



Etec Adolpho Berezin
Mongaguá/SP

Arthur Albuquerque Pinheiro
Fabiano Nascimento de Souza
Natã dos Santos Silva
Ricardo de Abreu Dias

TI ACHEI
Sistema Localizador de Pets e Pessoas
Equipe Tech Work

Orientador Prof. Graciete Henriques dos Santos

Mongaguá

07/2021

Arthur Albuquerque Pinheiro
Fabiano Nascimento de Souza
Natã dos Santos Silva
Ricardo de Abreu Dias

TI ACHEI
Sistema Localizador de Pets e Pessoas

Trabalho de Conclusão de Curso apresentado à Escola Técnica Adolpho Berezin, como parte dos requisitos para a obtenção do título de Técnico em Informática.

Orientador Profa.: Graciete Henriques dos Santos

Mongaguá
07/2020

Dedicamos

Aos Professores, Professora Orientadora e Coordenador do curso que são responsáveis pela concretização desse projeto, pelos ensinamentos, amizade e convívio ao longo desses meses

AGRADECIMENTOS

Agradecemos a instituição ETEC Adolpho Berezin, e a todos os professores que colaboraram no desenvolvimento do projeto, e no nosso ensino ao longo dos semestres, e em especial a professora Graciete Henriques dos Santos.

“A tecnologia move o mundo.”

Steve Jobs

RESUMO

O presente trabalho se baseia nos estudos feitos sobre o desaparecimento de pets, crianças e idosos. Com a evolução tecnológica, pensando na contribuição e impacto positivo na sociedade, foi desenvolvido um sistema gerenciador de pets e pessoas perdidas, possibilitando gerar um QR Code, que ao ser escaneado traz informações do responsável. O projeto tem como objetivo apresentar uma solução ou redução ao grande número desses desaparecidos, causando uma grande melhoria na sociedade e desafogamento de alguns órgãos públicos. Esse estudo é de natureza aplicada, pois procuramos conhecer os problemas específicos em relação a desaparecimentos. A pesquisa é de abordagem quantitativa, pois trata de estatísticas. Foi classificada como pesquisa exploratória, pois houve necessidade de pesquisa bibliográfica, e levantamento de campo para analisar experiências locais em relação a desaparecimentos. Diante dos resultados de pesquisa, observamos uma necessidade de solução a esses problemas, através de um sistema que faça esse gerenciamento de desaparecimento, pelos feedbacks que recebemos no levantamento de campo, vemos resultados positivos em relação ao projeto, esperamos causar uma melhoria na sociedade, trazendo segurança e agilidade na localização.

PALAVRAS-CHAVES: Evolução, Contribuição, Segurança

ABSTRACT

The present work is based on studies carried out on the disappearance of pets, children and the elderly. With the technological evolution, thinking about the contribution and positive impact on society, a management system for pets and lost people was developed, making it possible to generate a QR Code, which, when scanned, brings information about the person in charge. The project aims to present a solution or reduction to the large number of these disappeared, causing a great improvement in society and relieving some public bodies. This study is of an applied nature, as we seek to know the specific problems in relation to disappearances. The research has a quantitative approach, as it deals with statistics. It was classified as exploratory research, as there was a need for bibliographic research, and field survey to analyze local experiences in relation to disappearances. In view of the research results, we observe a need to solve these problems, through a system that manages this disappearance, from the feedbacks we received in the field survey, we see positive results in relation to the project, we hope to cause an improvement in society, bringing security and agility in the location.

Key Words: Evolution, Contribution, Security.

LISTA DE FIGURAS

Figura 1 - Logo da Equipe	19
Figura 2 - Logo do Projeto	20
Figura 3 – Tela de exibição quando o QR Code for acessado	21
Figura 4 - Wireframe Tela de Cadastro.....	22
Figura 5 - Wireframe Tela de Login	22
Figura 6 - Wireframe Cadastro de Pets	23
Figura 7 - Wireframe Cadastro de Pessoas.....	23
Figura 8 - Wireframe Cadastro de Objetos	24
Figura 9 - Modelagem do Banco de Dados do Projeto	25
Figura 10 - Tela cadastro.....	39
Figura 11 - Tela de Login.....	40
Figura 12 - Painel de Cadastro	40
Figura 13 - Tela para Adicionar Pets	41

LISTA DE GRÁFICOS

Gráfico 1 - Resposta sobre ter um pet em casa	11
Gráfico 2 - Resposta sobre pessoas que cuidam de pessoas dependentes ..	11
Gráfico 3 - Respostas sobre ter crianças com responsabilidade	12
Gráfico 4 - Resposta se já perdeu um animal.....	13
Gráfico 5 - Respostas sobre perda de crianças.....	14
Gráfico 6 - Resposta sobre desaparecimento de adultos	14
Gráfico 7 - Respostas a pergunta se o pet foi encontrado.....	14
Gráfico 8 - Respostas sobre aceitação do projeto	16
Gráfico 9 - Resposta de avaliação do projeto	16

SUMÁRIO

LISTA DE FIGURAS.....	7
LISTA DE GRÁFICOS	7
INTRODUÇÃO.....	10
1. Público-Alvo.....	11
1.1 Problematização.....	12
1.2 Proposta - Diferencial.....	15
2. TECNOLOGIAS UTILIZADAS	17
2.1 C#.....	17
2.2 MySQL	17
2.3 ASP Net MVC.....	17
2.4 Azure.....	18
3. EMPRESA	19
3.1 Missão.....	19
3.2 Visão	19
3.3 Valores	19
3.4 Logo	19
3.5 Slogan	20
4. ANÁLISE	21
4.1. Descrição das Funções.....	21
4.2. Wireframe.....	22
4.3. MER – Modelo do Banco de Dados	25
4.4. Create do Banco de Dados	25
4.5 Principais Selects do Banco de Dados.....	38
4.8. Prints das Telas Definitivas	39
4.9 Trecho do Código Fonte.....	41
5. MANUAL DO USUÁRIO	44

CONCLUSÃO	47
REFERÊNCIA BIBLIOGRÁFICA	48

INTRODUÇÃO

É inegável que um pet (animais de estimação) seja uma alegria ao ambiente caseiro, muitos gostam e adotam para cuidar, hoje no Brasil existem muitas opções, sendo possível comprar ou adotar pets abandonados. Segundo a Abinpet é estimado que o Brasil tem a segunda maior população de cães, gatos e aves

O Brasil tem a segunda maior população de cães, gatos e aves canoras e ornamentais em todo o mundo e é o terceiro maior país em população total de animais de estimação. São 54,2 milhões de cães, 23,9 milhões de gatos, 19,1 milhões de peixes, 39,8 milhões de aves e mais 2,3 milhões de outros animais. O total é de 139,3 milhões de pets, o que demonstra a força potencial do nosso setor na economia brasileira (ABINPET, 2022)

Por ser um país muito populoso de cães, gatos e aves, conseqüentemente, existem muitos casos de perda de animais e até abandonos, “a Organização Mundial da Saúde (OMS) estima que o Brasil tenha 30 milhões de animais vivendo nas ruas, sendo aproximadamente 20 milhões de cães e 10 milhões de gatos.” (Jornal SP Norte, 2021). Parte desses pets são perdidos diariamente, outros são abandonados. Essa nota da OMS foi publicada em 2013, por já se passarem quase 10 anos, o volume de pets perdido certamente está muito maior nos dias de hoje.

Outro caso que também tem números grandes que é importante se atentar, são crianças perdidas “Não há registros oficiais de todos os casos de crianças desaparecidas no Brasil, e isto ocorre devido à falta de informação sobre o assunto. A estimativa do Governo Federal são 40 mil crianças todo ano” (Assembleia Legislativa de Sergipe, 2022). Principalmente em lugares litorâneos, existe uma possibilidade grande de perda de crianças nesses ambientes.

É importante ressaltar também, idosos que sofrem de alguma doença de esquecimento como o Alzheimer, infelizmente é mais comum do que se imagina o desaparecimento deles, segundo dados do site do Governo de São Paulo “A 4ª Delegacia de Investigação de Pessoas Desaparecidas registra uma média de aproximadamente 80 desaparecimentos mensais de pessoas com idade acima de 65 anos.” (SP NOTÍCIAS, 2017)

1. PÚBLICO-ALVO

O público-alvo do projeto são as pessoas que possuem um pet, atualmente muitos adotam, seja um cão ou gato, na pesquisa local que fizemos (Gráfico 1), 69,6% das pessoas possuem um pet em casa, desta forma podemos confirmar o tamanho do nosso público alvo local.

Você tem algum pet em casa ?
69 respostas

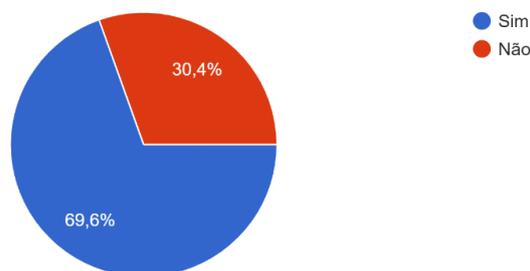


Gráfico 1 - Resposta sobre ter um pet em casa

Considerando que o Brasil tem a segunda maior população de cães e gatos, conseqüentemente, também muitos possuem um pet em casa.

