



UNifeob
| ESCOLA DE NEGÓCIOS



2023

**PROJETO DE CONSULTORIA
EMPRESARIAL**



UNIFEOB
CENTRO UNIVERSITÁRIO DA FUNDAÇÃO DE ENSINO
OCTÁVIO BASTOS
ESCOLA DE NEGÓCIOS
ANÁLISE E DESENVOLVIMENTO DE SISTEMAS
GESTÃO DE TECNOLOGIA DA INFORMAÇÃO

PROJETO INTEGRADO
ANIMAL SPHERE SYSTEM

SÃO JOÃO DA BOA VISTA, SP

JUNHO 2023

UNIFEOB
CENTRO UNIVERSITÁRIO DA FUNDAÇÃO DE ENSINO
OCTÁVIO BASTOS
ESCOLA DE NEGÓCIOS
ANÁLISE E DESENVOLVIMENTO DE SISTEMAS
GESTÃO DE TECNOLOGIA DA INFORMAÇÃO

PROJETO INTEGRADO
ANIMAL SPHERE SYSTEM

MÓDULO DESENVOLVIMENTO DESKTOP

Ferramentas de Desenvolvimento Web – Prof. Nivaldo de Andrade

Sistemas Operacionais – Prof. Rodrigo Marudi de Oliveira

Projeto de Desenvolvimento Web – Prof. Nivaldo de Andrade

Estudante:

Mara Vicentina Pinto, RA 1012023100321

SÃO JOÃO DA BOA VISTA, SP
JUNHO, 2023

SUMÁRIO

1 INTRODUÇÃO	4
2 DESCRIÇÃO DA EMPRESA	5
3 PROJETO DE CONSULTORIA EMPRESARIAL	6
3.1 FERRAMENTAS PARA DESENVOLVIMENTO WEB	6
3.1.1 PROJETANDO A FERRAMENTA	6
3.1.2 PROJETO DE INTERFACE COM O USUÁRIO	7
3.1.3 LINGUAGEM DE DESENVOLVIMENTO	7
3.2 SISTEMAS OPERACIONAIS	21
3.2.1 COMPONENTES DE SISTEMAS OPERACIONAIS	21
3.2.2 GERENCIAMENTO E FUNCIONALIDADES DO SISTEMA OPERACIONAL	22
3.2.3 GERENCIAMENTO DE HARDWARE PELO SISTEMA OPERACIONAL	22
4 CONCLUSÃO	23
4.1 ESTUDANTE NA PRÁTICA – LINK DO VÍDEO	23
5 REFERÊNCIAS	24

1 INTRODUÇÃO

No mundo cada vez mais digital em que vivemos, ter uma presença online é essencial para empresas e indivíduos. O desenvolvimento web desempenha um papel fundamental na criação de websites e aplicações online, permitindo que eles sejam funcionais, atrativos e interativos.

O intuito deste projeto é desenvolver uma ferramenta que possa ser executada na Web, por meio de diversos sistemas operacionais, para uma clínica veterinária de pequeno porte, contemplando as atividades básicas de uma clínica, como gestão de clientes, animais e fluxo de caixa.

A implementação do sistema web desenvolvido para a clínica veterinária trará uma série de benefícios ao automatizar processos manuais e otimizar tarefas rotineiras. Com essa solução, espera-se uma melhoria significativa na eficiência operacional, reduzindo o tempo gasto em atividades administrativas.

Além disso, o sistema proporcionará um maior controle na gestão das contas da clínica, possibilitando um acompanhamento mais preciso das finanças. Com isso, a expectativa é de uma gestão mais eficiente do negócio, resultando em maior satisfação dos clientes e melhores resultados para a clínica veterinária.

2 DESCRIÇÃO DA EMPRESA

A empresa contatada foi a clínica veterinária com o nome social “MODA4PATAS” CNPJ 26.699.162/0001-53 que está localizada na cidade de São Paulo, na rua Dr. Plinio Barreto, Apt 181, bloco B e bairro Bela Vista.

Nos encontros realizados com a cliente em questão, relatou-se que o controle das informações dos clientes e animais que atendem são por meio manual e algumas vezes realiza-se o uso de uma planilha, assim como o controle das finanças.

Após análise detalhada do caso, entramos em contato com os clientes para apresentar uma proposta de um sistema web organizado e de fácil acesso. Com essa solução, buscamos resolver os problemas enfrentados anteriormente, oferecendo uma gestão mais eficiente e precisa.

O sistema permitirá automatizar processos, reduzindo o tempo gasto em tarefas rotineiras e garantindo maior controle sobre as finanças, permitindo dessa forma que o cadastro, a edição, a listagem e a deleção se torne mais eficiente e rápida.

A cliente expressou o desejo de que as receitas e os agendamentos dos animais fossem incorporados ao sistema, a fim de aumentar a eficiência do seu trabalho. Dessa forma, devido ao curto prazo de realização do projeto, decidiu-se considerar as funcionalidades principais no momento e essa implementação foi vista como uma melhoria futura para o sistema em questão.

3 PROJETO DE CONSULTORIA EMPRESARIAL

Nesta etapa do PI serão apresentados os conteúdos que cada unidade de estudo utilizou para realizar o projeto, assim como a forma que foram aplicados na empresa escolhida para a realização do projeto.

3.1 FERRAMENTAS PARA DESENVOLVIMENTO WEB

Conceituação dos recursos de desenvolvimento web utilizados para a construção da ferramenta web.

3.1.1 PROJETANDO A FERRAMENTA

Uma aplicação web é um software que é acessado por meio de um navegador da web. São executadas em servidores web e podem ser acessadas por usuários em qualquer dispositivo com acesso à internet. Essas aplicações são desenvolvidas usando tecnologias como HTML, CSS e JavaScript, e fornecem funcionalidades interativas e dinâmicas aos usuários.

