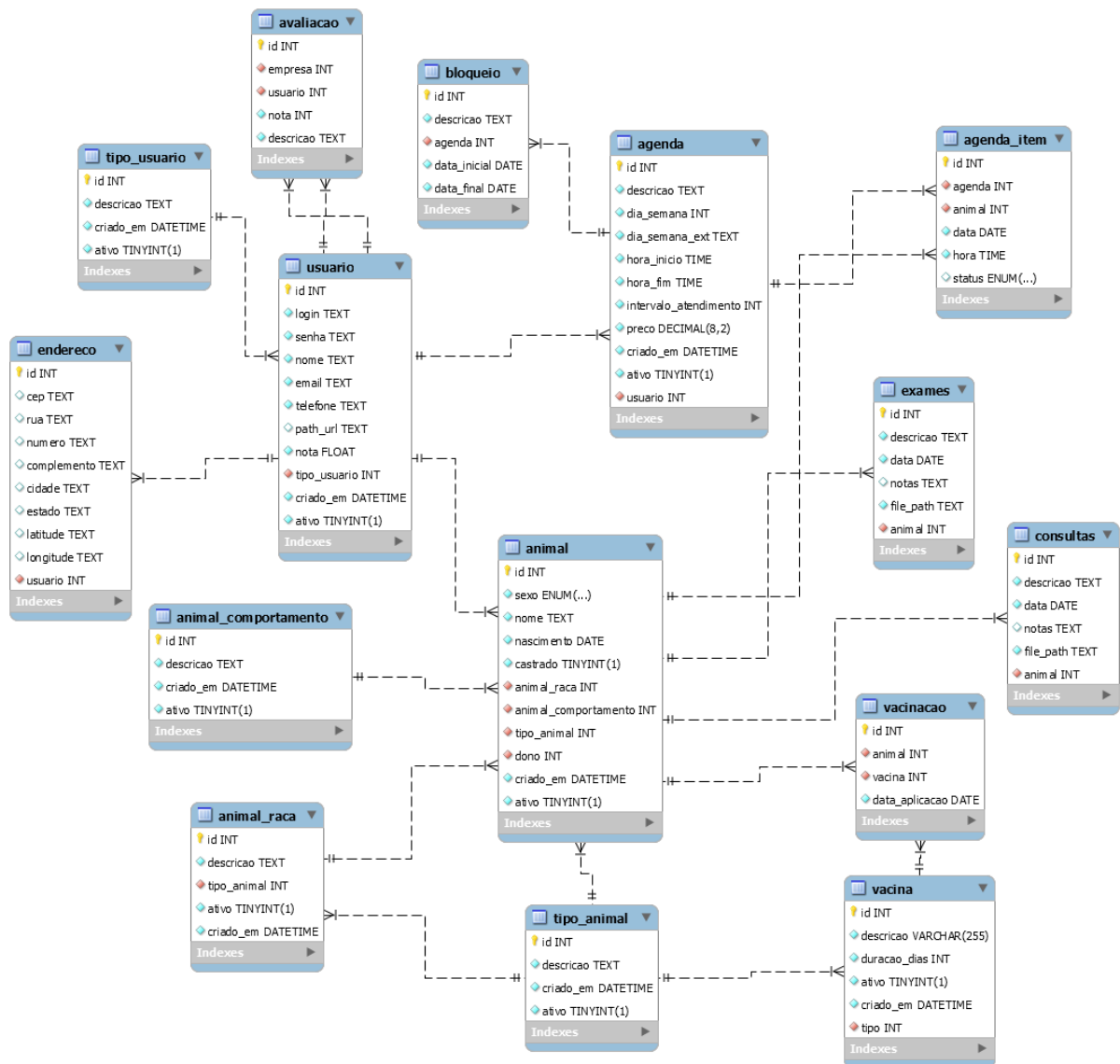
Figura 4: Diagrama de casos de uso geral



Fonte: elaboração do autor (2021).

O diagrama de casos de uso nos traz uma ideia geral de responsabilidades do sistema. Neste caso, o cliente poderá gerenciar seus pets, buscar serviços e terá a responsabilidade de manter os arquivos relacionados à saúde do seu pet que são vacinação, consultas e exames, respectivamente. Já a empresa poderá gerenciar sua agenda de forma geral e marcar consultas.

