

UNIVERSIDADE VIRTUAL DO ESTADO DE SÃO PAULO

André Paganotti Faber, RA: 2101905.

Claudio Santos Cuimar, RA: 2102202.

Jonathan Peres de Souza, RA: 2105831.

Lucas Cortibeli Ferreira, RA: 2106548.

Matheus Geraldini de Araújo, RA: 2103079.

Paloma Ap. Bonatte de Almeida, RA: 2101600.

Ricardo Eduardo Claro, RA: 2101134.

Simone Sabino Egas de Oliveira, RA: 2103419.

Cadê meu Pet? - Uma ferramenta facilitadora para localização de animais perdidos.

Vídeo de apresentação do Projeto Integrador

<https://www.youtube.com/watch?v=ho78vh2s27M>

UNIVERSIDADE VIRTUAL DO ESTADO DE SÃO PAULO

Cadê meu Pet? - Uma ferramenta facilitadora para localização de animais perdidos.

Relatório Técnico-Científico apresentado na disciplina de Projeto Integrador para o curso de Ciências de Dados, Engenharia da Computação e Tecnologia da informação da Universidade Virtual do Estado de São Paulo (UNIVESP).

André Paganotti ; FABER, Claudio Santos; CUIVAR, Jonathan Peres de; SOUZA, Lucas Cortibeli; FERREIRA, Matheus Geraldini de; ARAÚJO, Paloma Ap. Bonatte de; ALMEIDA, Ricardo Eduardo; CLARO, Simone Sabino Egas de; OLIVEIRA . **Cadê Meu Pet**. 00f. Relatório Técnico-Científico. Tecnologia da Informação, Ciências de Dados e Engenharia da Computação – **Universidade Virtual do Estado de São Paulo**. Tutor: Rafaela Molina de Angelo. Polo: Cordeirópolis, Pirassununga, Rio Claro e Saltinho, 2023.

RESUMO

O projeto em questão envolve o desenvolvimento de uma plataforma web destinada a simplificar a busca por animais perdidos e resgatados, abordando uma questão relevante na sociedade brasileira. Inicialmente concebido no Projeto Integrador I, empregou-se o framework Django e o banco de dados MySQL para criar a base da plataforma, permitindo o registro de animais desaparecidos e a busca por animais encontrados. No entanto, no Projeto Integrador II, identificamos a oportunidade de elevar a plataforma a um novo patamar, introduzindo JavaScript para melhorar a interatividade e funcionalidade do usuário. Essa adição possibilitará uma experiência mais dinâmica e eficiente, tornando a busca e o registro de animais mais acessíveis para a população. Além do impacto prático, o projeto possui relevância acadêmica, demonstrando a aplicação de conhecimentos em tecnologia da informação e programação para resolver um problema social. A plataforma representa uma valiosa ferramenta para a sociedade e as instituições envolvidas na busca por animais perdidos, destacando-se como um exemplo de inovação e aprimoramento contínuo no campo da tecnologia. Combinando esforços práticos e acadêmicos, o projeto busca contribuir significativamente para a solução de um problema frequente e fortalecer o papel da tecnologia como agente de mudança positiva na sociedade brasileira.

PALAVRAS-CHAVE: Plataforma Web; Animais Perdidos; JavaScript; Tecnologia da Informação; Solução Social.

SUMÁRIO

1 INTRODUÇÃO	5
2 DESENVOLVIMENTO.....	7
2.1 Objetivos.....	7
2.2 Justificativa e delimitação do problema	8
2.3 Fundamentação teórica	9
2.4 Aplicação das disciplinas estudadas no projeto integrador	11
2.5 Metodologia.....	12
3 RESULTADOS: SOLUÇÃO FINAL	15
4 CONSIDERAÇÕES FINAIS.....	32
REFERÊNCIAS	34

1 INTRODUÇÃO

O aumento do número de animais nas ruas constitui um desafio em constante crescimento em várias localidades ao redor do globo. No contexto brasileiro, de acordo com informações fornecidas pelo Instituto Pet Brasil no ano de 2021, aproximadamente 185 mil animais são objeto de resgate por organizações não governamentais, sendo que 96% desses são cães, enquanto os gatos representam 4% do total (Instituto Pet Brasil, 2021). É lamentável observar que muitos desses animais acabam por se extraviar ou são alvo de furto por parte de seus tutores, o que pode acarretar considerável angústia tanto para as famílias quanto para os próprios animais em questão. Com o propósito de contribuir para a localização desses animais, optou-se pela concepção de uma plataforma cujo escopo consiste na simplificação do processo de identificação e rastreamento, tendo em vista a sua reunião com seus tutores.

Neste trabalho, será empregado o framework Django, uma ferramenta de destacada eficácia para a criação de aplicações web mediante a utilização da linguagem de programação Python. Adicionalmente, o sistema gerenciador de banco de dados MySQL será adotado, desempenhando o papel crucial de armazenar os registros relacionados aos animais resgatados e seus respectivos tutores. A aplicação de Git para o controle de versões do projeto viabilizará a colaboração simultânea de múltiplos indivíduos no mesmo projeto. Tais recursos revelam-se fundamentais no contexto de exploração dos requisitos subjacentes ao tema orientador estabelecido pela UNIVESP.

Na continuação deste projeto, que se estende do PI I para o PI II, planeja-se incorporar a linguagem JavaScript à plataforma. Essa adição se justifica pela necessidade de proporcionar uma experiência mais interativa e dinâmica aos usuários da plataforma. A integração do JavaScript permitirá a implementação de recursos como validação de formulários em tempo real, exibição de informações de forma mais dinâmica, e interações mais ágeis entre o usuário e a aplicação.

A inclusão do JavaScript no desenvolvimento da plataforma aprimorará significativamente a usabilidade e a eficiência, tornando-a mais amigável e intuitiva para os tutores de animais perdidos e para aqueles que desejam registrar animais resgatados. Esse aprimoramento na interatividade contribuirá para a eficácia global da plataforma, reforçando seu potencial para auxiliar na localização e reunificação de animais perdidos com seus tutores,

bem como na documentação de animais resgatados, consolidando, assim, seu papel na redução do número de animais abandonados nas ruas.

O escopo do presente estudo consiste na elaboração de uma página de pesquisa que possibilitará que os tutores de animais perdidos efetuem o registro de informações pertinentes a seus animais de estimação. Tais informações incluirão dados como imagens, características físicas, endereço de residência, número de telefone para contato, bem como outros detalhes relevantes. Além disso, a plataforma viabilizará o registro de informações sobre animais resgatados, contemplando informações como a espécie, raça, coloração, idade, local onde foram encontrados, entre outros dados de interesse.

Antecipa-se que a plataforma concebida possa configurar-se como uma ferramenta de suma importância no contexto da busca por animais extraviados. Tal iniciativa tem o potencial de possibilitar aos tutores a recuperação de seus animais de maneira mais célere e eficaz, ao mesmo tempo em que contribui para a diminuição do número de animais abandonados nas vias públicas.

2 DESENVOLVIMENTO

2.1 OBJETIVOS

Este estudo se concentra no desenvolvimento de um sistema web destinado a auxiliar na localização de animais perdidos. Em termos gerais, o objetivo é criar uma plataforma que permita aos usuários cadastrar informações sobre animais desaparecidos e realizar buscas por animais encontrados. O propósito subjacente é contribuir para a resolução de um problema comum na sociedade, fornecendo uma ferramenta útil tanto para a população em geral quanto para as instituições envolvidas na busca por animais extraviados.

Para atingir esse objetivo, foram estabelecidos objetivos específicos como desenvolver uma interface de usuário amigável e intuitiva para o registro de informações sobre animais perdidos e encontrados, utilizar o sistema gerenciador de banco de dados MySQL para armazenar as informações dos animais cadastrados, implementar um sistema de busca que permita a localização de animais perdidos com base em critérios como raça, porte e localização utilizar o framework Django no desenvolvimento da plataforma, garantindo uma arquitetura escalável e segura, empregar o sistema de versionamento Git para controlar as alterações realizadas no código-fonte da plataforma, facilitando a colaboração entre os desenvolvedores e mantendo um registro histórico das mudanças.