As principais fases de desenvolvimento de uma aplicação web podem variar dependendo da metodologia adotada. Contudo, de forma geral, podemos considerar as seguintes etapas:

1. Análise de requisitos: Ocorre a compreensão das necessidades do cliente e a definição dos requisitos funcionais e não funcionais da aplicação web.
2. Design e prototipagem: Criação de um projeto visual da aplicação, design de interface, fluxos de navegação e protótipos interativos.
3. Desenvolvimento do back-end: Construção da lógica de negócio da aplicação, como o desenvolvimento dos componentes de servidor, banco de dados e integrações.
4. Desenvolvimento do front-end: Criação das interfaces de usuário da aplicação web e implementação da interação e da experiência do usuário.
5. Testes e depuração: Realização de testes para garantir o pleno funcionamento da aplicação. Identificação e correção de erros.
6. Implantação: Preparação da aplicação web para ser colocada em produção, configuração do ambiente de hospedagem e realização da migração de dados, se necessário.
7. Manutenção e evolução: Realização de atividades de manutenção, como correção de bugs, atualizações de segurança e melhorias contínuas com base no feedback dos usuários.

Os principais erros e problemas em uma aplicação web incluem falhas de validação e entrada de dados, lentidão e desempenho ruim devido ao tempo de carregamento e baixo desempenho em momentos de tráfego intenso, questões de segurança, incompatibilidade entre navegadores e conflitos de integração.

Além disso, um gerenciamento inadequado de erros e a falta de escalabilidade podem impactar negativamente a experiência do usuário. Desse modo, é fundamental a realização de testes abrangentes, a implementação de medidas de segurança adequadas e a manutenção da aplicação.

3.1.2 PROJETO DE INTERFACE COM O USUÁRIO

Os estudantes devem introduzir conceitos de Interação Humano-Computador (IHC) em ambientes Web, avaliar os princípios e as boas práticas de IHC.

A Interação Humano-Computador (IHC) em ambientes web, busca melhorar a experiência do usuário durante a interação com aplicações online. Isso é alcançado através de princípios e boas práticas específicas, como usabilidade, design centrado no usuário, acessibilidade, feedback e consistência, eficiência e desempenho, design responsivo e testes de usabilidade.

Esses conceitos visam criar interfaces intuitivas, fáceis de usar e eficientes, levando em consideração as necessidades dos usuários, proporcionando uma experiência satisfatória e acessível em diferentes dispositivos e contextos. A aplicação desses princípios e boas práticas contribui para uma interação mais efetiva e agradável entre seres humanos e computadores em ambientes web.

3.1.3 LINGUAGEM DE DESENVOLVIMENTO

No desenvolvimento do sistema web *Animal Sphere* utilizou-se a Linguagem de Marcação de Hipertexto HTML 5, Folhas de Estilo em Cascata CSS, JavaScript, PHP juntamente com o framework Laravel. Além disso, também utilizou-se o framework Bootstrap.

O Laravel foi escolhido por ser um dos frameworks PHP mais utilizados no mercado, sendo baseado na arquitetura MVC (Model-View-Controller, ou Modelo-Visão-Controle, em português). MVC é um padrão de arquitetura de software focado em reuso de código, no qual ocorre a divisão da estrutura lógica de um sistema em 3 camadas: a do Modelo, relacionada ao banco de dados; a de Visão, vinculada a visualização dos dados e das páginas; e a do

Controle, responsável pela conexão e transmissão de informações entre as camadas Modelo e Visão.

O projeto também possui versionamento de código git e para mais informações sobre os códigos desenvolvidos está disponível no repositório que se encontra no link: <https://github.com/mara-vicentina/animal-sphere.git>.

Para instalação do projeto e configurações acesse o README.md.

Uma View contém todo o conteúdo que será exibido para o usuário, o qual fará a interação com a mesma. No momento em que o usuário realiza uma interação com a view, essa por sua vez faz o processo de requisição para a Controller, que solicita para a Model os dados.

A partir dessa solicitação a Model faz a requisição para consulta no Banco de Dados e faz o retorno com os dados para a Controller que posteriormente repassa para a View que renderiza o conteúdo.

A seguir temos imagens dos códigos que exemplificam esse funcionamento na prática.

```
You, 19 seconds ago | 1 author (You)
<!DOCTYPE html>
<html lang="pt-br">

  @include('head')

  <body>| You, 2 days ago • alterando modais de cadastro e listagem e criação...
    <div class="container-fluid g-0 overflow-hidden">
      <div class="row g-0">
        <div class="col-md-4 col-lg-3 col-xs-4 col-sm-4 custom-height">
          @include('dashboard/sidebar/sidebar')
        </div>
        <div class="col-md-8 col-lg-9 col-xs-8 col-sm-8 custom-height">
          <div class="row flex-nowrap g-0 bg-light w-100">
            <div class="col-12">
              <div class="min-vh-100 d-flex flex-column">
                <div class="col-12">
                  <div class="card rounded-0 border-0 shadow-sm">
                    <div class="card-body fs-5 mid-gray">
                      Olá, {{ Auth::user()->name }}.
                    </div>
                  </div>
                </div>
                @yield('page-content')
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
    @include('js-content')
  </body>
</html>
```

Figura 1. Código da View Cliente

Fonte: Autor (2023)

```
You, 16 hours ago | 1 author (You)
@extends('dashboard/template')

@section('page-content')
<div class="card m-5 shadow">
  <div class="row">
    <div class="col-md-10 col-sm-7">
      <div class="card-body fs-5 mid-gray">
        Clientes
      </div>
    </div>
    <div class="col-md-2 col-sm-5">
      <div class="card-body fs-5">
        <button type="button" class="btn btn-primary custom-button" data-bs-toggle="modal" data-bs-target="#clientes">
          <i data-feather="plus" style="width:19px; height:19px;"></i>
          Cadastrar
        </button>
      </div>
    </div>
  </div>
</div>
</div>
```