Figura 5: Modelo lógico



Fonte: elaboração do autor (2021).

O modelo lógico nos traz uma percepção semelhante ao diagrama de classes. Nele, temos todas as entidades, bem como os relacionamentos entre elas. O detalhe aqui é que temos acesso às colunas, bem como suas chaves primárias e estrangeiras. Isso nos traz ainda mais clareza da estrutura de dados que a aplicação terá.

### 6.3 Requisitos do sistema

Para um melhor planejamento e uma chance de assertividade maior, foram definidos requisitos funcionais e não funcionais para a aplicação. Requisitos

funcionais são aqueles sem as quais o propósito básico da aplicação não seria possível. Por exemplo, vamos considerar um sistema que gerencia um mercado de bairro. Uma das atividades básicas que este sistema precisa fazer é armazenar produtos no banco de dados. Um sistema que gerencia um mercado e não armazena produtos perde a sua utilidade. Armazenar produtos é um requisito funcional deste sistema. Já requisitos não funcionais são requisitos indiretos às funcionalidades básicas do sistema, geralmente são requisitos técnicos. No caso de um software de mercado, exigir que ele fosse desenvolvido em PHP, por exemplo, é um requisito não funcional, pois a linguagem de programação não influenciará diretamente no resultado do produto.

#### 6.3.1 Requisitos funcionais

RF01 - Manter Usuários	
RF0101	Cadastrar usuário
RF0102	Atualizar usuário
RF0103	Listar usuários
RF0104	Excluir usuário
RF0105	Manter tabela de apoio para os tipos de usuários

RF02 - Manter Animais	
RF0201	Cadastrar animal
RF0202	Atualizar animal
RF0203	Listar animal
RF0204	Excluir animal
RF0205	Manter tabela de apoio para os tipos de animais
RF0206	Manter tabela de apoio para raça de animais
RF0207	Manter tabela de apoio para comportamento de animais

RF03 - Agendamento	
RF0301	Cadastrar agenda
RF0302	Atualizar agenda
RF0303	Listar agendas por usuário (empresa)
RF0304	Excluir agenda
RF0305	Manter tabela de bloqueio para agenda
RF0306	Manter tabela de itens (consultas) para cada agenda

RF04 - Vacinação	
RF0401	Cadastrar vacinação
RF0402	Atualizar vacinação
RF0403	Listar vacinações por animais de usuários
RF0404	Excluir vacinação
RF0405	Manter tabela de apoio para cadastrar as vacinas

RF05 - Exames	
RF0501	Cadastrar exames
RF0502	Editar exames
RF0503	Listar exames por animais de usuários
RF0504	Excluir exames

RF06 - Consultas	
RF0601	Cadastrar consultas
RF0602	Editar consultas
RF0603	Listar consultas por animais de usuários
RF0604	Excluir consultas

RF07 - Exames	
RF0701	Cadastrar exames
RF0702	Editar exames
RF0703	Listar exames por animais de usuários
RF0704	Excluir exames

RF08 - Sistema	
RF0801	Cliente deve poder manter os seus pets
RF0802	Cliente deve poder gerenciar a vacinação dos seus pets
RF0803	Cliente deve poder gerenciar os exames dos seus pets
RF0804	Cliente deve poder gerenciar consultas dos seus pets
RF0805	Cliente deve poder procurar serviços para os seus pets
RF0806	Cliente deve poder avaliar empresas
RF0807	Empresa deve poder gerenciar suas agendas

### 6.3.2 Requisitos não funcionais

RNF01	API deve suportar autenticação JWT
RNF02	Deve haver responsividade em todo o layout da aplicação
RNF03	Sistema deve enviar avisos aos clientes informando sobre serviços como a data de vacinação

## REFERÊNCIAS

ABINPET. **Dados de Mercado**. 2021. Disponível em:< <http://abinpet.org.br/mercado>>. Acesso em: 28 ago. 2021.

Folha de São Paulo. **Faturamento do mercado pet no país aumenta 13,5% em 2020**. 2021. Disponível em:< <https://www1.folha.uol.com.br/mpme/2021/03/faturamento-do-mercado-pet-no-pais-aumenta-135-em-2020.shtml>>. Acesso em: 28 ago. 2021.