Esses objetivos específicos foram delineados com o intuito de garantir a eficácia do projeto, atendendo às necessidades dos usuários e fazendo uso de tecnologias modernas e robustas. Além disso, é importante mencionar que nesta nova versão do projeto, será incorporada a linguagem JavaScript para melhorar a interatividade com os usuários da plataforma.

2.2 JUSTIFICATIVA E DELIMITAÇÃO DO PROBLEMA

Conforme explicitado na introdução deste relatório, a questão concernente ao aumento da população de animais abandonados nas vias públicas representa uma problemática comum na sociedade. Este problema afeta não somente os proprietários de animais, mas também a comunidade em geral. A perda de um animal de estimação causa angústia aos seus responsáveis, e o aumento da presença desses animais nas ruas é considerado um desafio de ordem social. Isso se deve ao potencial de ataques a pessoas e à disseminação de doenças que podem ser transmitidas à população.

Nesse contexto, a concepção de uma plataforma web destinada a facilitar a busca por esses animais assume uma relevância significativa. Além de contribuir para o bem-estar social ao simplificar a localização de animais perdidos por seus tutores, a plataforma também desempenhará um papel cultural importante. Isso se deve ao fato de que os dados serão fornecidos pela própria comunidade, incentivando a interação entre os cidadãos em busca de seus animais e promovendo o fortalecimento dos laços sociais no âmbito da plataforma.

Vale destacar que, por se tratar de uma plataforma on-line, seu alcance é potencialmente amplo, podendo ser adotada em diferentes bairros ou cidades, ampliando assim seu impacto e utilidade.

Por fim, é pertinente ressaltar a relevância acadêmica do projeto, uma vez que envolve a aplicação de conhecimentos em tecnologia da informação e programação para o desenvolvimento de uma plataforma web. Além disso, a plataforma pode servir como objeto de estudo para pesquisas relacionadas à tecnologia e à proteção animal, ampliando seu valor além do contexto imediato de auxílio na busca por animais perdidos.

2.3 FUNDAMENTAÇÃO TEÓRICA

A proposta deste projeto tem como objetivo a união de usuários que buscam por algo específico com usuários que oferecem esse determinado objeto, seguindo o modelo de uma loja virtual. Essa abordagem ganha relevância adicional ao considerarmos o crescente acesso à internet nos domicílios brasileiros, o qual aumentou de 71% para 83% (G1, 2021). A elevada penetração da internet no país destaca sua capacidade como um meio altamente propício para o desenvolvimento deste projeto.

Outro aspecto digno de nota é o expressivo crescimento do comércio online, como evidenciado no relatório "Webshoppers 43ª edição" da Ebit & Nielsen. Em 2020, o faturamento do comércio eletrônico atingiu a marca de R\$ 87,4 milhões, representando o maior crescimento em 13 anos no setor (Ebit & Nielsen, Webshoppers 43ª edição, 2021). Esses dados demonstram o impacto significativo das transações efetuadas pela internet e o potencial desta plataforma para simplificar a realização de diversas atividades cotidianas, incluindo a busca por animais perdidos.

Para fundamentar esta proposta, no Projeto Integrador I, realizou-se uma pesquisa com o intuito de avaliar as dificuldades enfrentadas na busca por animais perdidos. A pesquisa contou com a participação de 15 indivíduos que responderam a um formulário disponibilizado por meio da ferramenta "Formulário do Google". Os resultados obtidos revelaram que 60% dos entrevistados já tiveram a necessidade de realizar buscas por animais perdidos e 53,3% afirmaram ter resgatado algum animal e precisado empreender esforços para localizar o seu respectivo proprietário. Foi perguntado também, qual o meio utilizado para realizar tais buscas, sendo 86,7% das respostas "Redes Sociais". Sendo que, apenas 26,7% dos entrevistados, alegaram estimar o sucesso deste meio de busca acima dos 75% de chance de êxito.

No âmbito do Projeto Integrador II, foi realizado uma visita a uma instalação que se autodenomina uma Organização Não Governamental (ONG), embora não possua registro oficial. Esta entidade está situada na cidade de Rio Claro, especificamente no bairro Bom Sucesso, caracterizado como uma área periférica. A residência da Sra. Ana Silva abriga essa iniciativa, a qual é composta por uma moradia modesta com um espaçoso quintal e um canil improvisado, porém notável pela sua higiene e organização. Durante nossa visita, fomos cordialmente recebidos pela Sra. Ana Silva e pelo Sr. João Oliveira, um líder comunitário do

bairro que colabora na angariação de recursos financeiros e alimentos para sustentar essa iniciativa.

No momento da visita, o local abrigava 40 cães e 10 gatos, muitos dos quais residiam ali por longos períodos. Segundo os responsáveis, a população local frequentemente encontra animais perdidos nas ruas e os leva até a residência da Sra. Ana Silva. Entretanto, devido aos custos operacionais do local, que alcançam a cifra de R\$4.000,00 mensais, englobando despesas com alimentação, medicamentos e produtos de limpeza, a Sra. Ana cessou a admissão de novos animais. No entanto, isso resultou em uma prática prejudicial, na qual as pessoas começaram a abandonar animais em frente à residência, cientes de que a Sra. Ana não recusaria assistência a um animal necessitado.

De acordo com os entrevistados, a criação de um site destinado ao cadastro desses animais seria uma solução ideal. O bairro em que a ONG está situada é afastado e, portanto, as pessoas raramente buscam por animais perdidos localmente. Além disso, os anúncios em redes sociais muitas vezes não alcançam toda a comunidade. A criação de uma página web permitiria o registro desses animais, agregando todas as informações sobre animais perdidos nas ONGs espalhadas pela cidade e facilitando a promoção de ações de adoção.

Portanto, concluímos que embora as pessoas recorram às redes sociais para buscar animais perdidos, essa abordagem não é eficiente o suficiente. Com base nessa evidência, esperamos que uma ferramenta online específica para a busca por animais perdidos possa suprir essa deficiência.

2.4 APLICAÇÃO DAS DISCIPLINAS ESTUDADAS NO PROJETO INTEGRADOR

Durante o desenvolvimento do projeto, diversas disciplinas foram empregadas, as quais foram abordadas pela Univesp. As disciplinas Algoritmos e Programação de Computadores I e II, introduzidas no 3º e 4º Bimestres, foram consideradas essenciais para o projeto, pois proporcionaram a introdução aos conceitos de algoritmos, linguagens de programação, controle de versão (GIT) e criação de interfaces gráficas. Fundamentos de Internet e Web, apresentada no 4º Bimestre, foi amplamente utilizada, especialmente nas matérias relacionadas aos principais componentes da Web, linguagem HTML, CSS e Protocolo HTTP, bem como na criação de páginas web e aplicação de estilos seletivos aos elementos HTML.

Gestão de Operação e Desenvolvimento de Produtos, apresentada no 5º Bimestre, contribuiu para o processo de prototipagem de ideias, no qual foram ouvidas ideias dos clientes e do grupo, discutidas as melhores e feito o protótipo do projeto, apresentando ao cliente e, após a aprovação, produzindo o projeto definitivo.

Formação do Profissional em Computação, vista no 5º Bimestre, foi empregada principalmente nas matérias de Projeto de desenvolvimento de Software, algoritmo e implementação, padrões importantes da web, Html, Python, Bibliotecas e Frameworks e ciências de dados, introduzindo os conceitos necessários para o desenvolvimento web.

Programação Orientada a Objetos, também apresentada no 5º Bimestre, foi útil para o paradigma orientado a objetos e para o uso de frameworks de desenvolvimento.

A disciplina Banco de Dados, vista no 7º Bimestre, proporcionou uma introdução à linguagem SQL, indexação e ferramentas de manipulação de bancos de dados, além de consultas usando SQL, contribuindo com os conceitos de bancos de dados, sistemas de gerenciamento de bancos de dados e consulta de dados.

Por fim, no 9º Bimestre, a disciplina de Engenharia de Software desempenhou um papel fundamental ao abordar questões relacionadas à engenharia de requisitos, arquitetura e projetos de software, estratégias e técnicas de testes. Isso contribuiu significativamente para a elaboração bem-sucedida do projeto.