A pesquisa de campo foi realizada através pelo Google Forms durante os meses de maio e junho deste ano, sendo o formulário enviado para pessoas da região de Mongaguá e Itanhaém.

Na pesquisa incluímos também as pessoas que são responsáveis por alguém que sofre com alguma deficiência ou enfermidade que afeta a memória, pois também fazem parte do nosso público-alvo, de acordo com o levantamento de campo que fizemos e demonstrado no Gráfico 2.

Conhece alguém que sofre de Alzheimer ou algum outro tipo de doença de esquecimento, que tenha dependência?
69 respostas

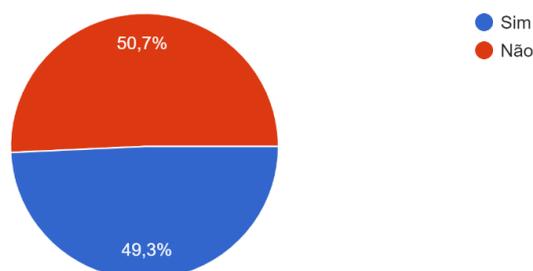


Gráfico 2 - Resposta sobre pessoas que cuidam de pessoas dependentes

Aqueles que possuem alguma criança em sua responsabilidade também são nosso alvo, por isso questionamos em nossa pesquisa esse assunto e demos o resultado no Gráfico 3.

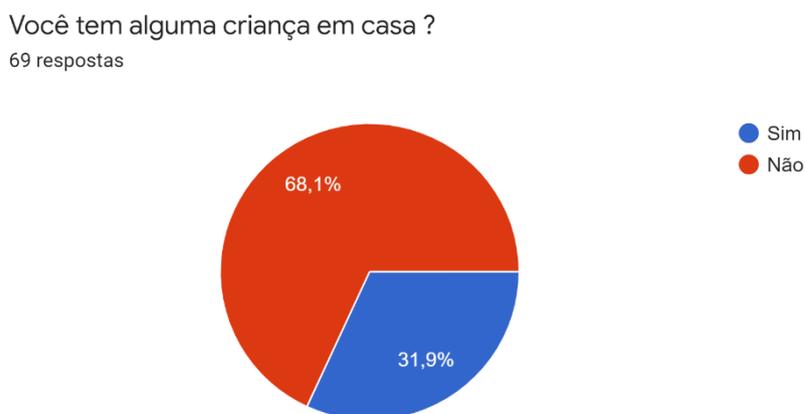


Gráfico 3 - Respostas sobre ter crianças com responsabilidade

1.1 Problematização

Sabemos a quão desesperadora pode ser a ideia de perder um animal de estimação, perder um cãozinho ou gato gera uma angústia grande nos donos, vivenciando isso, não conseguem lidar direito, e não há uma garantia de que será encontrado o pet. Quando essas situações acontecem, causam uma crise na sociedade, cada vez mais o número de pets perdidos cresce, conforme os dados que a OMS traz, muitos daqueles pets são abandonados, mas muito também são perdidos, seja em uma viagem, acidente ou mesmo em uma fuga.

Isso também se aplica a portadores de Alzheimer,

A doença de Alzheimer é uma doença degenerativa do cérebro que acomete pessoas com mais idade. Funções cerebrais como memória, linguagem, cálculo, comportamento são comprometidas de forma lentamente progressiva levando o paciente a uma dependência para executar suas atividades de vida diária. (EINSTEIN, 2022)

Ou seja, quando a pessoa possui a essa doença, perde a independência, precisando que alguém cuide dela e que fique atento aos movimentos. Mas não é sempre que se tem toda a atenção sobre o dependente, em algumas ocasiões a pessoa pode estar ocupada com algo, nesses momentos de desatenção os portadores de Alzheimer correm um risco, como trazem os dados acima, do site do governo de São Paulo, aproximadamente 80 desaparecimentos por mês é um número

consideravelmente grande, sabendo que se deve ter cuidado especial com tais pessoas (SP NOTÍCIAS, 2017).

Devemos falar também sobre as crianças, as quais se deve ter um cuidado especial, principalmente em lugares superlotados, como shows, praias ou eventos. As crianças não têm noção de onde entram ou onde estão, portanto, é muito comum que haja desaparecimento delas nesses lugares, por se dispersarem dos pais e entrarem no meio da multidão. Os números são grandes, um exemplo são as praias do litoral sul de São Paulo que segundo o G1 “Praia Grande tem o maior número de ocorrências da Baixada Santista. Em 2019, foram registrados 144 casos, frente a 152 no ano de 2018.” (G1, 2019) Isso significa que a cada 2 dias do ano quase 1 criança se perde apenas no município de Praia Grande, sendo de muito importante se atentar a esses números.

Ainda trazendo dados de nossa pesquisa de campo, verificamos justamente a perda de animais de estimação, sendo que 66,7% dos entrevistados confirmaram que já perderam ou conhecem alguém que perdeu um animal (Gráfico 4).

Você ou alguém que conhece, teve algum animal de estimação perdido ?
69 respostas

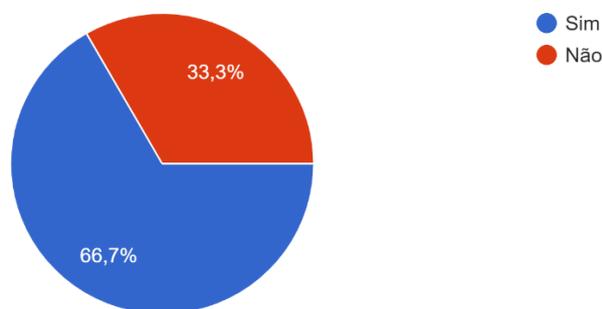


Gráfico 4 - Resposta se já perdeu um animal

Da mesma forma perguntamos sobre a perda de crianças, que retornou uma menor quantidade positiva – 44,9%, porém ainda grande se imaginarmos o desespero dos pais ou responsáveis por essas crianças (Gráfico 5)

Conhece alguém que já perdeu seu filho, ou alguma criança (seja na praia ou algum outro evento)?

69 respostas

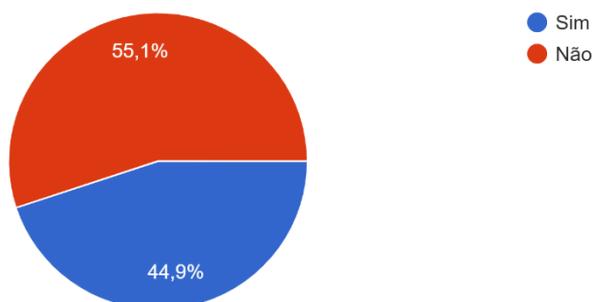


Gráfico 5 - Respostas sobre perda de crianças

Já em relação a desaparecimento de adultos ou pessoas dependentes o valor cresceu para 50,7% (Gráfico 6).

Alguns parentes, amigos ou vizinhos que tiveram problemas com desaparecimento de alguém?

69 respostas

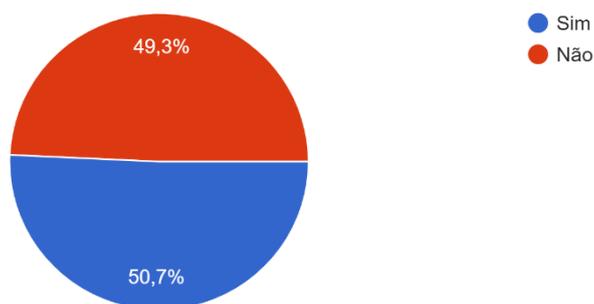


Gráfico 6 - Resposta sobre desaparecimento de adultos

Perceba que na última pergunta, a maior parte do gráfico é “Não foi localizado” (Gráfico 7). Infelizmente, é muito mais comum do que imaginamos esses casos de desaparecimento

A pessoa ou pet foi localizada? em quanto tempo?

69 respostas

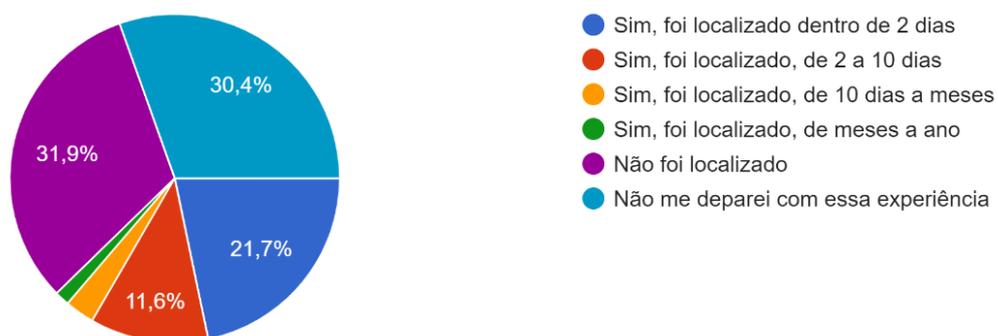


Gráfico 7 - Respostas a pergunta se o pet foi encontrado

1.2 Proposta - Diferencial

Pensando em apresentar possíveis soluções ou para reduzir esses tristes eventos, propomos desenvolver um sistema de gerenciamento de pets ou pessoas e objetos perdidos, que quando feito o cadastro e o login, abre uma possibilidade de cadastrar (seja pessoa, pet ou objetos) e informar a situação, se ele se encontra em segurança, ou se está perdido.

