

**CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA  
SOUZA**

**ETEC DA ZONA LESTE  
M-Tec Desenvolvimento De Sistemas**

**Jônatas Frinhani de Souza Palmieri  
Vinicius Augusto Rodrigues da Silva  
Vinicius Fernandes de Lima**

**T-Fence: Sistema de Rastreamento de Animais em Áreas Rurais**

**São Paulo**

**2025**

**Jônatas Frinhani de Souza Palmieri  
Vinicius Augusto Rodrigues da Silva  
Vinicius Fernandes de Lima**

**T-Fence: Sistema de Rastreamento de Animais em Áreas Rurais**

Trabalho de Conclusão de Curso  
apresentado ao M-Tec Desenvolvimento  
de Sistemas da Etec Zona Leste, orientado  
pelo Prof. Esp. Jeferson Roberto de Lima,  
como requisito parcial para a obtenção do  
título de técnico em Desenvolvimento de  
Sistemas

**São Paulo**

**2025**

## **T-Fence**

Sistema de Rastreamento de Animais em Áreas Rurais.

**Jônatas Frinhani de Souza Palmieri**  
**Vinicius Augusto Rodrigues da Silva**  
**Vinicius Fernandes de Lima**

Aprovada em \_\_\_\_/\_\_\_\_/\_\_\_\_.

### **BANCA EXAMINADORA**

---

Prof. Esp. Jeferson Roberto de Lima  
Universidade do Jeferson

---

Prof. (Professor avaliador)  
Universidade do Avaliador

---

Prof. (Professor avaliador)  
Universidade do Avaliador

## **AGRADECIMENTOS**

Após o encerramento deste Trabalho de Conclusão de Curso (TCC), pode-se concluir que o projeto representou um ponto importante na jornada de aprimoramento acadêmico e pessoal para todos aqueles que participaram de seu desenvolvimento. Ao final do projeto, todos os integrantes adquiriam conhecimento como consequência de todo o esforço e dedicação aplicados durante todo o processo. É necessário reconhecer que para os resultados obtidos, é essencial compartilhar essa conquista com aqueles que participaram durante todo esse percurso.

Primeiramente agradecemos a Deus por nos dar forças e sabedoria para lidar com as dificuldades e obstáculos que encontramos pelo caminho.

É importante mencionar nossas famílias que tanto amamos por seus conselhos, incentivos e pelo auxílio durante os momentos mais difíceis, e por seu amor incondicional.

Agradecimentos também são válidos aos nossos amigos que sempre estiveram do nosso lado nos apoiando e que não mediram esforços para nos ajudar quando mais precisávamos, além dos professores e coordenadores que compartilharam todo seu conhecimento, permitindo o desenvolvimento de um senso crítico para a validação de nosso trabalho.

Eterna gratidão a todos que auxiliaram de alguma forma!

"as vezes continuar, apenas continuar, é a conquista sobre-humana."

**Albert Camus**

## **RESUMO**

O projeto tem como objeto o desenvolvimento de um sistema IoT eficiente para monitoramento de gado em áreas rurais, focando em aumentar o controle dos produtores sobre seus animais e dando maior segurança e eficiência na administração do rebanho. A solução inclui o uso de dispositivos capazes de acompanhar a movimentação dos animais e delimitar um perímetro virtual, ajudando em diversas situações e ambientes. O sistema é constituído por um dispositivo acoplado ao animal, alimentado por energia fotovoltaica, que, por meio da comunicação LoRa, envia sua localização para um aplicativo móvel que possibilita ao produtor visualizar um mapa atualizado com a posição do rebanho além de fornecer outras informações úteis para o monitoramento. A tecnologia de Geofencing possibilita delimitar áreas e detectar quando um animal sai do perímetro estabelecido. A metodologia envolve a realização de testes em durante o desenvolvimento do dispositivo, junto de pesquisas quantitativas que nos auxiliam a entender melhor as dificuldades enfrentadas na gestão de rebanhos e a falta de informatização nas propriedades rurais. A problemática trata da falta de tecnologia nas áreas rurais e das dificuldades na gestão de rebanhos extensos. Conclui-se que a proposta oferece uma solução acessível, eficiente e adaptável, contribuindo para a transformação digital no campo, beneficiando incluem redução de perdas, maior controle do rebanho e melhora na segurança patrimonial das propriedades rurais.