2.5 METODOLOGIA

Conforme Prodanov (2006), a metodologia é descrita como um conjunto de regras que orientam a aquisição de conhecimento e é definida como um elemento facilitador no processo de pesquisa e aprendizagem.

O desenvolvimento do projeto aderiu aos princípios do Design Thinking (DT), seguindo as fases de ouvir, prototipar e implementar. Para embasar essa abordagem, utilizamos o livro "Para compreender o Design Thinking", de autoria de Cleyson de Moraes Mello, José Rogério Moura de Almeida Neto e Regina Pentagna Petrillo, que foi apresentado durante a disciplina de Gestão da Inovação e Desenvolvimento de Produtos. Essa obra nos proporcionou valiosos insights sobre a aplicação do Design Thinking no contexto do nosso projeto.

Na etapa de "Ouvir e Interpretar o Contexto", foram conduzidas atividades de empatia e compreensão das necessidades e emoções do público-alvo, que consiste nos tutores de animais perdidos. A equipe buscou compreender as dificuldades enfrentadas por esse grupo e identificar lacunas nos métodos existentes de busca de animais perdidos. Importante mencionar que alguns membros da equipe já haviam vivenciado a experiência de buscar por seus próprios animais de estimação, o que proporcionou uma compreensão mais profunda das emoções, frustrações e necessidades desse público. Conforme descrito na fundamentação teórica deste relatório a equipe realizou uma entrevista, em uma ONG local na cidade de Rio Claro e por meio de um questionário elaborado na plataforma "Google Forms". Essa escolha se baseou na afirmação dos autores Lakatos e Marconi (2007), que argumentam que os questionários de pesquisa possibilitam obter informações mais precisas dos participantes, incluindo detalhes que podem não estar disponíveis em manuais ou documentos. O questionário da entrevista foi divulgado por meio das mídias sociais dos autores, envolvendo um total de 15 participantes. O objetivo da entrevista foi compreender as principais dificuldades enfrentadas pelos tutores de animais perdidos e identificar lacunas nos métodos existentes de busca.

Para compreender os resultados, foi empregada uma técnica de análise quantitativa, uma vez que, de acordo com Vergara (2005), dados tratados quantitativamente são analisados usando métodos estatísticos. Após a análise desses dados, chegou-se à conclusão de que a ferramenta deveria ser disponibilizada online, e o cadastro de animais deveria incluir

informações como raça, cor, nome e características gerais, tais como manchas, cicatrizes, porte e raio de busca a ser considerado.

Na fase de criação e prototipagem, com base nas conclusões derivadas da análise da entrevista, a equipe realizou sessões de brainstorming para gerar ideias inovadoras sobre como a plataforma poderia resolver o problema e, ao mesmo tempo, atender aos requisitos do tema norteador: "Desenvolver um software com framework web que utilize noções de banco de dados, praticando controle de versão".

Para atender à proposta do tema norteador e facilitar a busca por animais perdidos, a equipe concluiu que a estrutura do projeto deveria incluir um banco de dados com informações sobre animais perdidos, que seria acessado online, atendendo a duas das exigências do tema norteador. A terceira exigência, que é a prática de controle de versão, não apresentou dificuldades na realização.

O projeto idealizado consistiu em uma página web, semelhante a um catálogo de compras, onde os "anúncios" seriam os animais perdidos, cadastrados por seus tutores, e os animais encontrados, cadastrados por pessoas ou ONGs que avistaram ou resgataram animais das ruas.

Com base nas ideias geradas, a equipe desenvolveu um protótipo da plataforma. Esse protótipo permitiu testar e validar conceitos e funcionalidades. Foi desenvolvido em uma plataforma Linux, e a distribuição escolhida foi o Fedora 38, devido à afinidade com esse sistema operacional e à presença da linguagem Python 3.6 instalada por padrão, como será discutido posteriormente.

Como mencionado anteriormente, o framework escolhido foi o Django, devido à praticidade que oferece para o desenvolvimento e manutenção de páginas web. Ele cria todas as APIs necessárias com apenas alguns comandos no console do computador, requerendo apenas pequenas edições. Além disso, a linguagem de programação utilizada em suas APIs é o Python, que também foi abordado na UNIVESP, na disciplina de Algoritmos e Programação de Computadores I.

O desenvolvimento começou com a criação de um aplicativo no framework, denominado "cade_meu_pet". Em seguida, foram realizadas configurações no arquivo "settings.py", onde o aplicativo recém-criado foi definido no campo "INSTALLED_APPS". Além disso, as configurações do banco de dados foram especificadas, com o MySQL sendo escolhido como SGBD. Por fim, o diretório de mídia da plataforma foi definido, uma vez que a página seria alimentada com fotos dos animais.

Com as configurações do arquivo "settings.py" concluídas, a equipe prosseguiu com a criação dos modelos. Para isso, acessou-se o arquivo "models.py", onde foram definidas duas classes: "AnimalPerdido" e "AnimalEncontrado". O propósito dessas classes era modelar os tipos de objetos a serem cadastrados no banco de dados, representando, respectivamente, os animais perdidos e os animais encontrados. Devido à natureza diferente desses objetos, o banco de dados registraria suas informações em tabelas distintas, nomeadas "cade_meu_pet_AnimalPerdido" e "cade_meu_pet_AnimalEncontrado".

3 RESULTADOS: SOLUÇÃO FINAL

Diante das exigências do tema norteador, optou-se pelo uso do framework Django em conjunto com o banco de dados MySQL. Essas ferramentas combinadas proporcionam uma página web na qual os usuários podem cadastrar e consultar informações sobre os animais na nuvem.

A integração da API `mysqlclient` foi necessária para estabelecer a comunicação entre o Django, que é baseado na linguagem Python, e o banco de dados MySQL, que utiliza a linguagem SQL. Além disso, foram empregados o JavaScript e suas bibliotecas para implementar as funcionalidades da página, melhorando a acessibilidade por meio de feedback sonoro ao interagir com os botões. O controle de versão foi realizado utilizando a plataforma Git.

O funcionamento da ferramenta assemelha-se a um catálogo de compras, no qual os "anúncios" correspondem aos animais encontrados, perdidos ou disponíveis para adoção.

Com base nas ideias concebidas, desenvolveu-se um protótipo da plataforma, possibilitando a validação de conceitos e funcionalidades, conforme será detalhado posteriormente.

O desenvolvimento do projeto ocorreu em uma plataforma Linux, sendo a distribuição selecionada o Fedora 38, devido à familiaridade com esse sistema operacional e à conveniência de já possuir o Python 3.6 instalado por padrão. Como mencionado anteriormente, o framework escolhido foi o Django, devido à sua praticidade no desenvolvimento e manutenção de páginas web, bem como à facilidade de implementação das APIs necessárias. Além disso, o Django foi apresentado durante a disciplina de Formação Profissional em Computação. É importante ressaltar que a linguagem de programação utilizada pelo Django é o Python, que também foi abordado no contexto da UNIVESP, especificamente na disciplina de Algoritmos e Programação de Computadores I.

O processo de desenvolvimento teve início com a criação de um aplicativo no framework, denominado `"cade_meu_pet"`. Em seguida, procedeu-se às configurações no arquivo `"settings.py"`. Nesse arquivo, o aplicativo recém-criado foi definido na seção `"INSTALLED_APPS"` (Figura 1). Além disso, foi configurada a seção `"DATABASES"`, na qual o MySQL foi estabelecido como o Sistema de Gerenciamento de Banco de Dados (SGBD),

e a API mysqlclient foi instalada (Figura 2). Por fim, o diretório de mídia da plataforma foi especificado, uma vez que a página seria alimentada com fotos dos animais.

```
36  INSTALLED_APPS = [  
37      'django.contrib.admin',  
38      'django.contrib.auth',  
39      'django.contrib.contenttypes',  
40      'django.contrib.sessions',  
41      'django.contrib.messages',  
42      'django.contrib.staticfiles',  
43      'cade_meu_pet',  
44      'allauth',  
45      'allauth.account',  
46      'allauth.socialaccount',  
47  ]
```

Figura 1 - Campo “INSTALLED_APPS” em *settings.py*. Autores, 2023.

```
84  DATABASES = {  
85      'default': {  
86          'ENGINE': 'django.db.backends.mysql',  
87          'NAME': 'animais_bd',  
88          'USER': 'root',  
89          'PASSWORD': '*****',  
90          'HOST': 'localhost',  
91          'PORT': '3306',  
92      }  
93  }
```

Figura 2 - Campo “DATABASES” em *settings.py*. Autores, 2023.