Figura 1. Código da View Cliente - conteúdo da página

Fonte: Autor (2023)

```
<div class="card mx-5 p-2 shadow overflow-auto custom-card pr-5 py-2">
  <table class="table">
    <thead>
      <tr>
        <th class="col title-head text-left">Nome Completo</th>
        <th class="col title-head text-left">E-mail</th>
        <th class="col title-head text-left">CPF</th>
        <th class="col title-head text-left">Telefone</th>
        <th class="col title-head text-left">Rua</th>
        <th class="col title-head text-left">Número</th>
        <th class="col title-head text-left">Bairro</th>
        <th class="col title-head text-left">Cidade</th>
        <th class="col title-head text-left">Estado</th>
        <th class="col title-head text-left">Ações</th>
      </tr>
    </thead>
```

Figura 1. Código da View Cliente

Fonte: Autor (2023)

```

<tbody>
  @foreach ($clientes as $cliente)
  <tr>
    <td class="col subtitle-list text-left">{{ $cliente->name }}</td>
    <td class="col subtitle-list text-left">{{ $cliente->email }}</td>
    <td class="col subtitle-list text-left">{{ $cliente->cpf }}</td>
    <td class="col subtitle-list text-left">{{ $cliente->phone }}</td>
    <td class="col subtitle-list text-left">{{ $cliente->street }}</td>
    <td class="col subtitle-list text-left">{{ $cliente->number }}</td>
    <td class="col subtitle-list text-left">{{ $cliente->neighborhood }}</td>
    <td class="col subtitle-list text-left">{{ $cliente->city }}</td>
    <td class="col subtitle-list text-left">{{ $cliente->state }}</td>
    <td class="col subtitle-list text-left">
      <div class="row">
        <div class="col-6 p-0">
          <button class="btn btn-link" onclick="editClient('{{ $cliente->id }}', '{{ $cliente->json_data }}')">
            <i data-feather="edit" class="d-inline edit-info"></i>
          </button>
        </div>
        <div class="col-6 p-0">
          <form method="POST" action="{{ url('/client') }}">
            <input type="hidden" name="_token" value="{{ csrf_token() }}">
            <input type="hidden" name="_method" value="DELETE">
            <input type="hidden" name="cliente_id" value="{{ $cliente->id }}">
            <button type="submit" class="btn btn-link">
              <i data-feather="trash-2" class="d-inline delete-info"></i>
            </button>
          </form>
        </div>
      </div>
    </td>
  </tr>
  @endforeach
</tbody>
</table>
</div>

```

Figura 1. Código da View Cliente

Fonte: Autor (2023)

```

@include('dashboard/clientes/cadastro-clientes')
@include('dashboard/clientes/edicao-clientes')
@endsection

```

Figura 1. Código da View Cliente

Fonte: Autor (2023)

```

You, 2 days ago | 1 author (You)
<?php You, 2 days ago • alterando modais de cadastro e

namespace App\Http\Controllers;

use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use Auth;
use App\Models\Cliente;
use App\Models\Animal;
use Redirect;

class ClienteController extends Controller
{
    public function index()
    {
        $clientes = Cliente::all();
        foreach ($clientes as $cliente) {
            $cliente->json_data = $cliente->toJson();
        }

        return view('dashboard/clientes/index', [
            'currentPage' => 'client',
            'clientes' => $clientes,
        ]);
    }
}

```

Figura 1. Código da ClienteController - function index

Fonte: Autor (2023)

```

public function create(Request $request)
{
    $validated = $request->validate([
        'nome_completo' => ['required', 'string'],
        'email' => ['required', 'email'],
        'telefone' => ['required', 'string', 'max:11'],
        'cpf' => ['required', 'string', 'max:11'],
        'cep' => ['required', 'string', 'max:8'],
        'rua' => ['required', 'string'],
        'numero' => ['required', 'numeric', 'max_digits:10'],
        'complemento' => ['required', 'string'],
        'bairro' => ['required', 'string'],
        'cidade' => ['required', 'string'],
        'estado' => ['required', 'string'],
    ]);

    $cliente = new Cliente();
    $cliente->name = $request->nome_completo;
    $cliente->email = $request->email;
    $cliente->phone = $request->telefone;
    $cliente->cpf = $request->cpf;
    $cliente->cep = $request->cep;
    $cliente->street = $request->rua;
    $cliente->number = $request->numero;
    $cliente->complement = $request->complemento;
    $cliente->neighborhood = $request->bairro;
    $cliente->city = $request->cidade;
    $cliente->state = $request->estado;
    $cliente->user_id = Auth::user()->id;
    $cliente->save();

    return Redirect::to('/dashboard/clientes');
}

```

Figura 1. Código da ClienteController - function create

Fonte: Autor (2023)

```

public function update(Request $request)
{
    $validated = $request->validate([
        'nome_completo' => ['required', 'string'],
        'email' => ['required', 'email'],
        'telefone' => ['required', 'string', 'max:11'],
        'cpf' => ['required', 'string', 'max:11'],
        'cep' => ['required', 'string', 'max:8'],
        'rua' => ['required', 'string'],
        'numero' => ['required', 'numeric', 'max_digits:10'],
        'complemento' => ['required', 'string'],
        'bairro' => ['required', 'string'],
        'cidade' => ['required', 'string'],
        'estado' => ['required', 'string'],
    ]);

    $cliente = Cliente::where('id', $request->client_id)->first();
    $cliente->name = $request->nome_completo;
    $cliente->email = $request->email;
    $cliente->phone = $request->telefone;
    $cliente->cpf = $request->cpf;
    $cliente->cep = $request->cep;
    $cliente->street = $request->rua;
    $cliente->number = $request->numero;
    $cliente->complement = $request->complemento;
    $cliente->neighborhood = $request->bairro;
    $cliente->city = $request->cidade;
    $cliente->state = $request->estado;
    $cliente->user_id = Auth::user()->id;
    $cliente->save();

    return Redirect::to('/dashboard/clientes');
}

```

Figura 1. Código da ClienteController - function update

Fonte: Autor (2023)

```

public function remove(Request $request)
{
    Animal::where('cliente_id', $request->cliente_id)->delete();
    Cliente::where('id', $request->cliente_id)->delete();
    return Redirect::to('/dashboard/clientes');
}
}

```

Figura 1. Código da ClienteController - function remove

Fonte: Autor (2023)

```

You, 2 days ago | 1 author (You)
<?php You, 2 weeks ago • inicio do projeto - tela de cad

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Cliente extends Model
{
    use HasFactory;
}

```

Figura 1. Código da Model Cliente

Fonte: Autor (2023)

Os procedimentos realizados no desenvolvimento foram:

1. Criação de uma single page application, com intuito de mostrar quais as funcionalidades existentes no sistema, criar uma seção para contato e botões para cadastro do usuário e login do mesmo na plataforma.