**Palavras-Chave:** Internet das Coisas; Monitoramento de Gado; LoRa; Geofencing; Energia Fotovoltaica.

## **ABSTRACT**

The project aims to develop an efficient IoT system for monitoring livestock in rural areas, focusing on increasing producers' control over their animals and providing greater security and efficiency in herd management. The solution includes the use of devices capable of tracking the movement of animals and delimiting a virtual perimeter, helping in various situations and environments. The system consists of a device attached to the animal, powered by photovoltaic energy, which, through LoRa communication, sends its location to a mobile application that allows the producer to view an updated map with the position of the herd, in addition to providing other useful information for monitoring. Geofencing technology makes it possible to delimit areas and detect when an animal leaves the established perimeter. The methodology involves conducting tests during the development of the device, along with quantitative research that helps us better understand the difficulties faced in herd management and the lack of computerization on rural properties. The problem concerns the lack of technology in rural areas and the difficulties in managing large herds. It is concluded that the proposal offers an accessible, efficient, and adaptable solution, contributing to digital transformation in the field, with benefits including reduced losses, greater herd control, and improved property security on rural properties.

**Keywords:** Internet of Things; Livestock Monitoring; LoRa; Geofencing; Photovoltaic Energy.

## **LISTA DE ILUSTRAÇÕES**

Figura 1 – Exemplo de código Javascript.....	18
Figura 2 – Resultado da codificação JavaScript.....	19
Figura 3 – Exemplo de código React Native .....	20
Figura 4 – Exemplo de código de estilização com React Native .....	20
Figura 5 – Exemplo de aplicação React Native.....	21
Figura 6 – Exemplo de C++ utilizado no ESP32 .....	23
Figura 7 – Circuito ESP32 com LED .....	25
Figura 8 – Imagem Geofencing .....	26
Figura 9 – Exemplo de wireframes de baixa fidelidade .....	27
Figura 10 – Exemplo de wireframes de alta fidelidade .....	28
Figura 11 – Exemplo de modelagem 3D T-Fence .....	29
Figura 12 – Exemplificação de um diagrama de caso de uso .....	30
Figura 13 – Exemplificação de um diagrama de caso de uso .....	31
Figura 14 – Exemplificação de um diagrama de atividade .....	32
Figura 15 – Exemplo de diagrama de Sequência.....	32
Figura 16 – Exemplo de diagrama de Máquina de Estados .....	33
Figura 17 – Imagem do ESP32 .....	34
Figura 18 – Módulo Transmissor LoRa SX1276.....	34
Figura 19 - Módulo controlador de carga TP4057 .....	35
Figura 20 – Mini painel solar .....	36
Figura 21 – Módulo GPS Neo-M8N.....	36
Figura 22 – Bateria de lítio-polímero .....	37
Figura 23 – Imagem do MCU-219 .....	38
Figura 24 - Diagrama de Caso de Uso: T-Fence.....	38
Figura 25 - Diagrama de Caso de Uso: T-Fence.....	39

Figura 26 – Diagrama de Atividade: Cadastrar Conta.....	59
Figura 27 – Diagrama de Atividade: Receber Notificação .....	59
Figura 28 – Diagrama de Atividade: Visualizar Mapa.....	60
Figura 29 – Diagrama de Atividade: Manter Cerca Virtual .....	60
Figura 30 – Diagrama de Atividade: Manter Animais .....	61
Figura 31 – Diagrama de Atividade: Manter propriedade .....	61
Figura 32 – Diagrama de Atividade: Manter Colaborador .....	62
Figura 33 – Diagrama de Atividade: Manter Coleiras .....	62
Figura 34 – Diagrama de Atividade: Realizar Login .....	63
Figura 35 – Diagrama de Atividade: Informar bateria.....	63
Figura 36 – Diagrama de Atividade: Identificar Fuga .....	64
Figura 37 – Diagrama de Atividade: Enviar Localização .....	64
Figura 38 – Diagrama de Atividade: Gerar Notificações .....	65
Figura 39 – Diagrama de Atividade: Verificar Conexão.....	65
Figura 40 – Diagrama de Atividade: Gerenciar Perfil .....	66
Figura 41 – Diagrama de Atividade: Emparelhar Central .....	66
Figura 42 – Diagrama de Sequência: Cadastrar Conta.....	67
Figura 43 – Diagrama de Sequência: Receber Notificação.....	68
Figura 44 – Diagrama de Sequência: Gerenciar Perfil .....	68
Figura 45 – Diagrama de Sequência: Visualizar Mapa .....	69
Figura 46 – Diagrama de Sequência: Manter Propriedade .....	70
Figura 47 – Diagrama de Sequência: Manter Coleiras .....	71
Figura 48 – Diagrama de Sequência: Manter Colaboradores .....	71
Figura 49 – Diagrama de Sequência: Manter Cerca Virtual .....	72
Figura 50 – Diagrama de Sequência: Manter Animais .....	73
Figura 51 – Diagrama de Sequência: Realizar Login .....	74