Após a configuração do arquivo "settings.py", procedeu-se à criação dos modelos, acessando o arquivo "models.py". Nesse arquivo, foram criadas três classes: "AnimalPerdido," "AnimalEncontrado," e "AnimalAdocao" (Figura 3). A função dessas classes é modelar o tipo de objeto que será cadastrado no banco de dados. Como esses objetos são distintos em natureza, o banco de dados os registrará em tabelas diferentes.

```
6 class AnimalPerdido(models.Model):
7     nome = models.CharField(max_length=100)
8     especie = models.CharField(max_length=100)
9     raca = models.CharField(max_length=100)
10    cor = models.CharField(max_length=100)
11    descricao = models.TextField()
12    foto = models.ImageField(upload_to='foto_encontrados/', blank=True, null=True)
13    data_perdido = models.DateField()
14    local_perdido = models.CharField(max_length=100)
15    nome_perdeu = models.CharField(max_length=100, default='Desconhecido')
16    tel_perdeu = models.CharField(max_length=100, default='00')
17
18    def __str__(self):
19        return self.nome
20
21 class AnimalEncontrado(models.Model):
22     nome = models.CharField(max_length=100, blank=True, null=True)
23     especie = models.CharField(max_length=100)
24     raca = models.CharField(max_length=100, blank=True, null=True)
25     cor = models.CharField(max_length=100)
26     descricao = models.TextField()
27     foto = models.ImageField(upload_to='foto_encontrados/', blank=True, null=True)
28     data_encontrado = models.DateField()
29     local_encontrado = models.CharField(max_length=100)
30     nome_encontrou = models.CharField(max_length=100, default='Desconhecido')
31     tel_encontrou = models.CharField(max_length=100, default='00')
32
33    def __str__(self):
34        return self.nome
35
36 class AnimalAdocao(models.Model):
37     nome = models.CharField(max_length=100)
38     especie = models.CharField(max_length=100)
39     raca = models.CharField(max_length=100)
40     cor = models.CharField(max_length=100)
41     descricao = models.TextField()
42     foto = models.ImageField(upload_to='foto_adocao/', blank=True, null=True)
43     idade = models.CharField(max_length=2)
44     nome_tutor = models.CharField(max_length=100, default='Desconhecido')
45     tel_tutor = models.CharField(max_length=100, default='00')
46     cidade_tutor = models.CharField(max_length=100)
47
48    def __str__(self):
49        return self.nome
50
```

Figura 3 - Classes em *models.py*. Autores, 2023.

Para que os dados fornecidos pelos usuários fossem registrados no banco de dados, optou-se por utilizar a estrutura de formulários disponível na biblioteca de APIs do Django. A implementação dessa estrutura envolveu a criação do arquivo "forms.py," onde foram criadas mais três classes: "AnimalPerdidoForm," "AnimalEncontradoForm," e "AdocaoForm" (Figura 4). A função dessas classes é criar um objeto para o formulário preenchido pelo usuário. Para que esse objeto possa ser registrado no banco de dados, é necessário que ele respeite a mesma estrutura dos modelos definidos em "models.py," conforme descrito no parágrafo anterior.

```
7 class AnimalPerdidoForm(forms.ModelForm):
8     foto = ImageField(required=False)
9
10     class Meta:
11         model = AnimalPerdido
12         fields = ('nome', 'especie', 'raca', 'cor',
13                 'descricao', 'foto', 'data_perdido',
14                 'local_perdido', 'nome_perdeu', 'tel_perdeu')
15
16 class AnimalEncontradoForm(forms.ModelForm):
17     foto = ImageField(required=False)
18
19     class Meta:
20         model = AnimalEncontrado
21         fields = ('nome', 'especie', 'raca', 'cor',
22                 'descricao', 'foto', 'data_encontrado',
23                 'local_encontrado', 'nome_encontrou', 'tel_encontrou')
24         widgets = {
25             'nome': forms.TextInput(attrs={'required': False}),
26             'raca': forms.TextInput(attrs={'required': False}),
27         }
28
29 class AdocaoForm(forms.ModelForm):
30     foto = ImageField(required=False)
31
32     class Meta:
33         model = AnimalAdocao
34         fields = ('nome', 'especie', 'raca', 'cor', 'descricao',
35                 'foto', 'idade', 'nome_tutor', 'tel_tutor', 'cidade_tutor')
```

Figura 4 - Classes em *forms.py*. Autores, 2023.

O próximo passo foi criar uma forma de disponibilizar as informações cadastradas aos usuários. Com base no conhecimento adquirido na disciplina de Banco de Dados, sabíamos que o MySQL oferece a capacidade de realizar esses retornos, incluindo a aplicação de filtros que facilitam as buscas. Assim, dentro do arquivo "views.py," cuja função é definir e renderizar as funcionalidades da página web, foram criadas funções em Python responsáveis por realizar requisições ao banco de dados e retornar os resultados ao usuário.

Foram definidas as seguintes funções: "perdidos," "encontrados," e "adocao" (Figura 5) para verificar se as informações fornecidas nos formulários são válidas e, se forem, salvá-las no banco de dados. Caso contrário, apresentam um erro ao usuário. Além disso, foram criadas as funções "lista_perdidos," "lista_encontrados," e "lista_adocao" (Figura 6) para exibir as listas de registros feitos, juntamente com opções de filtragem. Também foram desenvolvidas as funções "envio_sucesso" e "form_error" para apresentar ao usuário uma página de sucesso ou uma página de erro após o envio do formulário, respectivamente.

```
78 def adocao(request):
79     if request.method == 'POST':
80         form = AdocaoForm(request.POST, request.FILES)
81         if form.is_valid():
82             try:
83                 animal_adocao = form.save(commit=False)
84                 animal_adocao.save()
85                 return redirect('envio_sucesso')
86             except Exception as e:
87                 print(form.errors) # Imprime os erros do formulário para depuração
88                 return redirect('form_error') # Redireciona para a página de erro
89         else:
90             print(form.errors) # Imprime os erros do formulário para depuração
91             return redirect('form_error') # Redireciona para a página de erro
92     else:
93         form = AdocaoForm()
94         context = {'form': form}
95         return render(request, 'adocao.html', context)
```

Figura 5 - Função "adocao" em *views.py*. Autores, 2023.

```

185 def lista_adocao(request):
186     animais_adocao = AnimalAdocao.objects.all()
187
188     especies = AnimalAdocao.objects.values_list('especie', flat=True).distinct()
189     racas = AnimalAdocao.objects.values_list('raca', flat=True).distinct()
190     cores = AnimalAdocao.objects.values_list('cor', flat=True).distinct()
191     idades = AnimalAdocao.objects.values_list('idade', flat=True).distinct()
192     cidades = AnimalAdocao.objects.values_list('cidade_tutor', flat=True).distinct()
193
194     especie_filtro = request.GET.get('especie')
195     raca_filtro = request.GET.get('raca')
196     cor_filtro = request.GET.get('cor')
197     idade_filtro = request.GET.get('idade')
198     cidade_filtro = request.GET.get('cidade_tutor')
199
200     if especie_filtro:
201         animais_adocao = animais_adocao.filter(especie=especie_filtro)
202     if raca_filtro:
203         animais_adocao = animais_adocao.filter(raca=raca_filtro)
204     if cor_filtro:
205         animais_adocao = animais_adocao.filter(cor=cor_filtro)
206     if idade_filtro:
207         animais_adocao = animais_adocao.filter(idade=idade_filtro)
208     if cidade_filtro:
209         animais_adocao = animais_adocao.filter(cidade_tutor=cidade_filtro)
210
211     context = {
212         'animais_adocao': animais_adocao,
213         'especies': especies,
214         'racas': racas,
215         'cores': cores,
216         'idades': idades,
217         'cidades': cidades,
218         'especie_filtro': especie_filtro,
219         'raca_filtro': raca_filtro,
220         'cor_filtro': cor_filtro,
221         'idade_filtro': idade_filtro,
222         'cidade_filtro': cidade_filtro
223     }
224
225     return render(request, 'lista_adocao.html', context)

```

Figura 6 - Função “lista_adocao” em *views.py*. Autores, 2023.