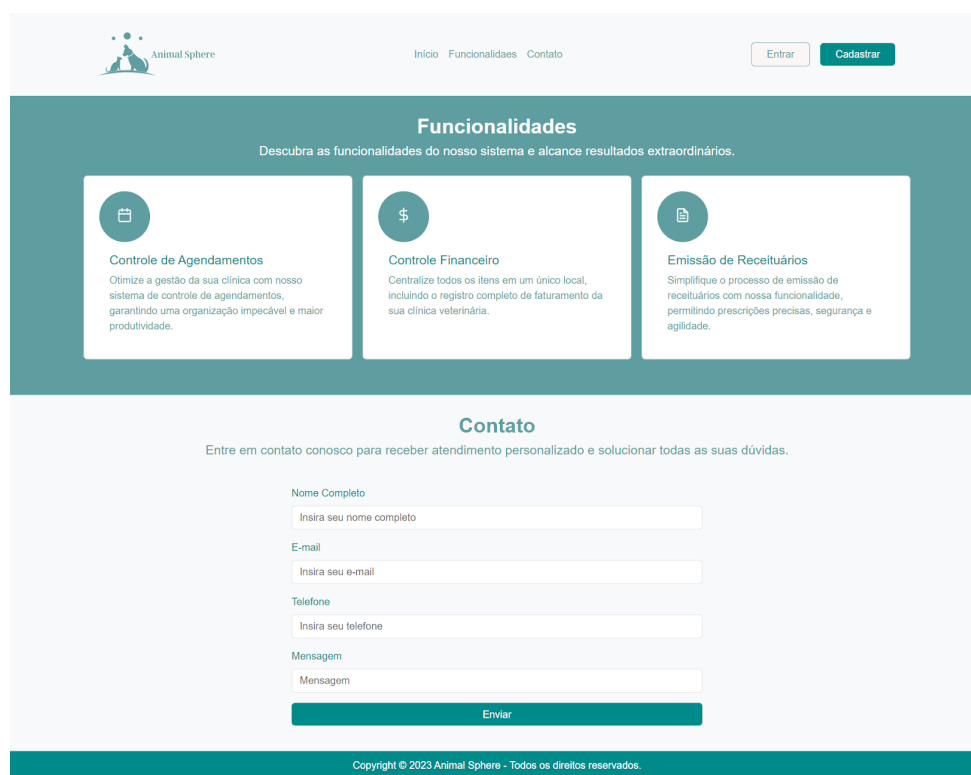


Figura 1. Single Page Application Animal Sphere

Fonte: Autor (2023)

2. Criação das modais de cadastro e login, a modal de cadastro permite que o usuário se cadastre no sistema, para que posteriormente consiga fazer seu login na modal correspondente.



Figura 1. Modal de Login

Fonte: Autor (2023)

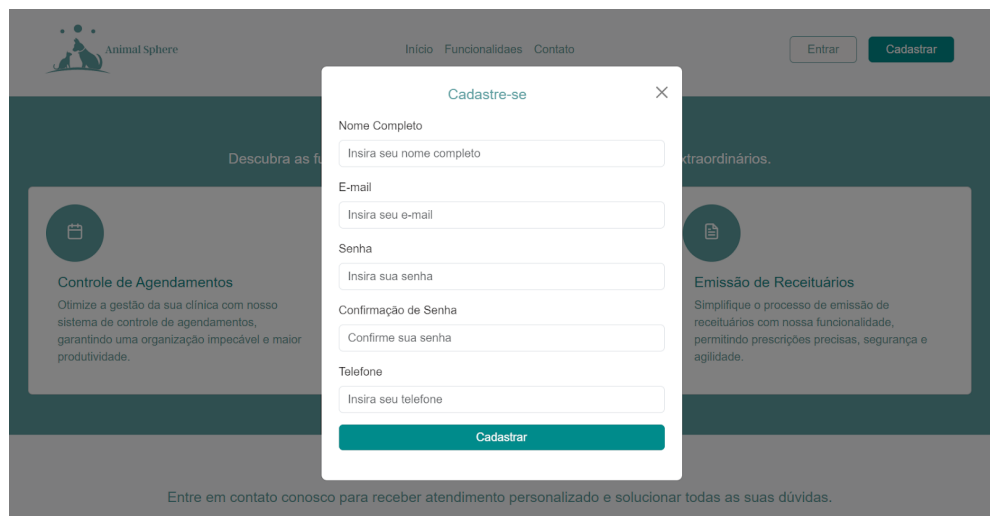


Figura 1. Modal de Cadastro

Fonte: Autor (2023)

3. Realização do menu lateral no qual o usuário navega entre as páginas de clientes, animais, fluxo de caixa, receituário, agendamentos e faz o logout da conta.