Figura 52 – Diagrama de Sequência: Informar Bateria .....	75
Figura 53 – Diagrama de Sequência: Identificar Fuga .....	75
Figura 54 – Diagrama de Sequência: Gerar Notificação .....	76
Figura 55 – Diagrama de Sequência: Enviar Localização .....	76
Figura 56 – Diagrama de Sequência: Verificar Conexão .....	77
Figura 57 – Diagrama de Sequência: Emparelhar Central .....	77
Figura 58 – Diagrama de Máquina de Estado: Realizar Login .....	78
Figura 59 – Diagrama de Máquina de Estado: Emparelhar Central .....	79
Figura 60 – Protótipo Final Coleira T-Fence.....	80
Figura 61 – Protótipo Final Central Receptora .....	80
Figura 62 – Circuito da Coleira.....	82
Figura 63 – Circuito da Central.....	82
Figura 64 – Wireframes de baixa e alta fidelidade do Cadastro .....	83
Figura 65 – Wireframes de baixa e alta fidelidade da segunda etapa do Cadastro ..	84
Figura 66 – Wireframes de baixa e alta fidelidade da terceira etapa do Cadastro ....	84
Figura 67 – Wireframes de baixa e alta fidelidade da quarta etapa do Cadastro .....	85
Figura 68 – Wireframes de baixa e alta fidelidade da quinta etapa do Cadastro .....	86
Figura 69 – Wireframes de baixa e alta fidelidade da sexta etapa do Cadastro.....	86
Figura 70 – Wireframes de baixa e alta fidelidade da sétima etapa do Cadastro.....	87
Figura 71 – Wireframes de baixa e alta fidelidade da sétima etapa do Cadastro.....	87
Figura 72 – Wireframes de baixa e alta fidelidade da última etapa do Cadastro.....	88
Figura 73 – Wireframes de baixa e alta fidelidade do Login .....	88
Figura 74 – Wireframes de baixa e alta fidelidade do Mapa.....	89
Figura 75 – Wireframes de baixa e alta fidelidade das notificações .....	89
Figura 76 – Wireframes de baixa e alta fidelidade da propriedade .....	90
Figura 77 – Wireframes de baixa e alta fidelidade de monitoramento.....	90

Figura 78 – Wireframes de baixa e alta fidelidade de Cercas ..... 91

Figura 79 – Wireframes de baixa e alta fidelidade do Perfil ..... 91

## **LISTA DE TABELAS**

Tabela 1 - Documentação do caso de uso: Cadastrar Conta .....	42
Tabela 2 - Documentação do caso de uso: Manter Propriedade .....	43
Tabela 3 - Documentação do caso de uso: Realizar Login .....	44
Tabela 4 - Documentação do caso de uso: Emparelhar Central .....	45
Tabela 5 - Documentação do caso de uso: Manter Coleiras .....	46
Tabela 6 - Documentação do caso de uso: Manter Animais .....	47
Tabela 7 - Documentação do caso de uso: Manter Cerca Virtual .....	49
Tabela 8 - Documentação do caso de uso: Manter Colaboradores .....	50
Tabela 9 - Documentação do caso de uso: Enviar Localização .....	51
Tabela 10 - Documentação do caso de uso: Informar Bateria .....	52
Tabela 11 - Documentação do caso de uso: Identificar Fuga .....	54
Tabela 12 - Documentação do caso de uso: Gerar Notificações .....	55
Tabela 13 - Documentação do caso de uso: Visualizar Mapa.....	55
Tabela 14 - Documentação do caso de uso: Gerenciar Perfil .....	56
Tabela 15 - Documentação do caso de uso: Receber Notificação.....	57
Tabela 16 - Documentação do caso de uso: Verificar Conexão.....	58