Para concluir o desenvolvimento, foi necessário criar as páginas que seriam exibidas aos usuários, juntamente com os scripts e arquivos de estilo. Esses arquivos foram escritos em linguagens HTML, JavaScript e CSS, todas abordadas durante o curso de Fundamentos de Internet e Web na UNIVESP. Um livro que nos orientou nesse processo foi "Desenvolvimento de Software II: Introdução ao Desenvolvimento Web com HTML, CSS, JavaScript e PHP" de Evandro Manara Miletto e Silvia de Castro Bertagnolli.

Foram criadas nove páginas: "index.html," "perdidos.html," "encontrados.html," "adocao.html," "lista_perdidos.html," "lista_encontrados.html," "lista_adocao.html," "form_error.html," e "envio_sucesso.html."

Podemos dividir essas páginas em quatro categorias: I - Menu principal, II - Formulários de Cadastro, III - Tabelas de Consulta e IV - Páginas de Aviso.

Na categoria I, temos a página "index.html" (Figura 7). Nela, o usuário pode escolher entre todas as opções de navegação da plataforma. Para melhorar a experiência do usuário, essas opções são apresentadas em dois módulos diferentes, "Área de Animais Perdidos" e "Área de Adoção de Animais." Para implementar essa funcionalidade, foram usadas duas bibliotecas JavaScript (Figura 8), "bootstrap" e "jQuery." Elas permitem que um menu flutuante apareça de acordo com a opção selecionada pelo usuário (Figuras 9, 10 e 11), tornando a experiência mais amigável.

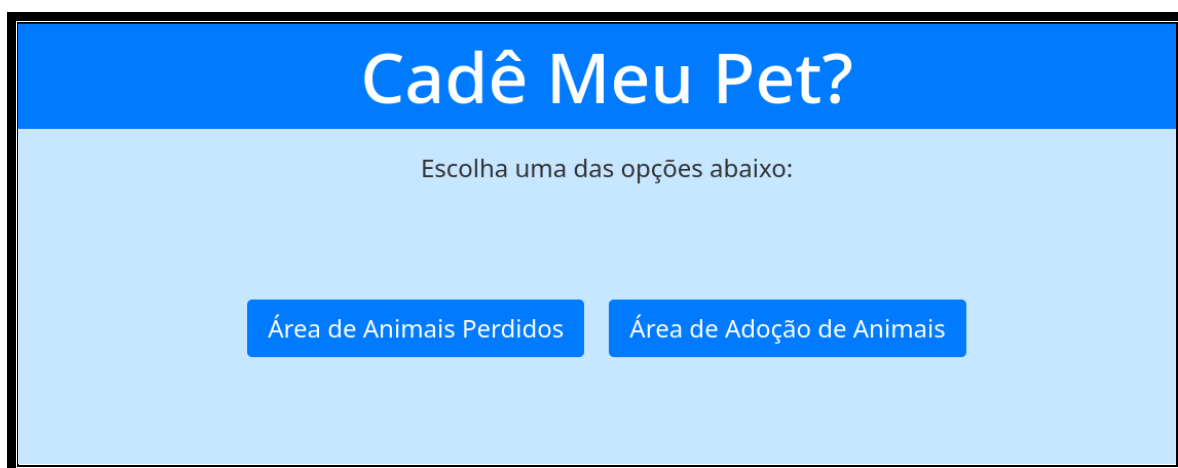


Figura 7 - "index.html". Autores, 2023.

```
3 <head>
4   <title>Cadê Meu Pet</title>
5   <!-- Bibliotecas JavaScript -->
6   <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/css/bootstrap.min.css"
7     integrity="sha384-x0oIHFLEh07PJGoPkLv1IbcEPTNtaed2xpHsD9ESMhqIYd0nLMwNLD69Npy4HI+N" crossorigin="anonymous">
8   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
9   <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>
10  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
11  <!-- Referências dos estilos -->
12  {% load static %}
13  <script src="{% static 'scripts.js' %}"></script>
14  <link rel="stylesheet" type="text/css" href="{% static 'index-styles.css' %}">
15 </head>
```

Figura 8 - "index.html" bibliotecas JavaScript. Autores, 2023.

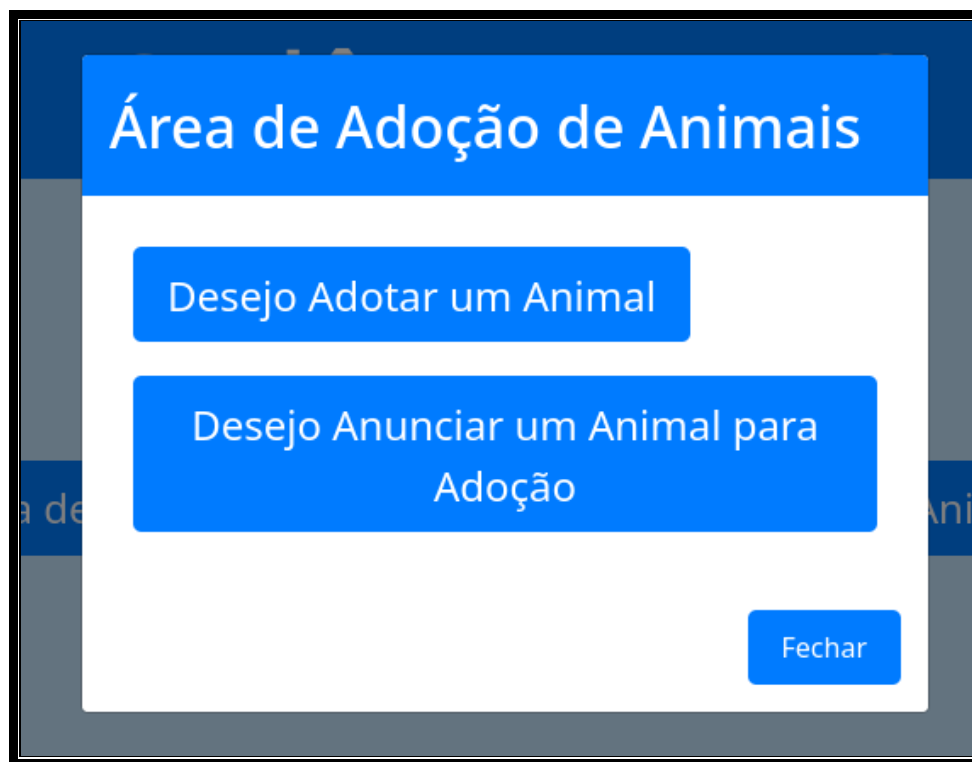


Figura 9 - Menu Área de Adoção de Animais. Autores, 2023.