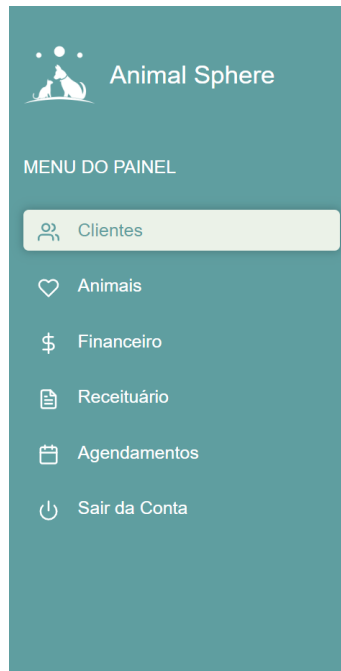


Figura 1. Menu Lateral

Fonte: Autor (2023)

4. Criação das páginas que são chamadas pelo menu, com botão para a modal de cadastro das páginas correspondentes, listagens, edições e deleções dos dados cadastrados.

Olá, Mara Vicentina Pinto.

Cientes + Cadastrar

Nome Completo	E-mail	CPF	Telefone	Rua	Número	Bairro	Cidade	Estado	Ações
João	joao@gmail.com	07527122000	19866563254	Rua Amália Santos	22	Centro	São João da Boa Vista	São Paulo	
Bruno	bruno@gmail.com	07527122001	19866568563	Rua São Jorge	562	Filomena	São João da Boa Vista	São Paulo	
Isabel	isabel@gmail.com	07527122501	19866763989	Rua Constância	96	Peru	São João da Boa Vista	São Paulo	
Alex	alex@gmail.com	07527122052	19866563252	Rua Florença	77	Brasil	São João da Boa Vista	São Paulo	
Lorenzo	lore@gmail.com	07527421254	19866563254	Rua José Maria	89	Hibisco	São João da Boa Vista	São Paulo	
Maria	mary@gmail.com42	07527122785	19868554523	Rua Gomes Almeida	52	Jardim da Flores	São João da Boa Vista	São Paulo	

Figura 1. Página de Clientes

Fonte: Autor (2023)

Olá, Mara Vicentina Pinto.

Animais + Cadastrar









Nome Completo	Espécie	Raça	Sexo	Animal Castrado	Tutor	Data de Nascimento	Ações
Phyo	Felina	Angorá	Macho	Não	Alex	13/01/2022	 
Sophya	Felina	Angorá	Fêmea	Sim	João	27/01/2023	 
Pipoca	Canina	Yorkshire Terrier	Fêmea	Sim	João	14/07/2022	 
Max	Canina	Poodle	Macho	Não	Maya	11/06/2021	 

Figura 1. Página de Animais

Fonte: Autor (2023)

Olá, Mara Vicentina Pinto.

Fluxo de Caixa da Clínica + Cadastrar













Nome	Tipo	Valor	Data	Ações
Conta de Água	Saída	300.00	01/06/2023	 
Consulta Max	Entrada	650.00	02/06/2023	 
Conta de Energia	Saída	290.00	01/06/2023	 
Consulta Phyo	Entrada	300.00	06/06/2023	 
Consulta Sophya	Entrada	900.00	10/06/2023	 
Compra de Equipamentos	Saída	1200.00	12/06/2023	 
Consulta Pipoca	Entrada	500.00	13/06/2023	
Entradas: 2350 reais		Saídas: 1790 reais		Total em Caixa: 560 reais

Figura 1. Página de Fluxo de Caixa

Fonte: Autor (2023)

Olá, Mara Vicentina Pinto.



Figura 1. Página de Receituário

Fonte: Autor (2023)

Olá, Mara Vicentina Pinto.



Figura 1. Página de Agendamentos

Fonte: Autor (2023)

5. Criação das modais de cadastro de clientes, animais e registro de fluxo de caixa.

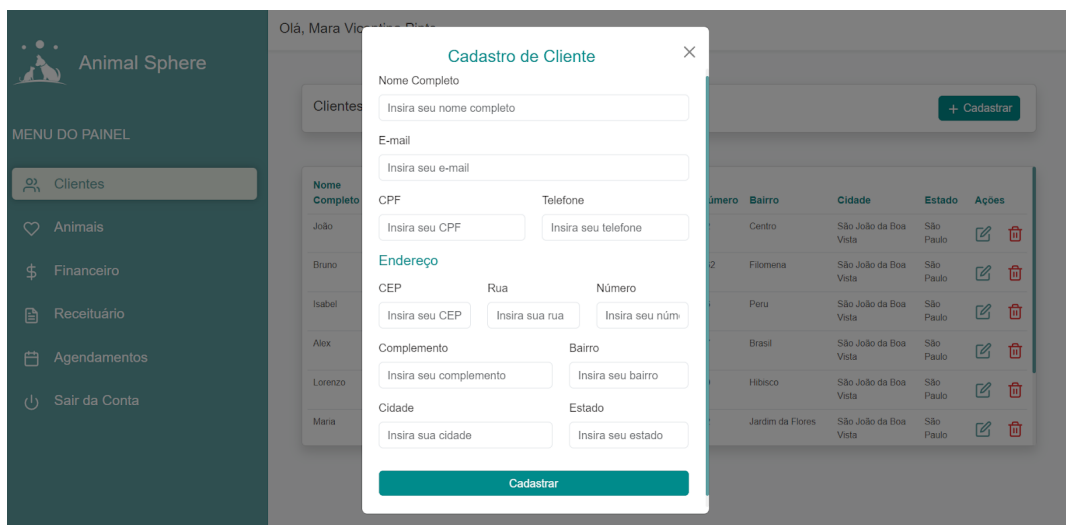


Figura 1. Modal Cadastro de Cliente

Fonte: Autor (2023)