## **LISTA DE ABREVIATURAS E SIGLAS**

*Application Programming Interface (API)*

*Centro de Apoio Operacional de Combate aos Crimes Contra o Agronegócio  
(CAOAGRO)*

*Centro de Estudos Avançados em Economia Aplicada (CEPEA)*

*Cascading Style Sheet (CSS)*

*Direct Current (Corrente Contínua) (DC)*

*Global Navigation Satellite System (GLONASS)*

*Global Navigation Satellite System (GNSS)*

*Global Positioning System (GPS)*

*HyperText Markup Language (HTML)*

*Input/Output (I/O)*

*Internet of Things (IoT)*

*JavaScript (JS)*

*JavaScript Object Notation (JSON)*

*Light Emitting Diode (LED)*

*Lítio-Polímero (Li-Po)*

*Long Range (LoRa)*

*Ministério da Agricultura e Pecuária (MAPA)*

*Node Package Manager (NPM)*

*Not Only SQL (NoSQL)*

*Produto Interno Bruto (PIB)*

*Random Access Memory (RAM)*

*Requisitos Funcionais (RF)*

*Regras de Negócio (RN)*

*Requisitos Não Funcionais (RFN)*

*Sistema de Gerenciamento de Banco de Dados (SGBD)*

*Structured Query Language (SQL)*

*Trabalho de Conclusão de Curso (TCC)*

*Unified Modeling Language (UML)*

*Universal Serial Bus (USB)*

*User Experience (UX)*

*Wireless Fidelity (Wifi-Fi)*

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO.....</b>	<b>14</b>
<b>2</b>	<b>REFERENCIAL TEÓRICO .....</b>	<b>17</b>
2.1	Segurança e Informatização nos Campos Rurais .....	17
2.2	Tecnologias Utilizadas .....	18
2.2.1	JavaScript .....	18
2.2.2	React Native .....	19
2.2.2.1	EXPO .....	22
2.2.3	C++ .....	22
2.2.4	Node.js e NPM.....	25
2.2.5	Google Maps API .....	25
2.2.6	Geofencing.....	26
2.2.7	MongoDB .....	26
2.2.8	Wireframes.....	26
2.2.8.1	Wireframes baixa fidelidade .....	27
2.2.8.2	Wireframes alta fidelidade .....	27
2.2.9	UX (User Experience) .....	28
2.2.10	LoRa .....	29
2.2.11	Modelagem 3D.....	29
2.2.12	UML (Unified Modeling Language).....	29
2.2.12.1	UML- Diagrama de Caso de Uso.....	30
2.2.12.2	UML- Documentação de Caso de Uso .....	30
2.2.12.3	UML- Diagrama de Atividade.....	31
2.2.12.4	UML- Diagrama de Sequência .....	32
2.2.12.5	UML- Diagrama de Máquina de Estado .....	33
2.2.13	IoT.....	33

2.2.13.1	ESP32 .....	33
2.2.13.2	Módulo LoRa .....	34
2.2.13.3	Módulo controlador de carga .....	35
2.2.13.4	Mini painel solar.....	35
2.2.13.5	Módulo GPS Neo-M8N.....	36
2.2.13.6	Bateria Li-Po.....	37
2.2.13.7	Módulo MCU-219 .....	37
2.2.13.8	Módulo CN3065.....	38
<b>3</b>	<b>DESENVOLVIMENTO .....</b>	<b>39</b>
3.1	Diagrama de Casos de Uso .....	39
3.2	Documentação Caso de Uso .....	39
3.3	Diagrama de Atividade.....	59
3.4	Diagrama de Sequência.....	67
3.5	Diagrama de Máquina de Estados .....	78
3.6	Impressão 3D.....	79
3.7	Montagem do circuito dos dispositivos IoT .....	80
3.8	Prototipação das telas.....	83
<b>4</b>	<b>CONSIDERAÇÕES FINAIS.....</b>	<b>92</b>
	<b>REFERÊNCIAS.....</b>	<b>93</b>

## 1 INTRODUÇÃO

A T-Fence é um sistema de monitoramento de animais voltado para áreas rurais, que busca auxiliar o pecuarista a gerir seu rebanho com mais eficiência e segurança. O sistema é composto por uma coleira IoT que envia a localização do animal para uma aplicação mobile que possibilita o pecuarista acompanhar e controlar seu rebanho de forma prática e precisa.