Quando é feito o cadastro, ao anexar a foto e informações, o sistema gera um *QR Code* que ficará disponível para o usuário, ele poderá fazer o download e imprimir, para algumas situações do dia a dia. Não será possível andar com ele apenas impresso, pois certamente o papel, pode estragar rápido, é necessário plastificar ou mandar fazer a laser no medalhão do pet. Ao apontar a câmera do celular para o código ele traz informações do pet como nome, raça e descrição, no caso de pessoas conterá nome, doença (se houver), idade e descrição, e nome do responsável, número de telefone, opção para ligar, mandar uma mensagem ou compartilhar a localização atual.

Com o avanço tecnológico encontramos essa maneira mais rápida e prática de encontrar o seu pet, e principalmente de forma que não se tenha gastos, pois será gratuito esse serviço, pois pretendemos trabalhar com parcerias e patrocínio de grandes empresas, sejam da indústria e serviços para pets como de órgãos públicos. Desta forma o usuário só terá o gasto de plastificar ou fazer qualquer outra proteção do papel ou gravação do *QR Code*, esse é nosso diferencial, fizemos uma pesquisa de um sistema que tenha a mesma ideia que essa, mas não encontramos, sendo ele ainda gratuito.

No levantamento de campo, após uma breve descrição de como seria o projeto, pedimos o *feedback* sobre o quanto seria viável ter um sistema assim. Esse *feedback* foi importante para continuarmos o projeto, pois localizamos o quão importante seria para a sociedade, desta forma perguntamos sobre a aceitação da ideia do uso do *QR Code* (Gráficos 7 e 8)

O que vocês acham de um localizador através do QR Code? Que após feito a leitura mostra um cartão de visitas com dados do responsável (do pet ou pessoa), com uma opção de mensagem, ligação e compartilhamento de localização atual?

69 respostas

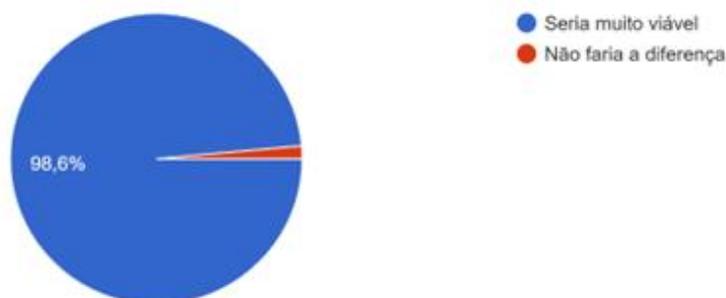


Gráfico 8 - Respostas sobre aceitação do projeto

De 1 a 10, o quanto esse projeto seria viável ?

69 respostas

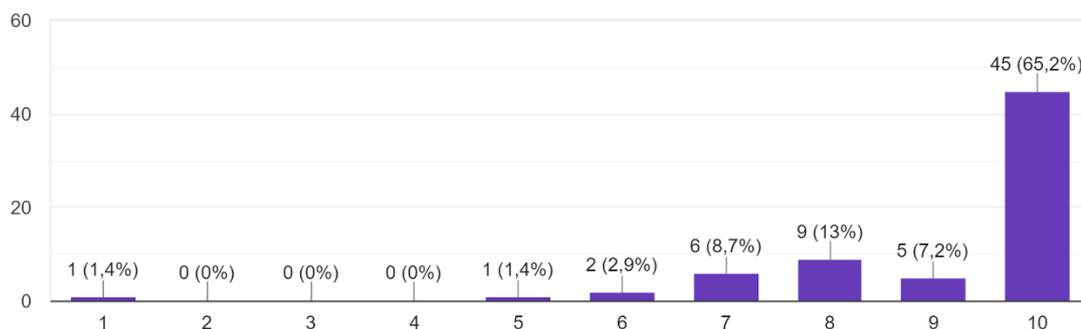


Gráfico 9 - Resposta de avaliação do projeto

Com esse sistema se viabiliza a eficiência da localização, assim podendo desafogar alguns órgãos públicos que sofrem com a grande massa de pets perdidos, que superlotam as zoonoses.

2. TECNOLOGIAS UTILIZADAS

No sistema será empregado as tecnologias as quais adquirimos conhecimento nas aulas de Banco de Dados e Programação, dentre as linguagens e ferramentas serão utilizados C#, ASP NET MVC, MYSQL e AZURE.

2.1 C#

C# é a linguagem de programação a qual utilizaremos para a base desse projeto, sendo uma linguagem de programação orientada a objetos, trará segurança maior para a aplicação e uma proteção dos códigos.

[...]é uma linguagem de programação moderna, orientada a objeto e de tipo seguro. O C# permite que os desenvolvedores criem muitos tipos de aplicativos seguros e robustos que são executados no .NET (MICROSOFT, 2022).

Em C# Uma das ferramentas a serem aplicadas é o Entity Framework que “[...]é um conjunto de tecnologias no ADO.NET que dão suporte ao desenvolvimento de aplicativos de software orientado a dados.” (MICROSOFT, 2022), será responsável pela criação das tabelas do banco de dados e com ele é possível fazer o mapeamento do banco de dados para os recursos da aplicação, leitura e inserção de dados.

2.2 MySQL

O gerenciador de banco de dados que iremos utilizar é o MySQL da Oracle, o processo será gerenciado pela função de migração do Entity-framework, que possibilita criar tabelas através de classes orientadas a objeto que são definidas no escopo do projeto.

MySQL, que é um SGBD (Sistema Gerenciador de Banco de Dados) relacional, que utiliza a linguagem SQL (Structured Query Language, ou traduzindo, Linguagem de Consulta Estruturada). MySQL também é multiusuário e multitarefas. (DEV MEDIA, 2022).

2.3 ASP Net MVC

A aplicação será desenvolvida em cima do framework ASP NET MVC, que foi criado pela Microsoft, ele é uma solução proposta para a criação de sistemas *web server-side*, uma aplicação que roda do lado do servidor utilizando a linguagem de programação C# e sendo executado em servidores IIS.

2.4 Azure

Ao concluir o desenvolvimento do sistema web, para o correto andamento do site e consistência das funcionalidades ele deve ser hospedado em algum lugar, esse ambiente seria uma máquina ou um serviço de hospedagem. O Azure será a plataforma utilizada para fazer essa sustentação do sistema.

“O Azure é uma plataforma de nuvem projetada para simplificar o processo de criação de aplicativos modernos. Se você optar por hospedar seus aplicativos inteiramente no Azure ou estender seus aplicativos locais com os serviços do Azure, o Azure ajuda você a criar aplicativos escalonáveis, confiáveis e mantenedíveis. (MICROSOFT, 2022).

3. EMPRESA

A Tech Work é uma nova empresa criada com intuito de ajudar pessoas através do uso de tecnologia para soluções de problemas, desenvolvida para o trabalho de TCC da Etec Adolpho Berezin.

3.1 Missão

Fornecer uma solução eficaz promovendo a satisfação do cliente em relação à nossa empresa e ao serviço prestado, criando fidelidade entre usuário e a empresa garantindo assim uma confiabilidade no produto.

3.2 Visão

Ser reconhecida em aplicar soluções eficientes aos problemas corriqueiros e diários na vida das pessoas com inovação tecnológica.

3.3 Valores

Contribui com a realidade do mundo atual, sendo transparente e ético ao público-alvo, tendo eficiência, inovação, serviço de qualidade e comprometimento com o crescimento.

3.4 Logo



Figura 1 - Logo da Equipe

Fonte: Criado pela equipe



Figura 2 - Logo do Projeto

Fonte: Criado pela equipe

3.5 Slogan

Tech Work, sempre buscando o melhor que a tecnologia pode proporcionar.

4. ANÁLISE

4.1. Descrição das Funções

Na tela de cadastro o usuário terá de colocar as suas informações, como e-mail, senha, nome, telefone e foto, facebook e Instagram são opcionais. Após efetuar esse cadastro, ele fará um login e será direcionado para o painel de cadastro de pet, no qual haverá um botão “adicionar” que quando clicado, redireciona para uma tela de cadastrar o seu pet, o nome que ele atende, se ele é gato ou cachorro, raça, anexar uma foto e a descrição do pet. Ou poderá cadastrar uma pessoa, seja criança ou idoso colocando nome, idade, deficiência ou doença (se houver alguma) e descrição breve sobre ela. Quando esse processo é concluído, é gerado um QR Code pelo qual conterà todas essas informações cadastradas em forma de um cartão de visita, então, quando escaneado, será redirecionado para um site semelhante à Figura 3:

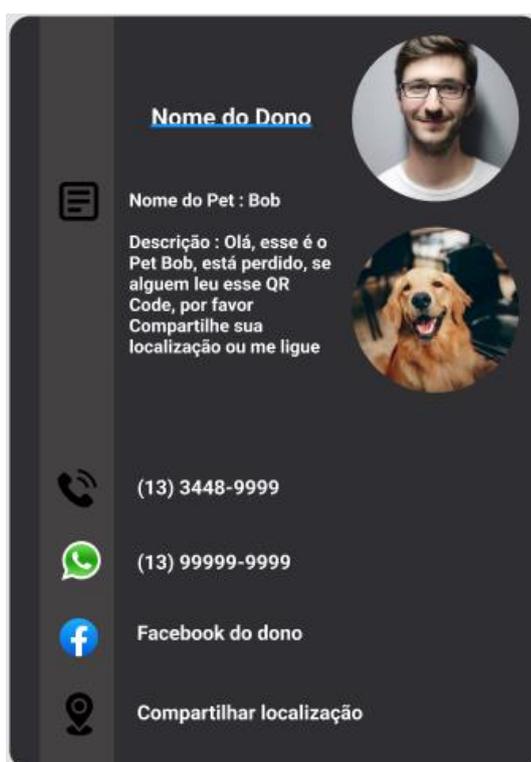


Figura 3 – Tela de exibição quando o QR Code for acessado

Fonte: Autores

No painel de cadastro, terão 3 botões, um para excluir um cadastro feito, outro para visualizar o QR Code (baixá-lo e ver como aparece quando escaneado) outro para alterar informações ou status, por exemplo, quando um pet sumir terá que alterar o status dele para desaparecido, assim quando alguém o encontrar e escanear o código, verá que ele está perdido e o devolverá em segurança para o dono.