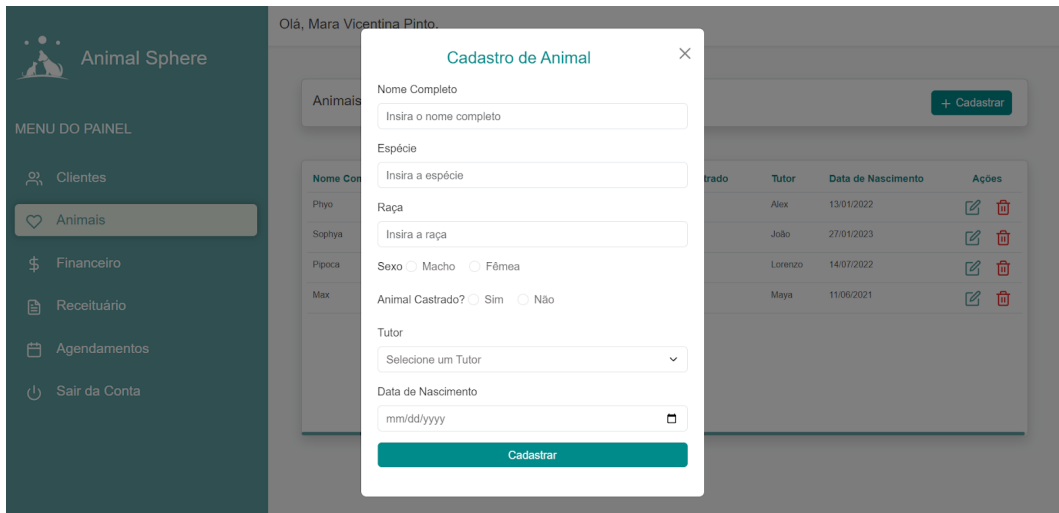


Figura 1. Modal Cadastro de Animal

Fonte: Autor (2023)

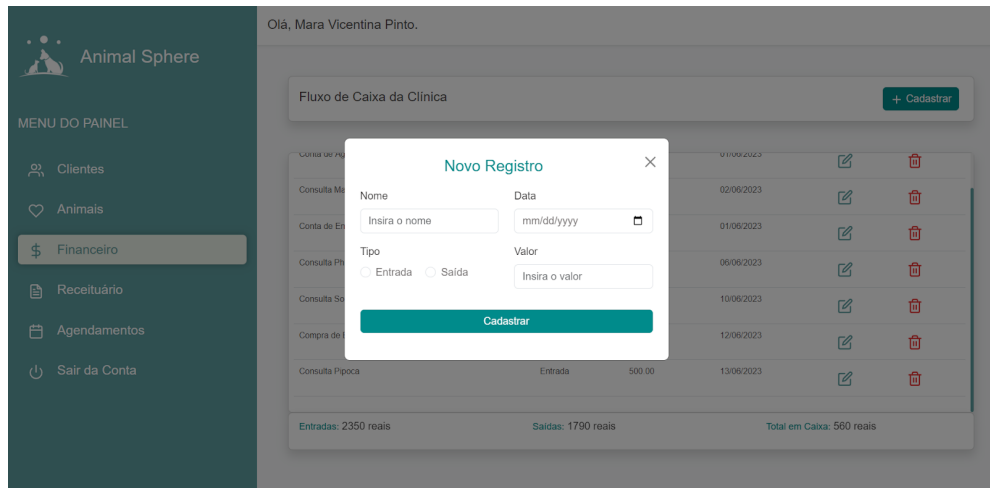


Figura 1. Modal Novo Registro

Fonte: Autor (2023)

5. Criação das modais de edição de clientes, animais e registro de fluxo de caixa.

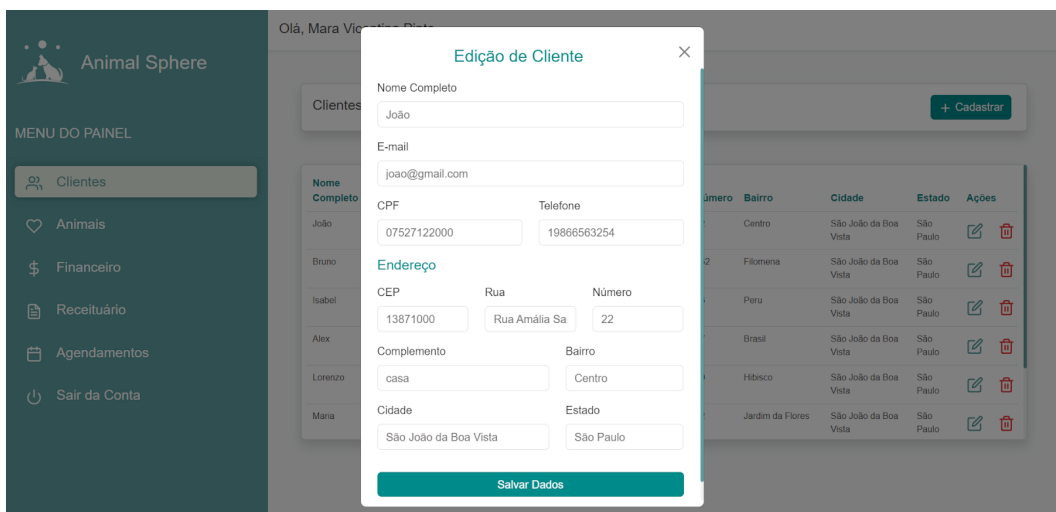


Figura 1. Modal de Edição de Cliente

Fonte: Autor (2023)

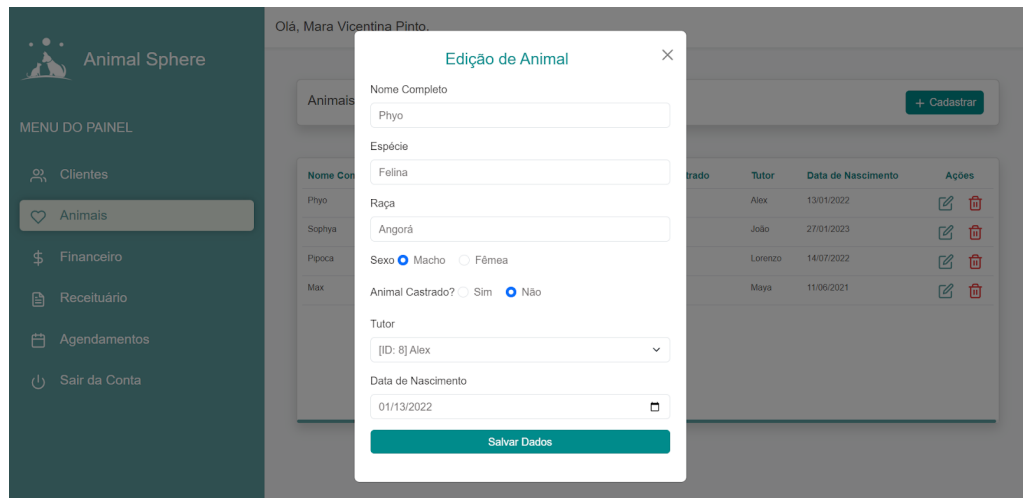


Figura 1. Modal de Edição de Animal

Fonte: Autor (2023)