O objetivo geral desse projeto é desenvolver um sistema de rastreamento para monitoramento de gado em áreas rurais auxiliando pecuaristas na gestão de seus animais, proporcionando mais eficiência, controle e segurança. Isso contribui para a modernização e inovação tecnológica no setor agropecuário.

A agricultura tem uma forte relevância quando se trata sobre o crescimento econômico do país. O Centro de Estudos Avançados em Economia Aplicada (CEPEA, 2025), abordou que em 2025, o agronegócio pode representar cerca de 29,4% do PIB nacional. Ainda assim, o setor agropecuário sofre com a alta taxa de criminalidade na região, o que afeta negativamente a produção agrícola e pecuária. Isso evidencia a necessidade de investimento tecnológico no setor, tendo em vista a sua importância no panorama nacional. Nos últimos anos, furtos e perdas de animais têm gerado grandes prejuízos no meio rural, afetando a produtividade e rentabilidade do agronegócio no Brasil. Segundo os dados divulgados pela Polícia Civil de Santa Catarina (2024), foram registrados pelo Centro de Apoio Operacional de Combate aos Crimes Contra o Agronegócio (CAOAGRO), 327 boletins de ocorrência associados a criminalidade em áreas rurais, sendo que o abigeato (furto de animais bovinos) representou 28,4% de todos os boletins de ocorrências, totalizando 273 bovinos furtados, um aumento de 13,7% comparado ao primeiro quadrimestre de 2023. Esses dados, apresentam a dificuldade que os pecuaristas enfrentam diariamente, em relação a gestão de suas propriedades e de seus animais. Esses crimes podem acabar desestimulando a produção rural, já que geram grandes prejuízos ao produtor rural.

De início, acreditava-se que o problema principal era a ausência de ação do poder público, em tomar medidas contra esse tipo de criminalidade. Outra hipótese levantada era a negligência dos produtores rurais, aliado a limitação da infraestrutura nesses espaços, justificando as tais ocorrências.

Os principais objetivos foram desenvolver um dispositivo capaz de capturar a sua localização GPS e transmitir sua posição utilizando o LoRa para uma central, projetar um sistema com baixo consumo de energia e de longa duração capaz de se auto alimentar utilizando tecnologia fotovoltaica, utilizar a tecnologia de Geofencing para que seja possível a criação de um perímetro virtual, desenvolver um sistema capaz de identificar se o animal saiu do perímetro delimitado e gerar um alerta, integrar o dispositivo com uma aplicação para exibição das informações por meio de um mapa.

Em razão da grande extensão territorial das propriedades rurais, além da pobre infraestrutura tecnológica, o monitoramento dos animais se torna difícil, prejudicando a resolução desta situação. Com isso, foi pensada uma solução energeticamente independente e adaptável em cenários extremos, como áreas extensas e remotas, com pouca disponibilidade de energia elétrica e internet escassa. Com isso, a proposta consiste no desenvolvimento de um sistema de monitoramento via GPS, capaz de aumentar a segurança e otimizar a gestão do gado.

Durante a concepção de nosso projeto, utilizamos uma metodologia de abordagem quantitativa para coletar dados e tirar conclusões. Como afirma Marconi e Lakatos (2017), esse método tem como objetivo quantificar resultados e tratá-los de forma estatística, permitindo uma análise com mais precisão, além de possibilitar identificar padrões no cenário estudado. Assim, foi possível a partir da metodologia quantitativa reconhecer as limitações enfrentadas durante o manejo do gado no ambiente rural, resultando no desenvolvimento de uma solução que se adequasse com as necessidades do público-alvo.

Nesta pesquisa serão utilizados os principais autores e fontes relevantes para o trabalho. Para trazer dados sobre a relevância da agropecuária no país e sua importância na economia nacional, foram consultados Ministério da Agricultura e Pecuária (MAPA, 2023) e CEPEA (2025). Já para o desenvolvimento do nosso IoT, utilizamos os seguintes autores: Magrani (2018). Para o desenvolvimento do nosso aplicativo, utilizamos Escudelario e Pinho (2021) e Falcão (2022), bem como a documentação oficial do React Native (REACT NATIVE, 2025).

Este documento se divide em 4 capítulos que auxiliam na compreensão do desenvolvimento do projeto: no capítulo 1 é abordado uma introdução sobre o tema, no capítulo 2 é apresentado o referencial teórico com as ferramentas e tecnologias

utilizadas, no capítulo 3 é descrito o desenvolvimento do trabalho e por fim as considerações finais sobre o projeto está contido no capítulo 4.