4.2. Wireframe

The wireframe shows a registration form with the following elements:

- Title:** Cadastrar-se
- Left Panel:** A vertical grey box containing the text "Ti Achei" at the top and "Ja tem conta ? Fazer Login" at the bottom.
- Form Fields:**
 - Nome: A single-line text input field.
 - Telefone: A single-line text input field.
 - E-mail: A single-line text input field.
 - Senha: A single-line text input field.
 - Facebook(url): A single-line text input field.
 - Instagram(url): A single-line text input field.
 - Upload Imagem: A wide, rounded rectangular button.
 - Cadastrar: A wide, rounded rectangular button at the bottom.

Figura 4 - Wireframe Tela de Cadastro

The wireframe shows a login form with the following elements:

- Title:** Login
- Left Panel:** A vertical grey box containing the text "Ti Achei" at the top and "Não possui conta ? Cadastrar-se" at the bottom.
- Form Fields:**
 - E-mail: A single-line text input field.
 - Senha: A single-line text input field.
 - Entrar: A wide, rounded rectangular button at the bottom.

Figura 5 - Wireframe Tela de Login



Figura 6 - Wireframe Cadastro de Pets



Figura 7 - Wireframe Cadastro de Pessoas



Figura 8 - Wireframe Cadastro de Objetos

4.3. MER – Modelo do Banco de Dados

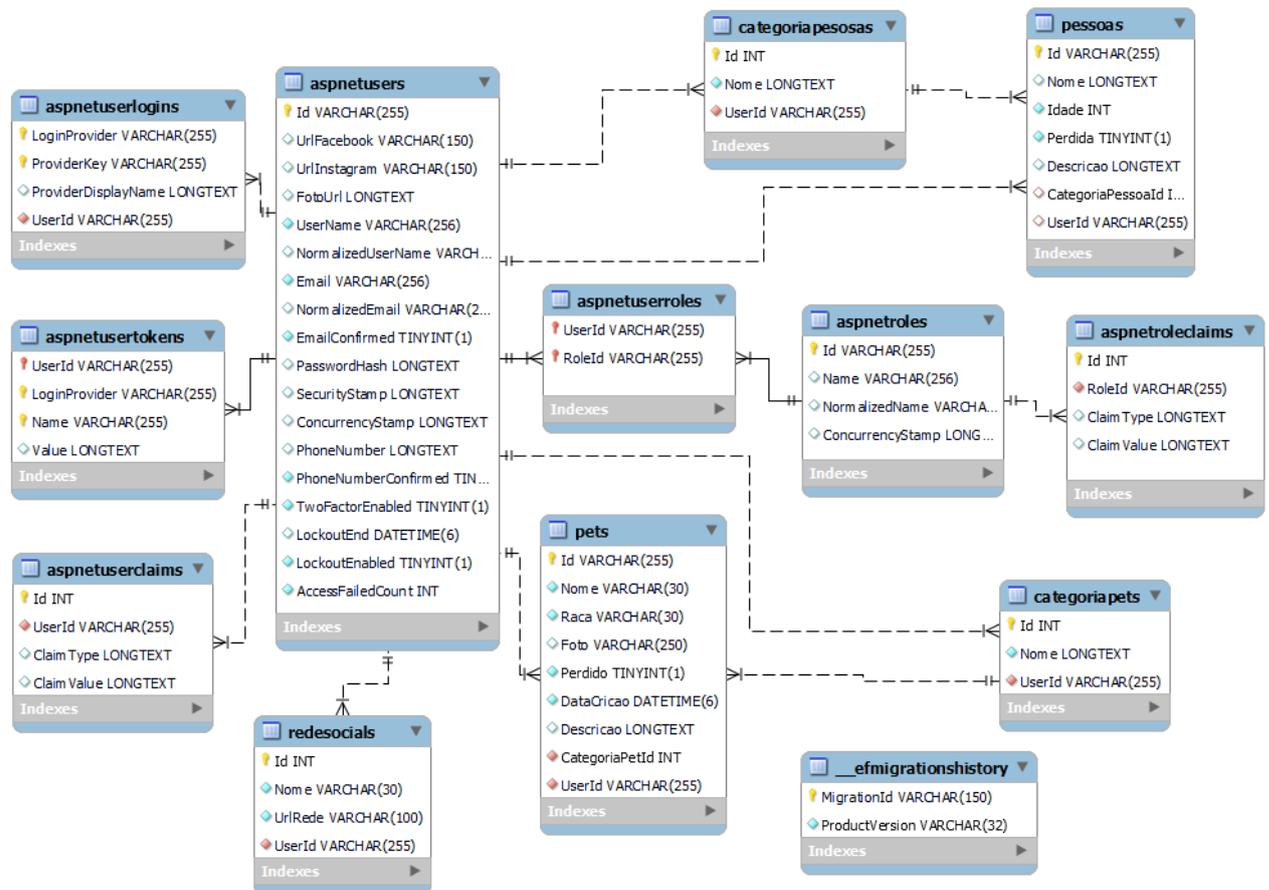


Figura 9 - Modelagem do Banco de Dados do Projeto

4.4. Create do Banco de Dados

O código abaixo representa uma criação do banco de dados MySQL através da ferramenta Entity Framework.

```
using
System;

using Microsoft.EntityFrameworkCore.Metadata;
using Microsoft.EntityFrameworkCore.Migrations;

#nullable disable

namespace TiAchei_Tcc.Migrations
{
    public partial class init : Migration
    {
        protected override void Up(MigrationBuilder migrationBuilder)
```

```

{
    migrationBuilder.AlterDatabase()
        .Annotation("MySQL:CharSet", "utf8mb4");

    migrationBuilder.CreateTable(
        name: "AspNetRoles",
        columns: table => new
        {
            Id = table.Column<string>(type: "varchar(255)", nullable:
false)
                .Annotation("MySQL:CharSet", "utf8mb4"),
            Name = table.Column<string>(type: "varchar(256)",
maxLength: 256, nullable: true)
                .Annotation("MySQL:CharSet", "utf8mb4"),
            NormalizedName = table.Column<string>(type:
"varchar(256)", maxLength: 256, nullable: true)
                .Annotation("MySQL:CharSet", "utf8mb4"),
            ConcurrencyStamp = table.Column<string>(type: "longtext",
nullable: true)
                .Annotation("MySQL:CharSet", "utf8mb4")
        },
        constraints: table =>
        {
            table.PrimaryKey("PK_AspNetRoles", x => x.Id);
        })
        .Annotation("MySQL:CharSet", "utf8mb4");

    migrationBuilder.CreateTable(
        name: "AspNetUsers",
        columns: table => new
        {
            Id = table.Column<string>(type: "varchar(255)", nullable:
false)
                .Annotation("MySQL:CharSet", "utf8mb4"),
            UrlFacebook = table.Column<string>(type: "VARCHAR(150)",
nullable: true)
                .Annotation("MySQL:CharSet", "utf8mb4"),
            UrlInstagram = table.Column<string>(type: "VARCHAR(150)",
nullable: true)
                .Annotation("MySQL:CharSet", "utf8mb4"),
            FotoUrl = table.Column<string>(type: "longtext", nullable:
true)
                .Annotation("MySQL:CharSet", "utf8mb4"),

```

```

        UserName = table.Column<string>(type: "varchar(256)",
maxLength: 256, nullable: false)
        .Annotation("MySQL:CharSet", "utf8mb4"),
        NormalizedUserName = table.Column<string>(type:
"varchar(256)", maxLength: 256, nullable: true)
        .Annotation("MySQL:CharSet", "utf8mb4"),
        Email = table.Column<string>(type: "varchar(256)",
maxLength: 256, nullable: false)
        .Annotation("MySQL:CharSet", "utf8mb4"),
        NormalizedEmail = table.Column<string>(type:
"varchar(256)", maxLength: 256, nullable: true)
        .Annotation("MySQL:CharSet", "utf8mb4"),
        EmailConfirmed = table.Column<bool>(type: "tinyint(1)",
nullable: false),
        PasswordHash = table.Column<string>(type: "longtext",
nullable: true)
        .Annotation("MySQL:CharSet", "utf8mb4"),
        SecurityStamp = table.Column<string>(type: "longtext",
nullable: true)
        .Annotation("MySQL:CharSet", "utf8mb4"),
        ConcurrencyStamp = table.Column<string>(type: "longtext",
nullable: true)
        .Annotation("MySQL:CharSet", "utf8mb4"),
        PhoneNumber = table.Column<string>(type: "longtext",
nullable: true)
        .Annotation("MySQL:CharSet", "utf8mb4"),
        PhoneNumberConfirmed = table.Column<bool>(type:
"tinyint(1)", nullable: false),
        TwoFactorEnabled = table.Column<bool>(type: "tinyint(1)",
nullable: false),
        LockoutEnd = table.Column<DateTimeOffset>(type:
"datetime(6)", nullable: true),
        LockoutEnabled = table.Column<bool>(type: "tinyint(1)",
nullable: false),
        AccessFailedCount = table.Column<int>(type: "int",
nullable: false)
    },
    constraints: table =>
    {
        table.PrimaryKey("PK_AspNetUsers", x => x.Id);
    })
    .Annotation("MySQL:CharSet", "utf8mb4");

migrationBuilder.CreateTable(
    name: "AspNetRoleClaims",