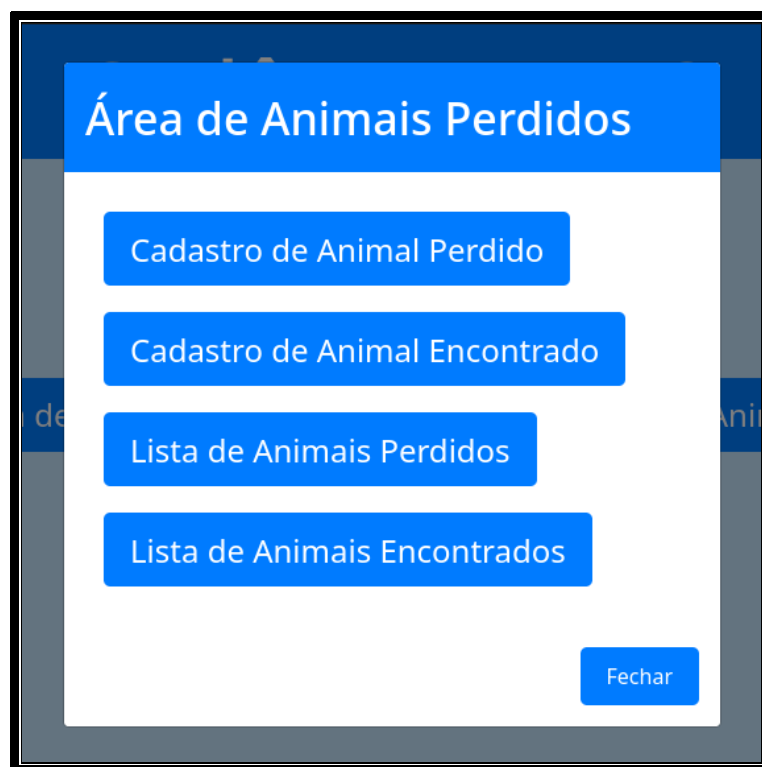


Figura 10 - Menu Área de Animais Perdidos. Autores, 2023.

```

3 <head>
4   <title>Cadê Meu Pet</title>
5   <!-- Bibliotecas JavaScript -->
6   <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/css/bootstrap.min.css"
7     integrity="sha384-xOoIHFLEh07PJGoPkLv1IbcEPTNTaied2xpHsD9ESMhqIYd0nLMwNLD69Npy4HI+N" crossorigin="anonymous">
8   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
9   <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>
10  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
11  <!-- Referências dos estilos -->
12  {% load static %}
13  <script src="{% static 'scripts.js' %}"></script>
14  <link rel="stylesheet" type="text/css" href="{% static 'index-styles.css' %}">
15 </head>

```

Figura 11 - Implementação dos Menus Flutuantes em “index.html”. Autores, 2023.

```

23 <!-- Modal para a Área de Animais Perdidos -->
24 <div class="modal fade" id="animaisPerdidosModal" tabindex="-1" role="dialog" aria-labelledby="animaisPerdidosModallabel" aria-hidden="true">
25   <div class="modal-dialog" role="document">
26     <div class="modal-content">
27       <div class="modal-header">
28         <h5 class="modal-title" id="animaisPerdidosModallabel">Área de Animais Perdidos</h5>
29       </div>
30       <div class="modal-body">
31         <!-- opções de animais perdidos -->
32         <button class="custom-button" onclick="abrir_perdidos()">Cadastro de Animal Perdido</button>
33         <button class="custom-button" onclick="abrir_encontrados()">Cadastro de Animal Encontrado</button>
34         <button class="custom-button" onclick="abrir_lista_perdidos()">Lista de Animais Perdidos</button>
35         <button class="custom-button" onclick="abrir_lista_encontrados()">Lista de Animais Encontrados</button>
36       </div>
37       <div class="modal-footer">
38         <button type="button" class="btn btn-secondary" data-dismiss="modal">Fechar</button>
39       </div>
40     </div>
41   </div>
42 </div>
43 <!-- Modal para a Área de Adoção de Animais -->
44 <div class="modal fade" id="adocaoModal" tabindex="-1" role="dialog" aria-labelledby="adocaoModallabel" aria-hidden="true">
45   <div class="modal-dialog" role="document">
46     <div class="modal-content">
47       <div class="modal-header">
48         <h5 class="modal-title" id="adocaoModallabel">Área de Adoção de Animais</h5>
49       </div>
50       <div class="modal-body">
51         <!-- opções de Adoção de Animais -->
52         <button class="custom-button" onclick="abrir_lista_adocao()">Desejo Adotar um Animal</button>
53         <button class="custom-button" onclick="abrir_adocao()">Desejo Anunciar um Animal para Adoção</button>
54       </div>
55       <div class="modal-footer">
56         <button type="button" class="btn btn-secondary" data-dismiss="modal">Fechar</button>
57       </div>
58     </div>
59   </div>
60 </div>

```

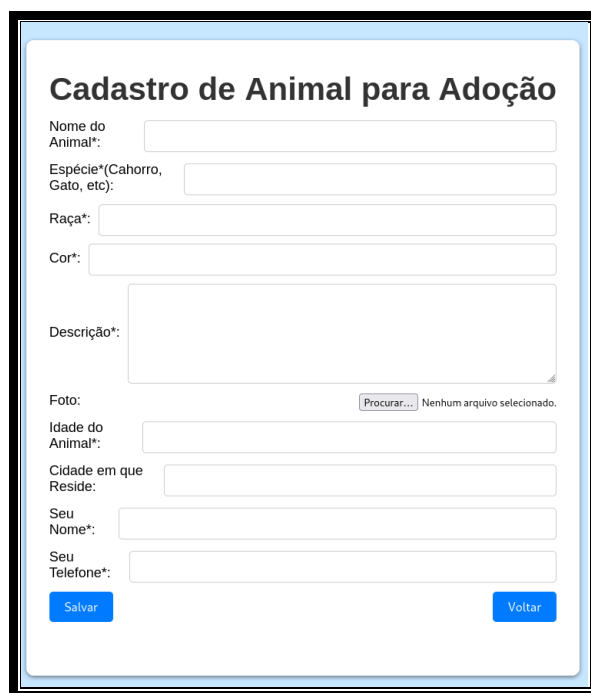
Figura 12 - Implementação dos Menus Flutuantes em “index.html”. Autores, 2023.

Para que os botões direcionem para as outras páginas da plataforma, foram implementadas funções em *JavaScript* (Figura 13) que as abrem em uma nova aba.

```
1  function abrir_adocao() {  
2      // Abre a página de cadastro de animal para adoção em uma nova janela  
3      window.open('/adocao/', '_blank');  
4  }  
5  
6  function abrir_lista_adocao() {  
7      // Abre a página de lista de animal para adoção em uma nova janela  
8      window.open('/lista_adocao/', '_blank');  
9  }  
10  
11 function abrir_perdidos() {  
12     // Abre a página de cadastro de animal perdido em uma nova janela  
13     window.open('/perdidos/', '_blank');  
14 }  
15  
16 function abrir_encontrados() {  
17     // Abre a página de cadastro de animal encontrado em uma nova janela  
18     window.open('/encontrados/', '_blank');  
19 }  
20  
21 function abrir_lista_perdidos() {  
22     // Abre a página de lista de animais perdidos em uma nova janela  
23     window.open('/lista_perdidos/', '_blank');  
24 }  
25  
26 function abrir_lista_encontrados() {  
27     // Abre a página de lista de animais encontrados em uma nova janela  
28     window.open('/lista_encontrados/', '_blank');  
29 }  
30
```

Figura 13 - Implementação das funções dos menus em “scripts.js”. Autores, 2023.

Na categoria II, encontramos as páginas "perdidos.html", "encontrados.html" e "adocao.html." Elas implementam uma API do Django que permite a utilização de formulários para enviar os dados para a nuvem, ou seja, o banco de dados MySQL. Os três códigos compartilham a mesma lógica, variando apenas em seus atributos (Figuras 14 e 15).



The image shows a web form titled "Cadastro de Animal para Adoção". The form is enclosed in a light blue border. It contains several input fields for user registration: "Nome do Animal*", "Espécie*(Cahorro, Gato, etc):", "Raça*", "Cor*", "Descrição*", "Idade do Animal*", "Cidade em que Reside:", "Seu Nome*", and "Seu Telefone*". There is a "Foto:" section with a "Procurar..." button and the text "Nenhum arquivo selecionado.". At the bottom, there are two blue buttons: "Salvar" and "Voltar".