## 2 REFERENCIAL TEÓRICO

Neste capítulo, serão apresentados os principais fundamentos teóricos e as tecnologias utilizadas no desenvolvimento do nosso projeto.

### 2.1 Segurança e Informatização nos Campos Rurais

O setor agrário tem uma grande importância na economia brasileira, possuindo um papel fundamental no desenvolvimento do país. O Ministério da Agricultura e Pecuária (MAPA, 2023), afirma que em 2023, a agropecuária cresceu 15,1%, refletindo diretamente no desenvolvimento do Produto Interno Bruto (PIB) do país, aumentando 2,9% em relação a 2022. Entretanto, mesmo sendo um dos setores fundamentais para a economia do Brasil, a agropecuária enfrenta grandes problemas em relação à segurança de suas propriedades, o que pode facilitar a ocorrência de crimes contra o setor.

Segundo Oliveira, Medina e Teixeira (2021), cerca de 90% dos crimes cometidos em áreas rurais de Goiás nos anos de 2017 e 2018 foram classificados como furtos, reconhecidos juridicamente como abigeatos, tendo como um dos principais alvos os animais bovinos. De acordo com a Confederação da Agricultura e Pecuária do Brasil (2017), os crimes que ocorrem nas propriedades rurais, muitas vezes são enquadrados como conflitos agrários, apresentando os produtores rurais como causadores dos problemas e não como vítimas dessa situação. Levando em consideração, a importância do setor no cenário nacional e as ocorrências apresentadas anteriormente, esses fatos podem acabar desestimulando a população agrícola em suas atividades, impactando diretamente na economia nacional.

Outro fator que deve ser levado em consideração é a dificuldade na aplicação de soluções tecnológicas que trazem a informatização para o campo, tendo em vista a infraestrutura pobre da região. Hoje em dia o cenário rural demonstra uma demanda por soluções inovadoras, tendo em vista essa necessidade, a Embrapa (2021), apontou que um proprietário rural de Minas Gerais, gasta R\$78 mil por ano para realizar a contagem dos animais em seu rebanho de forma manual. Isso nos mostra a necessidade de soluções mais eficientes e rentáveis.

Em resumo, mesmo sendo fundamental para a economia do país, o setor agropecuário continua enfrentando problemas relacionados à falta de segurança e à escassez de tecnologias, o que pode dificultar o processo produtivo. A implementação

de novas tecnologias no setor primário, adequadas à realidade do meio rural, pode melhorar a eficiência e a qualidade da produção, além de impulsionar a economia agropecuária e a economia nacional como um todo.

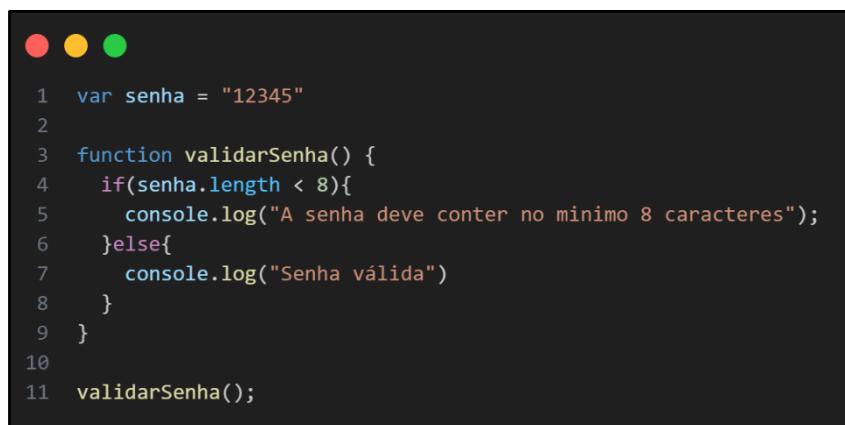
## 2.2 Tecnologias Utilizadas

A seguir veremos as tecnologias envolvidas no desenvolvimento e prototipagem do nosso trabalho.