```

```

        columns: table => new
        {
            Id = table.Column<int>(type: "int", nullable: false)
                .Annotation("MySQL:ValueGenerationStrategy",
MySQLValueGenerationStrategy.IdentityColumn),
            RoleId = table.Column<string>(type: "varchar(255)",
nullable: false)
                .Annotation("MySQL:CharSet", "utf8mb4"),
            ClaimType = table.Column<string>(type: "longtext",
nullable: true)
                .Annotation("MySQL:CharSet", "utf8mb4"),
            ClaimValue = table.Column<string>(type: "longtext",
nullable: true)
                .Annotation("MySQL:CharSet", "utf8mb4")
        },
        constraints: table =>
        {
            table.PrimaryKey("PK_AspNetRoleClaims", x => x.Id);
            table.ForeignKey(
                name: "FK_AspNetRoleClaims_AspNetRoles_RoleId",
                column: x => x.RoleId,
                principalTable: "AspNetRoles",
                principalColumn: "Id",
                onDelete: ReferentialAction.Cascade);
        })
        .Annotation("MySQL:CharSet", "utf8mb4");

migrationBuilder.CreateTable(
    name: "AspNetUserClaims",
    columns: table => new
    {
        Id = table.Column<int>(type: "int", nullable: false)
            .Annotation("MySQL:ValueGenerationStrategy",
MySQLValueGenerationStrategy.IdentityColumn),
        UserId = table.Column<string>(type: "varchar(255)",
nullable: false)
            .Annotation("MySQL:CharSet", "utf8mb4"),
        ClaimType = table.Column<string>(type: "longtext",
nullable: true)
            .Annotation("MySQL:CharSet", "utf8mb4"),
        ClaimValue = table.Column<string>(type: "longtext",
nullable: true)
            .Annotation("MySQL:CharSet", "utf8mb4")
    },
    constraints: table =>

```

```

        {
            table.PrimaryKey("PK_AspNetUserClaims", x => x.Id);
            table.ForeignKey(
                name: "FK_AspNetUserClaims_AspNetNetUsers_UserId",
                column: x => x.UserId,
                principalTable: "AspNetUsers",
                principalColumn: "Id",
                onDelete: ReferentialAction.Cascade);
        })
        .Annotation("MySql:CharSet", "utf8mb4");

migrationBuilder.CreateTable(
    name: "AspNetUserLogins",
    columns: table => new
    {
        LoginProvider = table.Column<string>(type: "varchar(255)",
nullable: false)
            .Annotation("MySql:CharSet", "utf8mb4"),
        ProviderKey = table.Column<string>(type: "varchar(255)",
nullable: false)
            .Annotation("MySql:CharSet", "utf8mb4"),
        ProviderDisplayName = table.Column<string>(type:
"longtext", nullable: true)
            .Annotation("MySql:CharSet", "utf8mb4"),
        UserId = table.Column<string>(type: "varchar(255)",
nullable: false)
            .Annotation("MySql:CharSet", "utf8mb4")
    },
    constraints: table =>
    {
        table.PrimaryKey("PK_AspNetUserLogins", x => new {
x.LoginProvider, x.ProviderKey });
        table.ForeignKey(
            name: "FK_AspNetUserLogins_AspNetNetUsers_UserId",
            column: x => x.UserId,
            principalTable: "AspNetUsers",
            principalColumn: "Id",
            onDelete: ReferentialAction.Cascade);
    })
    .Annotation("MySql:CharSet", "utf8mb4");

migrationBuilder.CreateTable(
    name: "AspNetUserRoles",
    columns: table => new

```

```

        {
            UserId = table.Column<string>(type: "varchar(255)",
nullable: false)
                .Annotation("MySql:CharSet", "utf8mb4"),
            RoleId = table.Column<string>(type: "varchar(255)",
nullable: false)
                .Annotation("MySql:CharSet", "utf8mb4")
        },
        constraints: table =>
        {
            table.PrimaryKey("PK_AspNetUserRoles", x => new {
x.UserId, x.RoleId });
            table.ForeignKey(
                name: "FK_AspNetUserRoles_AspNetRoles_RoleId",
                column: x => x.RoleId,
                principalTable: "AspNetRoles",
                principalColumn: "Id",
                onDelete: ReferentialAction.Cascade);
            table.ForeignKey(
                name: "FK_AspNetUserRoles_AspNetUsers_UserId",
                column: x => x.UserId,
                principalTable: "AspNetUsers",
                principalColumn: "Id",
                onDelete: ReferentialAction.Cascade);
        })
        .Annotation("MySql:CharSet", "utf8mb4");

migrationBuilder.CreateTable(
    name: "AspNetUserTokens",
    columns: table => new
    {
        UserId = table.Column<string>(type: "varchar(255)",
nullable: false)
            .Annotation("MySql:CharSet", "utf8mb4"),
        LoginProvider = table.Column<string>(type: "varchar(255)",
nullable: false)
            .Annotation("MySql:CharSet", "utf8mb4"),
        Name = table.Column<string>(type: "varchar(255)",
nullable: false)
            .Annotation("MySql:CharSet", "utf8mb4"),
        Value = table.Column<string>(type: "longtext", nullable:
true)
            .Annotation("MySql:CharSet", "utf8mb4")
    },
    constraints: table =>

```

```

        {
            table.PrimaryKey("PK_AspNetUserTokens", x => new {
                x.UserId, x.LoginProvider, x.Name });
            table.ForeignKey(
                name: "FK_AspNetUserTokens_AspNetUsers_UserId",
                column: x => x.UserId,
                principalTable: "AspNetUsers",
                principalColumn: "Id",
                onDelete: ReferentialAction.Cascade);
        })
        .Annotation("MySQL:CharSet", "utf8mb4");

migrationBuilder.CreateTable(
    name: "CategoriaPesosas",
    columns: table => new
    {
        Id = table.Column<int>(type: "int", nullable: false)
            .Annotation("MySQL:ValueGenerationStrategy",
                MySQLValueGenerationStrategy.IdentityColumn),
        Nome = table.Column<string>(type: "longtext", nullable:
false)
            .Annotation("MySQL:CharSet", "utf8mb4"),
        UserId = table.Column<string>(type: "varchar(255)",
nullable: false)
            .Annotation("MySQL:CharSet", "utf8mb4")
    },
    constraints: table =>
    {
        table.PrimaryKey("PK_CategoriaPesosas", x => x.Id);
        table.ForeignKey(
            name: "FK_CategoriaPesosas_AspNetUsers_UserId",
            column: x => x.UserId,
            principalTable: "AspNetUsers",
            principalColumn: "Id",
            onDelete: ReferentialAction.Cascade);
    })
    .Annotation("MySQL:CharSet", "utf8mb4");

migrationBuilder.CreateTable(
    name: "CategoriaPets",
    columns: table => new
    {
        Id = table.Column<int>(type: "int", nullable: false)

```

```

        .Annotation("MySql:ValueGenerationStrategy",
MySqlValueGenerationStrategy.IdentityColumn),
        Nome = table.Column<string>(type: "longtext", nullable:
false)

        .Annotation("MySql:CharSet", "utf8mb4"),
        UserId = table.Column<string>(type: "varchar(255)",
nullable: false)

        .Annotation("MySql:CharSet", "utf8mb4")
    },
    constraints: table =>
    {
        table.PrimaryKey("PK_CategoriaPets", x => x.Id);
        table.ForeignKey(
            name: "FK_CategoriaPets_AspNetUsers_UserId",
            column: x => x.UserId,
            principalTable: "AspNetUsers",
            principalColumn: "Id",
            onDelete: ReferentialAction.Cascade);
    })
    .Annotation("MySql:CharSet", "utf8mb4");

migrationBuilder.CreateTable(
    name: "RedeSocials",
    columns: table => new
    {
        Id = table.Column<int>(type: "int", nullable: false)
            .Annotation("MySql:ValueGenerationStrategy",
MySqlValueGenerationStrategy.IdentityColumn),
        Nome = table.Column<string>(type: "VARCHAR(30)", nullable:
false)

        .Annotation("MySql:CharSet", "utf8mb4"),
        UrlRede = table.Column<string>(type: "VARCHAR(100)",
nullable: false)

        .Annotation("MySql:CharSet", "utf8mb4"),
        UserId = table.Column<string>(type: "varchar(255)",
nullable: false)

        .Annotation("MySql:CharSet", "utf8mb4")
    },
    constraints: table =>
    {
        table.PrimaryKey("PK_RedeSocials", x => x.Id);
        table.ForeignKey(
            name: "FK_RedeSocials_AspNetUsers_UserId",
            column: x => x.UserId,
            principalTable: "AspNetUsers",