Figura 14- Página "adocao.html". Autores, 2023.

```

1  | DOCTYPE html|
2  <html>
3  <head>
4      {% load static %}
5      <link rel="stylesheet" type="text/css" href="{% static 'form-styles.css' %}">
6      <meta charset="UTF-8">
7      <title>Cadastro de Animal para Adoção</title>
8  </head>
9  <body>
10     <div class="container">
11         <h1>Cadastro de Animal para Adoção</h1>
12         <form method="post" enctype="multipart/form-data">
13             {% csrf_token %}
14             <div class="form-group">
15                 <label class="label" for="nome">Nome do Animal*:</label>
16                 <input class="input" type="text" id="nome" name="nome" required>
17             </div>
18             <div class="form-group">
19                 <label class="label" for="especie">Espécie*(Cahorro, Gato, etc):</label>
20                 <input class="input" type="text" id="especie" name="especie" required>
21             </div>
22             <div class="form-group">
23                 <label class="label" for="raca">Raça*:</label>
24                 <input class="input" type="text" id="raca" name="raca" required>
25             </div>
26             <div class="form-group">
27                 <label class="label" for="cor">Cor*:</label>
28                 <input class="input" type="text" id="cor" name="cor" required>
29             </div>
30             <div class="form-group">
31                 <label class="label" for="descricao">Descrição*:</label>
32                 <textarea class="input" id="descricao" name="descricao" rows="5" cols="40" required></textarea>
33             </div>
34             <div class="form-group">
35                 <label class="label" for="foto">Foto:</label>
36                 <input type="file" id="foto" name="foto">
37             </div>
38             <div class="form-group">
39                 <label class="label" for="idade">Idade do Animal*:</label>
40                 <input class="input" type="text" id="idade" name="idade" required>
41             </div>
42             <div class="form-group">
43                 <label class="label" for="cidade_tutor">Cidade em que Reside:</label>
44                 <input class="input" type="text" id="cidade_tutor" name="cidade_tutor" required>
45             </div>
46             <div class="form-group">
47                 <label class="label" for="nome_tutor">Seu Nome*:</label>
48                 <input class="input" type="text" id="nome_tutor" name="nome_tutor" required>
49             </div>
50             <div class="form-group">
51                 <label for="tel_tutor">Seu Telefone*:</label>
52                 <input type="text" id="tel_tutor" name="tel_tutor" required>
53             </div>
54             <div class="form-group">
55                 <button type="submit">Salvar</button>
56                 <a href="http://127.0.0.1:8000/"><button type="button">Voltar</button></a>
57                 {% if error %}
58                 <p class="error-message">{{ error }}</p>
59                 {% endif %}
60             </div>
61         </form>
62     </div>
63 </body>
64 </html>

```

Figura 15 - Implementação dos formulários em “adocao.html”. Autores, 2023.

Na categoria III, encontramos as páginas "lista_perdidos.html", "lista_encontrados.html" e "lista_adocao.html". Elas fazem requisições ao banco de dados por meio da API mysqlclient para obter as informações cadastradas nele. Os três códigos compartilham a mesma lógica, variando apenas em seus atributos (Figuras 16, 17 e 18).

Lista de Animais para Adoção

Espécie:

Todas as espécies

Raça:

Todas as raças

Cor:

Todas as cores

Data:

Todas as idades

Local:

Todos os locais

Filtrar

Voltar

Nome do Animal	Espécie	Raça	Cor	Descrição	Idade do Animal	Local	Anunciado por	Telefone	Foto
teste	teste	teste	teste	teste	00	teste	teste	teste	Baixar Foto
teste 2	teste 2	teste 2	teste 2	teste 2	02	teste 2	teste 2	teste 2	Baixar Foto
teste 3	teste 3	teste 3	teste 3	teste 3	3	teste 3	teste 3	teste 3	Baixar Foto

Figura 16 - Página "lista_adocao.html". Autores, 2023.

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4  {% load static %}
5  <link rel="stylesheet" type="text/css" href="{% static 'tabela-styles.css' %}">
6  <title>Lista de Animais para Adoção</title>
7  </head>
8  <body>
9  <h1>Lista de Animais para Adoção</h1>
10
11  <div class="filters">
12  <form method="GET" action="{% url 'lista_adocao' %}">
13  <label for="especie">Espécie:</label>
14  <select name="especie" id="especie">
15  <option value="">Todas as espécies</option>
16  {% for especie in especies %}
17  <option value="{{ especie }}">{{ especie }}</option>
18  {% endfor %}
19  </select>
20
21  <label for="raca">Raça:</label>
22  <select name="raca" id="raca">
23  <option value="">Todas as raças</option>
24  {% for raca in racas %}
25  <option value="{{ raca }}">{{ raca }}</option>
26  {% endfor %}
27  </select>
28
29  <label for="cor">Cor:</label>
30  <select name="cor" id="cor">
31  <option value="">Todas as cores</option>
32  {% for cor in cores %}
33  <option value="{{ cor }}">{{ cor }}</option>
34  {% endfor %}
35  </select>
36
37  <label for="idade">Idade:</label>
38  <select name="idade" id="idade">
39  <option value="">Todas as Idades</option>
40  {% for idade in idades %}
41  <option value="{{ idade }}">{{ idade }}</option>
42  {% endfor %}
43  </select>
44

```

Figura 17- Implementação das tabelas e filtros em “lista_adocao.html” (linhas 1 à 44). Autores, 2023.