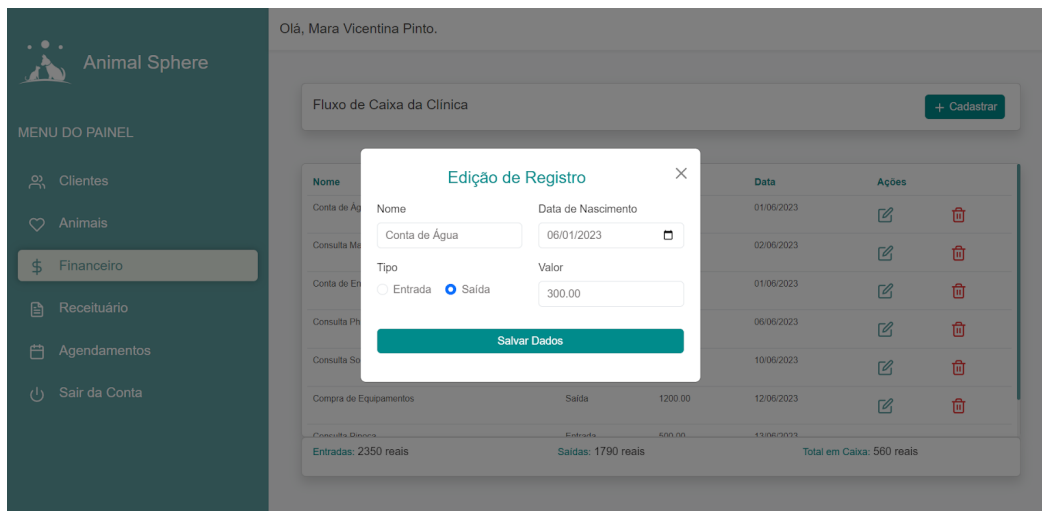


Figura 1. Modal de Edição de Registro

Fonte: Autor (2023)

Após a criação visual do sistema, deu-se início a criação do banco de dados MySQL. Para uma visualização de forma clara e precisa das informações que serão armazenadas e como elas se relacionam entre si no banco de dados, optou-se pela criação do Diagrama Entidade-Relacionamento (DER) que segue abaixo.

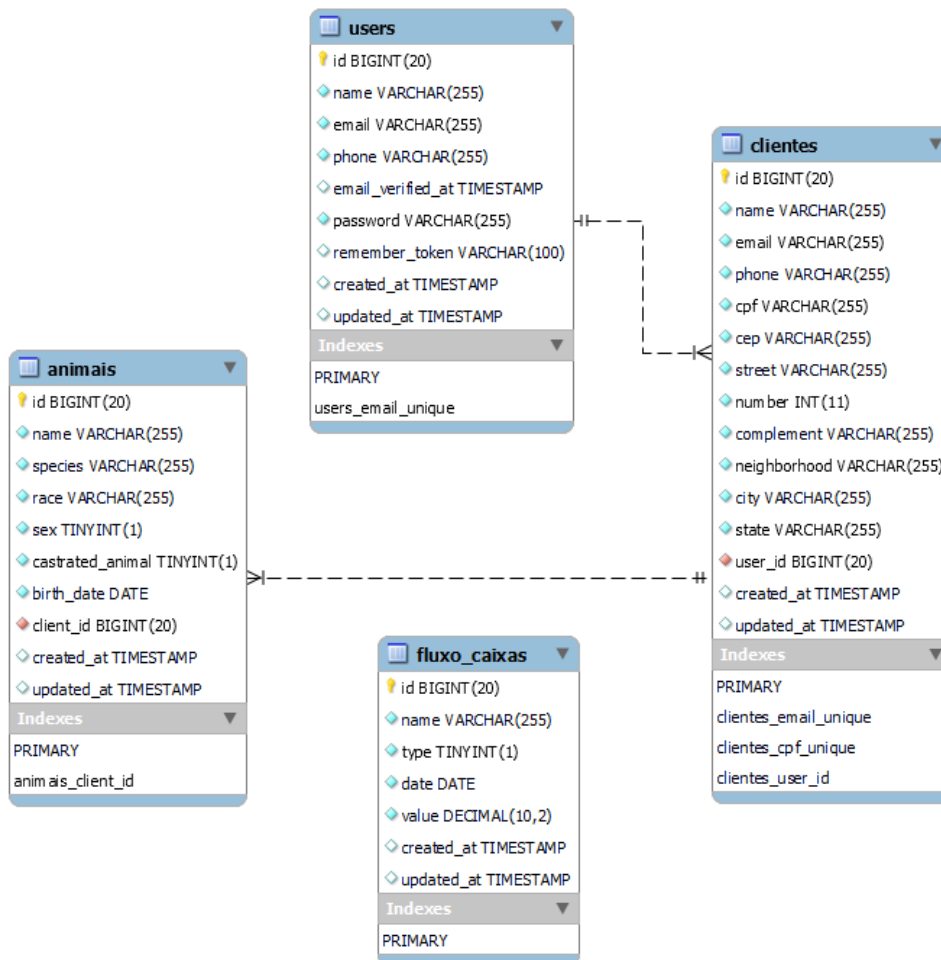


Figura 2. Diagrama Entidade Relacionamento

Fonte: Autor (2023)

3.2 SISTEMAS OPERACIONAIS

Buscou-se abranger todos os sistemas operacionais por meio da utilização do servidor HTTP, assim o projeto se torna mais viável e sem muitas dependências.