```

```

        principalColumn: "Id",
        onDelete: ReferentialAction.Cascade);
    })
    .Annotation("MySql:CharSet", "utf8mb4");

migrationBuilder.CreateTable(
    name: "Pessoas",
    columns: table => new
    {
        Id = table.Column<string>(type: "varchar(255)", nullable:
false)
            .Annotation("MySql:CharSet", "utf8mb4"),
        Nome = table.Column<string>(type: "longtext", nullable:
true)
            .Annotation("MySql:CharSet", "utf8mb4"),
        Idade = table.Column<int>(type: "int", nullable: false),
        Perdida = table.Column<bool>(type: "tinyint(1)", nullable:
false),
        Descricao = table.Column<string>(type: "longtext",
nullable: true)
            .Annotation("MySql:CharSet", "utf8mb4"),
        CategoriaPessoaId = table.Column<int>(type: "int",
nullable: true),
        UserId = table.Column<string>(type: "varchar(255)",
nullable: true)
            .Annotation("MySql:CharSet", "utf8mb4")
    },
    constraints: table =>
    {
        table.PrimaryKey("PK_Pessoas", x => x.Id);
        table.ForeignKey(
            name: "FK_Pessoas_AspNetUsers_UserId",
            column: x => x.UserId,
            principalTable: "AspNetUsers",
            principalColumn: "Id");
        table.ForeignKey(
            name: "FK_Pessoas_CategoriaPesosas_CategoriaPessoaId",
            column: x => x.CategoriaPessoaId,
            principalTable: "CategoriaPesosas",
            principalColumn: "Id");
    })
    .Annotation("MySql:CharSet", "utf8mb4");

migrationBuilder.CreateTable(

```

```

name: "Pets",
columns: table => new
{
    Id = table.Column<string>(type: "varchar(255)", nullable:
false)
        .Annotation("MySQL:CharSet", "utf8mb4"),
    Nome = table.Column<string>(type: "VARCHAR(30)", nullable:
false)
        .Annotation("MySQL:CharSet", "utf8mb4"),
    Raca = table.Column<string>(type: "VARCHAR(30)", nullable:
false)
        .Annotation("MySQL:CharSet", "utf8mb4"),
    Foto = table.Column<string>(type: "VARCHAR(250)",
nullable: true)
        .Annotation("MySQL:CharSet", "utf8mb4"),
    Perdido = table.Column<bool>(type: "tinyint(1)", nullable:
false),
    DataCricao = table.Column<DateTime>(type: "datetime(6)",
nullable: false),
    Descricao = table.Column<string>(type: "longtext",
nullable: true)
        .Annotation("MySQL:CharSet", "utf8mb4"),
    CategoriaPetId = table.Column<int>(type: "int", nullable:
false),
    UserId = table.Column<string>(type: "varchar(255)",
nullable: false)
        .Annotation("MySQL:CharSet", "utf8mb4")
},
constraints: table =>
{
    table.PrimaryKey("PK_Pets", x => x.Id);
    table.ForeignKey(
        name: "FK_Pets_AspNetUsers_UserId",
        column: x => x.UserId,
        principalTable: "AspNetUsers",
        principalColumn: "Id",
        onDelete: ReferentialAction.Cascade);
    table.ForeignKey(
        name: "FK_Pets_CategoriaPets_CategoriaPetId",
        column: x => x.CategoriaPetId,
        principalTable: "CategoriaPets",
        principalColumn: "Id",
        onDelete: ReferentialAction.Cascade);
})
.Annotation("MySQL:CharSet", "utf8mb4");

```

```
migrationBuilder.CreateIndex(  
    name: "IX_AspNetRoleClaims_RoleId",  
    table: "AspNetRoleClaims",  
    column: "RoleId");
```

```
migrationBuilder.CreateIndex(  
    name: "RoleNameIndex",  
    table: "AspNetRoles",  
    column: "NormalizedName",  
    unique: true);
```

```
migrationBuilder.CreateIndex(  
    name: "IX_AspNetUserClaims_UserId",  
    table: "AspNetUserClaims",  
    column: "UserId");
```

```
migrationBuilder.CreateIndex(  
    name: "IX_AspNetUserLogins_UserId",  
    table: "AspNetUserLogins",  
    column: "UserId");
```

```
migrationBuilder.CreateIndex(  
    name: "IX_AspNetUserRoles_RoleId",  
    table: "AspNetUserRoles",  
    column: "RoleId");
```

```
migrationBuilder.CreateIndex(  
    name: "EmailIndex",  
    table: "AspNetUsers",  
    column: "NormalizedEmail");
```

```
migrationBuilder.CreateIndex(  
    name: "UserNameIndex",  
    table: "AspNetUsers",  
    column: "NormalizedUserName",  
    unique: true);
```

```
migrationBuilder.CreateIndex(  
    name: "IX_AspNetUsers_UserId",  
    table: "AspNetUsers",  
    column: "UserId");
```

```
        name: "IX_CategoriaPesosas_UserId",
        table: "CategoriaPesosas",
        column: "UserId");

migrationBuilder.CreateIndex(
    name: "IX_CategoriaPets_UserId",
    table: "CategoriaPets",
    column: "UserId");

migrationBuilder.CreateIndex(
    name: "IX_Pessoas_CategoriaPessoaId",
    table: "Pessoas",
    column: "CategoriaPessoaId");

migrationBuilder.CreateIndex(
    name: "IX_Pessoas_UserId",
    table: "Pessoas",
    column: "UserId");

migrationBuilder.CreateIndex(
    name: "IX_Pets_CategoriaPetId",
    table: "Pets",
    column: "CategoriaPetId");

migrationBuilder.CreateIndex(
    name: "IX_Pets_UserId",
    table: "Pets",
    column: "UserId");

migrationBuilder.CreateIndex(
    name: "IX_RedeSocials_UserId",
    table: "RedeSocials",
    column: "UserId");
}

protected override void Down(MigrationBuilder migrationBuilder)
{
    migrationBuilder.DropTable(
        name: "AspNetRoleClaims");
}
```

```
migrationBuilder.DropTable(  
    name: "AspNetUserClaims");  
  
migrationBuilder.DropTable(  
    name: "AspNetUserLogins");  
  
migrationBuilder.DropTable(  
    name: "AspNetUserRoles");  
  
migrationBuilder.DropTable(  
    name: "AspNetUserTokens");  
  
migrationBuilder.DropTable(  
    name: "Pessoas");  
  
migrationBuilder.DropTable(  
    name: "Pets");  
  
migrationBuilder.DropTable(  
    name: "RedeSocials");  
  
migrationBuilder.DropTable(  
    name: "AspNetRoles");  
  
migrationBuilder.DropTable(  
    name: "CategoriaPesosas");  
  
migrationBuilder.DropTable(  
    name: "CategoriaPets");  
  
migrationBuilder.DropTable(  
    name: "AspNetUsers");  
    }  
    }  
}
```

4.5 Principais Selects do Banco de Dados

Selects feitos com o ORM Entity Framework

```

using System.Reflection.Metadata.Ecma335;
using Microsoft.AspNetCore.Routing.Tree;
using Microsoft.EntityFrameworkCore;
using Microsoft.EntityFrameworkCore.Metadata.Internal;
using TiAchei_Tcc.Db;
using TiAchei_Tcc.Models;
using TiAchei_Tcc.Repository.Interfaces;

namespace TiAchei_Tcc.Repository
{
    public class PetRepository : IPetRepository
    {
        private readonly AppDbContextMysql _dbContext;
        public PetRepository(AppDbContextMysql dbContext)
        {
            _dbContext = dbContext;
        }
        public async Task CreatePet(Pet model)
        {
            await _dbContext.Pets.AddAsync(model);
            await _dbContext.SaveChangesAsync();
        }

        public async Task<bool> DeletePet(string id)
        {
            var petResult = await GetBydId(id);
            if(petResult == null) return false;
            _dbContext.Pets.Remove(petResult);
            if(await _dbContext.SaveChangesAsync() > 0){
                return true;
            }
            else{
                return false;
            }
        }

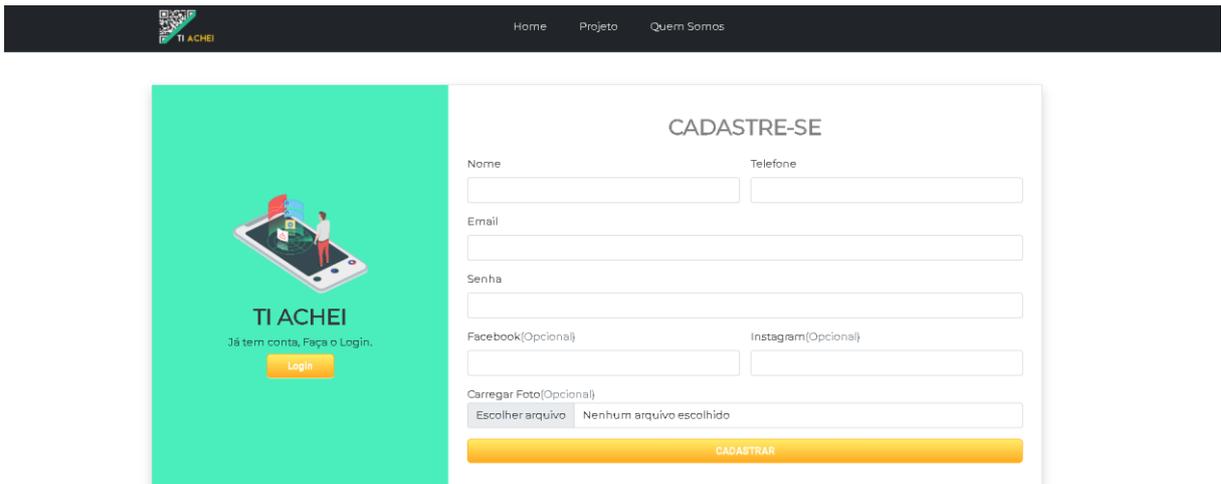
        public async Task<List<Pet>> GetAllPetsUserCurrent(User model)
            => await _dbContext.Pets.Where(c => c.UserId ==
model.Id).ToListAsync();

        public async Task<Pet> GetBydId(string id)
            => await _dbContext.Pets.Where(x => x.Id ==
id).FirstOrDefaultAsync();