```

45     <label for="cidade_tutor">Local:</label>
46     <select name="cidade_tutor" id="cidade_tutor">
47         <option value="">Todos os locais</option>
48         {% for cidade in cidades %}
49             <option value="{{ cidade_tutor }}">{{ cidade_tutor }}</option>
50         {% endfor %}
51     </select>
52
53     <button type="submit">Filtrar</button>
54     <a href="http://127.0.0.1:8000/" class="button"><button type="button">Voltar</button></a>
55 </form>
56 </div>
57 <table>
58     <tr>
59         <th>Nome do Animal</th>
60         <th>Espécie</th>
61         <th>Raça</th>
62         <th>Cor</th>
63         <th>Descrição</th>
64         <th>Idade do Animal</th>
65         <th>Local</th>
66         <th>Anunciado por</th>
67         <th>Telefone</th>
68         <th>Foto</th>
69     </tr>
70     {% for animal in animais_adocao %}
71         <tr>
72             <td>{{ animal.nome }}</td>
73             <td>{{ animal.especie }}</td>
74             <td>{{ animal.raca }}</td>
75             <td>{{ animal.cor }}</td>
76             <td>{{ animal.descricao }}</td>
77             <td>{{ animal.idade }}</td>
78             <td>{{ animal.cidade_tutor }}</td>
79             <td>{{ animal.nome_tutor }}</td>
80             <td>{{ animal.tel_tutor }}</td>
81             <td></td>
82         </tr>
83     {% endfor %}
84 </table>
85 </body>
86 </html>

```

Figura 18 - Implementação das tabelas e filtros em “lista_adocao.html” (linhas 45 à 86). Autores, 2023.

Na categoria IV, encontramos as páginas "form_error.html" e "envio_sucesso.html". Estas páginas são exibidas após o usuário enviar um formulário, sendo uma para caso de sucesso (Figura 19) e a outra para caso de erro (Figura 20).

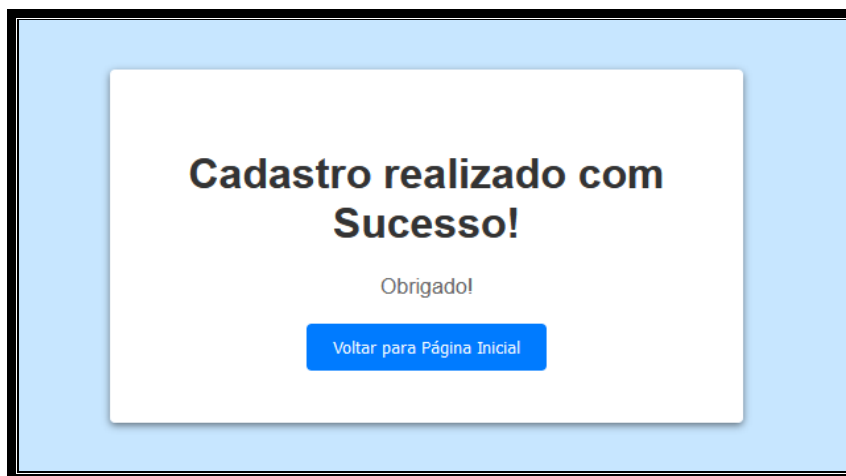


Figura 19 - "envio_sucesso.html". Autores, 2023.

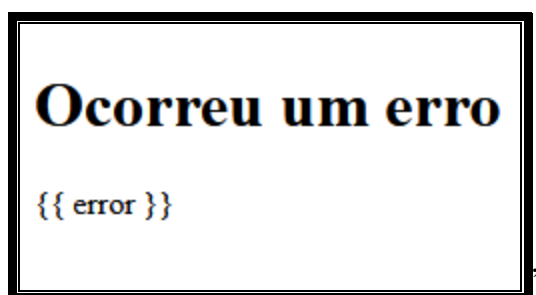


Figura 20 - "form_error.html". Autores, 2023.

O versionamento do projeto foi realizado por meio da plataforma GIT durante todo o seu desenvolvimento. A cada função implementada, uma nova versão era registrada na plataforma. Isso permitiu que toda a equipe participasse do desenvolvimento ativo simultaneamente e protegeu o projeto de possíveis erros de desenvolvimento. Em caso de necessidade, o projeto poderia ser recuperado em versões anteriores. O link para acessar o projeto está disponível em: <https://github.com/Mathguitar/projetointegrador2>.

Na etapa de implementação e teste, o projeto foi desenvolvido com base nas técnicas de desenvolvimento incremental. Isso significa que cada funcionalidade foi desenvolvida separadamente e implementada assim que se tornava funcional. Essa abordagem foi descrita no livro "Engenharia de Software, 10ª edição" de Ian Sommerville, apresentado na disciplina de Engenharia de Software da UNIVESP.

Durante o desenvolvimento do projeto, foram conduzidos testes internos para avaliar a estabilidade da plataforma e identificar possíveis pontos de melhoria. Considerando que a equipe já tinha experiência anterior com as ferramentas utilizadas no projeto, não foram observados erros significativos na fase de implementação.

4 CONSIDERAÇÕES FINAIS

A plataforma que desenvolvemos surge como um importante recurso para apoiar tanto tutores de animais perdidos quanto aqueles interessados em adotar um animal de estimação. Não apenas com o objetivo de ajudar a ONG que visitamos, mas abrangendo toda a comunidade interessada em qualquer região, uma vez que o projeto está disponível no GitHub. Após apresentarmos o projeto à comunidade externa, recebemos muitos elogios, especialmente por incluir uma seção dedicada à adoção. Fomos surpreendidos com a ideia de unir todas as ONGs da cidade em uma única plataforma, a fim de concentrar as informações sobre os animais, facilitando a busca e a localização dos mesmos.

Este projeto foi desenvolvido com base em princípios de Design Thinking, abordagens de desenvolvimento incremental e a utilização de tecnologias sólidas. O objetivo principal é melhorar a vida de animais e pessoas.

Uma das funcionalidades centrais do projeto é fornecer um meio para que tutores de animais perdidos compartilhem informações vitais sobre seus entes queridos peludos. Com a capacidade de especificar raça, cor, nome e detalhes físicos, o objetivo é facilitar o encontro entre animais e seus tutores.

Além disso, a definição de um raio de busca permite uma pesquisa mais precisa, conectando tutores a animais que podem estar relativamente próximos. A empatia desempenhou um papel crucial nesse processo, com a equipe compreendendo as necessidades e emoções dos tutores, baseando-se em experiências pessoais e na coleta de dados através de entrevistas e questionários.

A página também oferece um espaço para a promoção da adoção responsável de animais. Animais disponíveis para adoção são apresentados de forma atraente, incentivando aqueles que desejam adotar um novo membro da família.

Com o objetivo de contribuir para a redução do abandono de animais e promover a adoção como uma alternativa humanitária, o projeto aborda questões sociais importantes.

O desenvolvimento foi fundamentado em tecnologias sólidas, como o framework Django e o banco de dados MySQL. A escolha dessas ferramentas permitiu a criação de uma página web confiável e funcional. Além disso, o uso de bibliotecas JavaScript, como Bootstrap e JQuery, garantiu uma experiência de usuário intuitiva, tornando a navegação fácil e eficaz.

O uso do GIT para o versionamento do projeto permitiu que a equipe colaborasse de forma eficaz e protegesse o projeto de possíveis erros. A abordagem de desenvolvimento incremental, com testes internos durante todo o processo, resultou em uma plataforma estável e eficaz.

A disponibilidade do código no GitHub representa a intenção de compartilhar essa ferramenta com a comunidade. Esperamos que ela seja usada por pessoas em busca de seus animais de estimação desaparecidos e por aqueles que desejam dar a um animal de resgate um lar amoroso. O impacto positivo na vida de animais e pessoas é o cerne deste projeto.

Este projeto, moldado pelo Design Thinking e desenvolvido com paixão e compromisso, é um exemplo de como a tecnologia pode ser usada para o bem. Esperamos que essa plataforma ajude a reunir famílias com seus animais de estimação perdidos e inspire a adoção responsável de animais.

Por meio do uso de tecnologia e empatia, a plataforma aborda questões importantes relacionadas ao bem-estar animal e à conexão entre seres humanos e seus amigos de quatro patas. Estamos orgulhosos deste trabalho e ansiosos para ver o impacto positivo que ele terá na comunidade.

REFERÊNCIAS

ABNT – Associação Brasileira de Normas Técnicas. **NBR 14724**: Informação e documentação. Trabalhos Acadêmicos - Apresentação. Rio de Janeiro: ABNT, 2002.

APPIGATLA, Karthik; MEHTA, Chintan. MySQL 8 Cookbook: Soluções para desafios comuns de programação SQL. São Paulo: Novatec Editora, 2019.

DAUZON, Samuel; BENDORAITIS, Aidas; KRONIKA, Jake. Django 3: Web Development with Python and Django - Third Edition. Packt Publishing, 2020.

DOCS. GITHUB. Introdução a sua conta do Github. Disponível em <https://docs.github.com/pt/get-started/onboarding/getting-started-with-your-github-account>. Acesso em 15 set. 2023.

Ebit & Nielsen. (2021). Webshoppers 43ª edição. Recuperado de https://www.mobiletime.com.br/wp-content/uploads/2021/03/Webshoppers_43.pdf?cdConvenio=CVTR00002030. Acesso em 15 set. 2023.

GOOGLE. Planilha eletrônica. Disponível em: <https://docs.google.com/spreadsheets/d/1raXbGWWkVZN2PaajkvHdPyAoINe4O4mJkmFdyZnup yI/edit?resourcekey#gid=600885320>. Acesso em 15 set. 2023.

G1. (2021, 18 de agosto). Uso da internet no Brasil cresce e chega a 81% da população, diz pesquisa. Recuperado de <https://g1.globo.com/economia/tecnologia/noticia/2021/08/18/uso-da-internet-no-brasil-cresce-e-chega-a-81percent-da-populacao-diz-pesquisa.ghtml>. Acesso em 15 set. 2023.

GOMES, André. Aprendendo Git: do básico ao avançado. Casa do Código, 2021.

Instituto Pet Brasil. Brasil tem quase 185 mil animais resgatados por ONGs, diz instituto. CNN Brasil, São Paulo, 26 fev. 2021. Disponível em: <https://www.cnnbrasil.com.br/nacional/brasil-temquase-185-mil-animais-resgatados-por-ongs-diz-instituto/>. Acesso em 15 set. 2023.

MATTAR, F. N. Pesquisa de marketing. 3.ed. São Paulo: Atlas, 2001.

MARCONI, Marina de Andrade; LAKATOS, Eva Maria. Metodologia do trabalho científico procedimentos básicos, pesquisa bibliográfica, projeto e relatório, publicações e trabalhos científicos. 7. ed. São Paulo: Atlas, 2007.

MELLO, Cleyson de Moraes; ALMEIDA NETTO, José Rogério Moura de; PETRILLO, Regina Pentagna. Para compreender o Design Thinking. Rio de Janeiro: Processo, 2021.

MILETTO, Evandro Manara; BERTAGNOLLI, Silvia de Castro. Desenvolvimento de Software II: Introdução ao Desenvolvimento Web com HTML, CSS, JavaScript e PHP. Porto Alegre: Bookman Companhia Editora, 2014.

Tutorial 01: Escrevendo seu primeiro app Django. Disponível em: <https://docs.djangoproject.com/pt-br/3.1/intro/tutorial01/>. Acesso em 15 set. 2023.

VERGARA, Sylvia Constant. Projetos e Relatórios de Pesquisa em Administração. 6. ed. São Paulo: Atlas, 2005.