3.2.1 COMPONENTES DE SISTEMAS OPERACIONAIS

O projeto foi desenvolvido utilizando o servidor HTTP, devido a esse fato, não depende de um sistema operacional específico. No caso em questão o tipo do servidor utilizado foi o HTTP Apache.

O uso do Apache como servidor HTTP facilita a hospedagem e o acesso à aplicação em diferentes ambientes operacionais. Sua escolha foi feita levando em consideração sua

ampla adoção e estabilidade comprovada no desenvolvimento do projeto. Esse servidor fornece a infraestrutura necessária para o funcionamento da aplicação web, garantindo sua disponibilidade.

3.2.2 GERENCIAMENTO E FUNCIONALIDADES DO SISTEMA OPERACIONAL

Não houve a necessidade de escolher um sistema operacional específico, devido a escolha de um servidor HTTP. Todos os sistemas operacionais atendem o projeto desenvolvido.

3.2.3 GERENCIAMENTO DE HARDWARE PELO SISTEMA OPERACIONAL

O sistema animal sphere funciona localmente, sendo acessado por meio de um navegador, na rede local. Ele funciona instalado em servidores locais, permitindo o acesso e processamento de dados internamente, sem depender de uma conexão com a internet.

As vantagens incluem maior controle e segurança dos dados, velocidade de acesso e desempenho aprimorados, além da capacidade de personalização para atender às necessidades específicas da organização. No entanto, as desvantagens são a limitação de acesso remoto, a necessidade de manutenção de servidores locais, custos de infraestrutura e menor flexibilidade comparado a sistemas web baseados na nuvem.

O banco de dados funciona localmente, armazenando as informações, operando no computador local. Ele é instalado e executado na máquina local, permitindo o acesso aos dados sem a necessidade de uma conexão à internet.

As vantagens de um banco de dados local incluem maior desempenho e velocidade de acesso aos dados, maior controle sobre a segurança e privacidade dos dados, além de não depender da disponibilidade de uma conexão com a internet.

No entanto, as desvantagens são a necessidade de manutenção e backup regular dos dados, menor escalabilidade em comparação com bancos de dados distribuídos e a dificuldade de compartilhar informações entre diferentes sistemas e dispositivos sem uma conexão de rede.

Contudo pensando na escalabilidade do projeto, vê-se a necessidade de futuramente realizar a hospedagem do sistema e do banco de dados na nuvem, onde as informações serão armazenadas em um servidor remoto acessível pela internet.

As vantagens incluem fácil acesso aos dados de qualquer lugar e dispositivo, escalabilidade sob demanda, menor necessidade de manutenção e backup de dados, além da possibilidade de integração com outros serviços na nuvem. Atualizações automáticas, alta disponibilidade e menor necessidade de manutenção de infraestrutura local.

No entanto, as desvantagens podem incluir preocupações com segurança e privacidade dos dados, dependência de uma conexão estável com a internet e possível custo adicional conforme o uso e armazenamento de dados aumentam.

4 CONCLUSÃO

Este trabalho mostrou a importância contínua dos sistemas web na atualidade e como eles ainda são fundamentais para muitas empresas e em especial as de pequeno porte.

Além disso, o trabalho apresentou a aplicação prática de um sistema web para uma clínica veterinária de pequeno porte, demonstrando como a tecnologia pode ajudar a otimizar processos rotineiros e aumentar a eficiência dos negócios, nos atendimentos e no gerenciamento financeiro.

Em resumo, a evolução tecnológica nos últimos anos tem trazido diversas mudanças para o mundo dos negócios, e os sistemas web permitem a automação de tarefas, o armazenamento de dados em nuvem e a integração de processos em uma ampla gama de setores. Isso resulta em maior eficiência operacional, redução de erros e economia de tempo e recursos.

4.1 ESTUDANTE NA PRÁTICA – LINK DO VÍDEO

Nesta etapa foi produzido um vídeo no formato de pitch, onde o grupo apresenta o projeto a ser utilizado pelo cliente alvo. Segue o link do vídeo:

PROJETO INTEGRADO - ADS - 2023 - Animal Sphere System -
<https://youtu.be/IH7gBrd9G8E>

5 REFERÊNCIAS

- BENTO, E. Desenvolvimento web com PHP e MySQL. [S.l.]: Casa do Código, 2014.
- DEITEL, H.M. Sistemas Operacionais. 3ª.ed. Tradução de Arlete Simille Marques. São Paulo: Pearson Prentice Hall, 2005.
- ENGINES, D. DB-Engines Ranking - popularity ranking of database management systems. 2019. Disponível em: <https://db-engines.com/en/ranking> . Acesso em: 01 de junho de 2023.
- GAMMA, E. Padrões de Projetos: Soluções Reutilizáveis. [S.l.]: Bookman, 2006.
- MACHADO, Francis Berenger. Arquitetura de Sistemas Operacionais. Rio de Janeiro: Editora LTC, 2004.
- MARCORATTI, J. C. O processo de Software. 2014. Disponível em: https://www.macoratti.net/proc_sw1.htm . Acesso em: 10 de junho de 2023.
- NIEDERAUER, J. Desenvolvendo Websites com PHP. [S.l.]: Novatec, 2011.
- OTWELL, T. The PHP Framework for Web Artisans. 2019. Disponível em: <https://laravel.com/> . Acesso em: 02 de junho de 2023.
- SILBERSCHATZ, A. & GAGNE, G. & GALVIN, P. B. Fundamentos de Sistemas Operacionais. Tradução de Adriana Cashin Rieche. Rio de Janeiro, 2004.
- TANENBAUM, A.S. Sistemas Operacionais Modernos. 2ª.ed. Tradução de Ronaldo A. L Gonçalves. São Paulo, 2009.