```

```
public async Task<bool> Update(Pet newModel)
{
    _dbContext.Pets.Update(newModel);
    if( await _dbContext.SaveChangesAsync() > 0)
    {
        return true;
    }
    else
    {
        return false;
    }
}
}
```

4.8. Prints das Telas Definitivas



The image shows a web application interface for registration. At the top, there is a dark navigation bar with the TI ACHEI logo on the left and links for 'Home', 'Projeto', and 'Quem Somos' on the right. The main content area is split into two sections. On the left, a green sidebar contains an illustration of a smartphone with a person standing next to it, the text 'TI ACHEI', and a 'Login' button. On the right, a white box titled 'CADASTRE-SE' contains several input fields: 'Nome' and 'Telefone' (two-line), 'Email' (one-line), 'Senha' (two-line), 'Facebook(Opcional)' and 'Instagram(Opcional)' (two-line), and 'Carregar Foto(Opcional)' with a file selection button and the text 'Nenhum arquivo escolhido'. A large yellow 'CADASTRAR' button is at the bottom of the white box.

Figura 10 - Tela cadastro

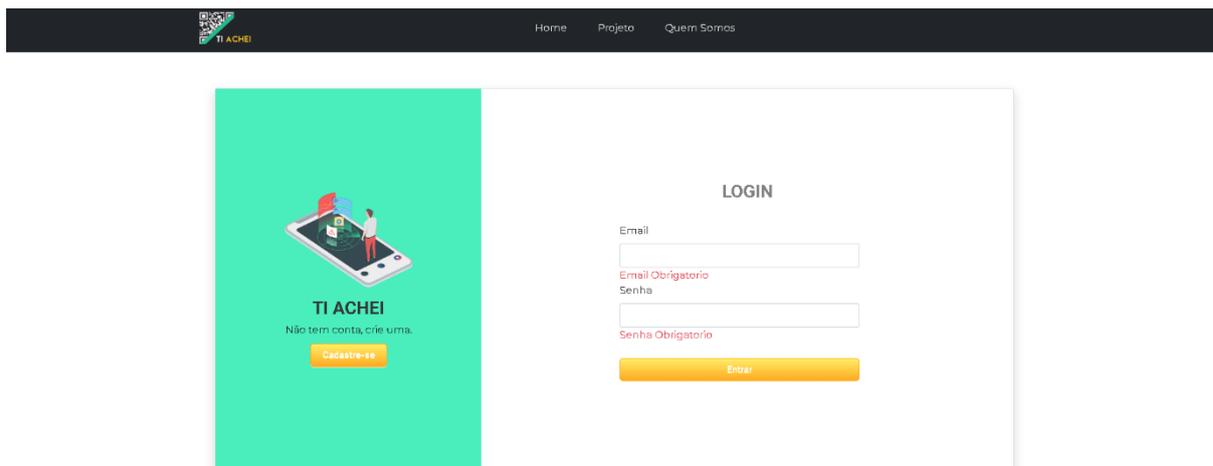


Figura 11 - Tela de Login

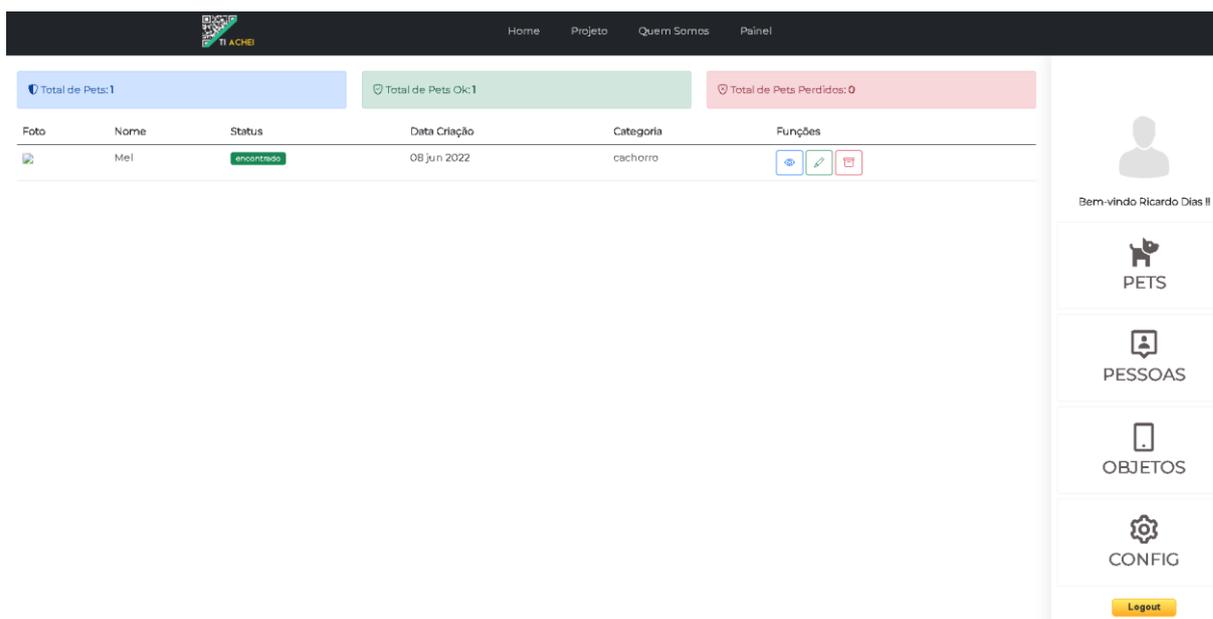


Figura 12 - Painel de Cadastro

Figura 13 - Tela para Adicionar Pets

4.9 Trecho do Código Fonte

```

namespace TiAchei_Tcc.Controllers
{
    [Authorize]
    public class PainelController: Controller
    {
        private readonly IPetRepository _repository;
        private readonly UserManager<User> _userManager;
        private readonly ServiceUploadFile _serviceUpload;
        public PainelController(IPetRepository repository,      UserManager<User>
userManager ,
        ServiceUploadFile serviceUpload)
        {
            _userManager = userManager;
            _repository = repository;
            _serviceUpload = serviceUpload;
        }

        [HttpGet]
        public async Task<IActionResult> Buscar(string id) => Json(await
_repository.GetById(id));

        [HttpGet]
        public async Task<IActionResult> Pessoas() => View(await
_userManager.GetUserAsync(User));

        [HttpGet]
        public async Task<IActionResult> Objetos() => View(await
_userManager.GetUserAsync(User));
    }
}

```

```

        [HttpGet]
        public async Task<IActionResult> CadastrarPet() => View(await
        _userManager.GetUserAsync(User));
        [HttpGet]
        public async Task<IActionResult> CadastrarPessoa() => View(await
        _userManager.GetUserAsync(User));

        [HttpGet]
        public async Task<IActionResult> Config() => View(await
        _userManager.GetUserAsync(User));

        [HttpGet]
        public async Task<IActionResult> Editar(string id) => View(await
        _repository.GetBydId(id));
        [HttpDelete]
        public async Task<IActionResult> Deletar(string id) => await
        _repository.DeletePet(id) ? Ok() : BadRequest();

        [HttpGet]
        public async Task<IActionResult> Index()
            => View(new PaineiViewModel()
            {
                Pets = await _repository.GetAllPetsUserCurrent(await
        _userManager.GetUserAsync(User)),
                Usuario = await _userManager.GetUserAsync(User)
            });

        [HttpPost]
        public async Task<IActionResult> CadastrarPet(PetViewModel model)
        {
            var userCurrent = await _userManager.GetUserAsync(User);

            var petCurrent = new Pet()
            {
                Nome = model.Nome,
                Raca = model.Raca,
                Foto = _serviceUpload.UploadFilePet(model),
                Perdido = false,
                UserId = userCurrent.Id,
                Descricao = model.Descricao
            };
            await _repository.CreatePet(petCurrent);
            this.MostrarMensagem("Pet Cadastrado com sucesso", TipoMensagem.Sucesso);

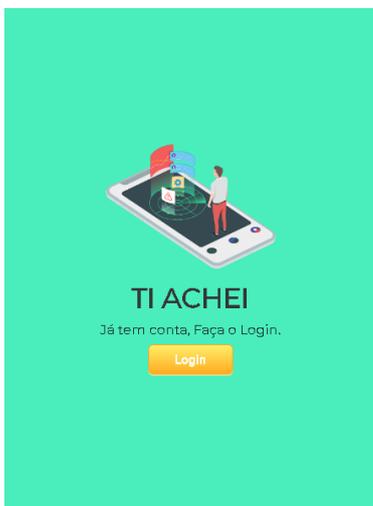
```

```
        return RedirectToAction("CadastrarPet");
    }

    [HttpPost]
    public async Task<IActionResult> Editar([FromForm]Pet model)
    {
        var userCurrent = await _userManager.GetUserAsync(User);
        model.UserId = userCurrent.Id;
        if(!await _repository.Update(model)) return BadRequest();
        return RedirectToAction("Index", "Painel");
    }
}
}
```

5. MANUAL DO USUÁRIO

Para fazer um cadastro basta inserir seu nome, número de telefone que use, um e-mail e criar uma senha, Facebook e Instagram são opcionais, mas se quiser preenchê-los, basta copiar o url do seu perfil de qualquer um dos dois e colar nos campos, na última opção tem o upload de foto, deve ser inserido uma foto do usuário, a qual também é opcional, porém, é importante que tenha, pois, a identificação será mais fácil. Feito isso, é só clicar no botão amarelo “Cadastrar”, como exibido na imagem abaixo.



TI ACHEI
Já tem conta, Faça o Login.
[Login](#)

CADASTRE-SE

Nome Telefone

Email

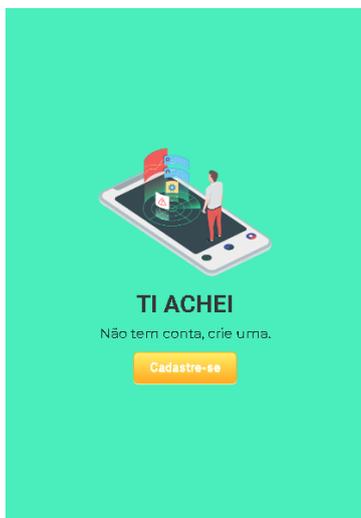
Senha

Facebook(Opcional) Instagram(Opcional)

Carregar Foto(Opcional)
 Nenhum arquivo escolhido

[CADASTRAR](#)

Após esses simples passos, clique no botão “Login” que se encontra na esquerda do formulário, ele irá redirecionar para a página abaixo, é só preencher os campos e-mail e senha cadastrados no passo anterior e clicar no botão “Entrar”, a partir daqui você terá acesso ao sistema para fazer os cadastros de pets.



TI ACHEI
Não tem conta, crie uma.
[Cadastre-se](#)

LOGIN

Email
Email Obrigatorio

Senha
Senha Obrigatorio

[Entrar](#)

Ao entrar na sua conta, a primeira tela a ser exibida, já é o Painel de Cadastro, logo acima vemos o total de cadastros efetuados, o status que se encontram: se está desaparecido ou em segurança. Logo abaixo fica localizado uma tabela com detalhes dos cadastros feitos, contendo foto, nome, status, data da criação, categoria e algumas funções como excluir cadastro, editar e visualizar. Na opção “Editar” que tem um ícone de lápis podemos mudar qualquer informação que tenha atualizado, por exemplo o status, de “seguro” para “desaparecido”. Em “visualizar” podemos baixar o *QR Code* gerado e ver uma prévia de como o cartão de visitas será mostrado ao ler o código. A direita situa-se um painel onde terão as opções de cadastro de pet, pessoa, objeto e opção de modificação do perfil do usuário, ou seja, ele poderá alterar seu número de telefone, sua senha, qualquer informação que tenha mudado.

The screenshot displays a dashboard with a dark header containing the logo and navigation links: Home, Projeto, Quem Somos, and Painel. Below the header, there are three summary cards: 'Total de Pets: 1' (blue), 'Total de Pets Ok: 1' (green), and 'Total de Pets Perdidos: 0' (red). The main content area features a table with the following data:

Foto	Nome	Status	Data Criação	Categoria	Funções
	Mel	encontrado	08 Jun 2022	cachorro	

On the right side, there is a user profile section with the name 'Bem-vindo Ricardo Dias II' and a 'Logout' button. Below this are four main navigation categories: PETS, PESSOAS, OBJETOS, and CONFIG, each with a corresponding icon.

Para realizar um cadastro basta clicar em qualquer um dos itens a direita, pegamos o “Pet” apenas para fazer uma demonstração. Coloque o nome do pet, a raça dele, se é cachorro ou gato, escolha uma foto e o anexe na opção “escolher arquivo” e abaixo coloque uma descrição.

The image shows a web form titled "ADICIONAR PET". It contains the following elements:

- Nome:** A text input field containing "Mel".
- Raca:** A text input field containing "dsa".
- Species:** A dropdown menu with "cachorro" selected.
- File Upload:** A button labeled "Escolher arquivo" and a text area showing "Nenhum arquivo escolhido".
- Description:** A large text area containing "dsada".
- Submit:** A prominent yellow button at the bottom labeled "Adicionar".

Depois é só clicar em “Adicionar” e ele será exibido no painel de cadastro. Clique em visualizar e “Ver QR Code”, ele vai te redirecionar para o cartão de visitas que será exibido ao ler o código. Baixe o *QR Code*, imprima e plastifique-o para uso do dia a dia.

CONCLUSÃO

Esperamos que esse projeto cause impacto positivo na sociedade, que traga pets, crianças e idosos para casa em segurança, de uma forma ágil e eficaz. Pelos *feedbacks* que recebemos no levantamento de campo, vemos resultados positivos em relação ao projeto, esperamos causar uma melhoria em relação a esses desaparecimentos, trazendo segurança e agilidade na localização.

A oportunidade de desenvolver este projeto nos trouxe a vivência de equipe e nos sentimos mais próximos do mercado de trabalho com a possibilidade de construir algo útil à sociedade, mas que também aprimorou nosso aprendizado técnico.

REFERÊNCIA BIBLIOGRÁFICA

ABINPET. Informações gerais do setor Pet. Disponível em: <http://abinpet.org.br/infos_gerais/#:~:text=O%20Brasil%20tem%20a%20segunda,3%20milh%C3%B5es%20de%20outros%20animais.> Acesso em: 30/05/2022

JORNAL SP NORTE. Cães e Gatos perdidos estão entre os 30 milhões de animais vivendo nas ruas do Brasil. Disponível em: <<https://www.jornalspnorte.com.br/caes-e-gatos-perdidos-estao-entre-os-30-milhoes-de-animais-vivendo-nas-ruas-do-brasil/>> Acesso em: 30/05/2022

ASSEMBLEIA LEGISLATIVA DE SERGIPE. Crianças Desaparecidas Disponível em: < <https://al.se.leg.br/saiba-os-motivos-que-levam-criancas-e-adolescentes-a-desaparecerem/#:~:text=N%C3%A3o%20h%C3%A1%20registros%20oficiais%20de,%20n%C3%BAmero%20%C3%A9%20muito%20maior.> > Acesso em: 01/06/2022

G1. Cresce o número de crianças perdidas em praias do litoral de São Paulo Publicado em: 24/01/2019. Disponível em <https://g1.globo.com/sp/santos-regiao/noticia/2019/01/24/cresce-o-numero-de-criancas-perdidas-em-praias-do-litoral-de-sao-paulo.ghtml>> Acesso em: 01/06/2022

SP NOTÍCIAS. Desaparecimento de idosos. Publicado em: 06/05/2017 Disponível em : < <https://www.saopaulo.sp.gov.br/spnoticias/o-que-fazer-quando-pessoas-mais-velhas-nao-voltam-para-casa/> > Acesso em: 01/06/2022

EINSTEIN, Hospital. Alzheimer. Disponível em: < <https://www.einstein.br/guia-doencas-sintomas/info/#5>> Acesso em :03/06/2022

MICROSOFT. Um tour pela linguagem C#. Disponível em:<<https://docs.microsoft.com/pt-br/dotnet/csharp/tour-of-csharp/>> Acesso em: 07/06/22

MICROSOFT. Visão geral de Entity Framework. Disponível em: <
<https://docs.microsoft.com/pt-br/dotnet/framework/data/adonet/ef/overview> > Acesso
em: 07/06/2022

DEVMEDIA. Introdução ao MySQL . Disponível em: <
<https://www.devmedia.com.br/introducao-ao-mysql/27799> > Acesso em: 27/06/22

MICROSOFT. Visão geral do Azure para desenvolvedores. Disponível em: <
<https://docs.microsoft.com/pt-BR/azure/developer/intro/azure-developer-overview>>
Acesso em: 07/06/22

APÊNDICES

(Opcional – só use se seu trabalho necessitar – Apêndice são documentos ou conteúdos **produzidos pelos autores do trabalho**, e foram citados no corpo do trabalho)

ANEXOS

(Opcional – só use se for necessário no seu trabalho. Anexos são documentos ou conteúdos **não produzidos pelos autores do trabalho**, mas que comprovam conteúdos descritos no corpo de trabalho e foram citados no corpo do trabalho)