### 2.2.1 JavaScript

Criado pela Netscape e licenciado pela atual Oracle, o JavaScript, além de uma marca, é uma linguagem web que tem como principal função especificar o comportamento das páginas (FLANAGAN, 2013). É considerada uma das linguagens mais usadas no mundo inteiro, pois pode ser usada no front-end e no back-end (GRONER, 2018). De acordo com Silva (2010), o Javascript é capaz de acessar campos em um formulário feito em HTML e validá-los. Veja abaixo um exemplo simples de validação com a explicação do código:

Figura 1 – Exemplo de código Javascript



```

1  var senha = "12345"
2
3  function validarSenha() {
4      if(senha.length < 8){
5          console.log("A senha deve conter no minimo 8 caracteres");
6      }else{
7          console.log("Senha válida")
8      }
9  }
10
11 validarSenha();

```

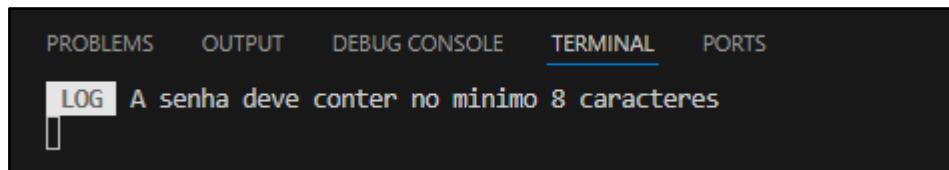
Fonte: De Autoria Própria, 2025.

- Var: “var” é usado para declarar uma variável no JavaScript, permitindo guardar valores.
- Function: “function validarSenha()” é usado para definir uma função que pode agrupar instruções que só serão executadas quando a função for chamada.
- If/else: “if/else” é usado para criar condições, verificando se uma regra é verdadeira ou não.

- Console.log: “console.log()” é usado para enviar uma mensagem para o console, permitindo que o usuário visualize avisos ou resultados durante a execução do código.
- Validar Senha: “validarSenha()” é usado para executar a função definida anteriormente.
- Length: “senha.length” é usado para trazer a quantidade de caracteres dentro da variável senha.

O Resultado aparecerá assim:

Figura 2 – Resultado da codificação JavaScript



Fonte: De Autoria Própria, 2025.

### 2.2.2 React Native

Segundo Escudelario e Pinho (2021), o React Native é um framework, baseada no React e criada por engenheiros do Facebook, no qual busca trazer modernidade aos aplicativos mobiles, utilizando as melhores ferramentas do mercado para o desenvolvimento front-end (HTML, CSS e JS).

O React Native permite a criação de aplicativos nativos, sem a quebra de experiência dos seus usuários, trazendo a programação em React, para plataformas como Android e iOS (React Native).

Como observa Escudelario e Pinho (2021), a capacidade de transformar código JavaScript em código nativo das plataformas, é o principal atrativo do React Native. Isso facilita no desenvolvimento, já que uma única aplicação pode ser utilizada em ambas as plataformas, reduzindo custos e oferecendo uma melhor experiência.

Para maior clareza, segue abaixo um exemplo do código com as funcionalidades explicadas:

Figura 3 – Exemplo de código React Native

```
import React from 'react';
import { View, Text, TouchableOpacity, StyleSheet, Image } from 'react-native';

const WelcomeScreen = () => {
  return (
    <View style={styles.container}>
      <Image
        source={require('../assets/cadeado.png')}
        style={styles.image}
      />
      <Text style={styles.title}>Bem-vindo!</Text>
      <Text style={styles.subtitle}>
        Comece sua jornada na fazenda digital com segurança e praticidade.
      </Text>
      <TouchableOpacity style={styles.button}>
        <Text style={styles.buttonText}>Começar</Text>
      </TouchableOpacity>
    </View>
  );
};

export default WelcomeScreen;
```

Fonte: De Autoria Própria, 2025.

Figura 4 – Exemplo de código de estilização com React Native

```
const styles = StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: 'center',
    alignItems: 'center',
    backgroundColor: '#fff',
    padding: 20,
  },
  image: {
    width: 80,
    height: 80,
    marginBottom: 15,
  },
  title: {
    fontSize: 28,
    fontWeight: 'bold',
    color: '#156C1E',
  },
  subtitle: {
    fontSize: 14,
    color: '#555',
    textAlign: 'center',
    marginVertical: 15,
  },
  button: [
    {
      backgroundColor: '#156C1E',
      paddingVertical: 12,
      paddingHorizontal: 40,
      borderRadius: 5,
    },
  ],
  buttonText: {
    color: '#fff',
    fontWeight: 'bold',
    fontSize: 16,
  },
});

export default WelcomeScreen;
```

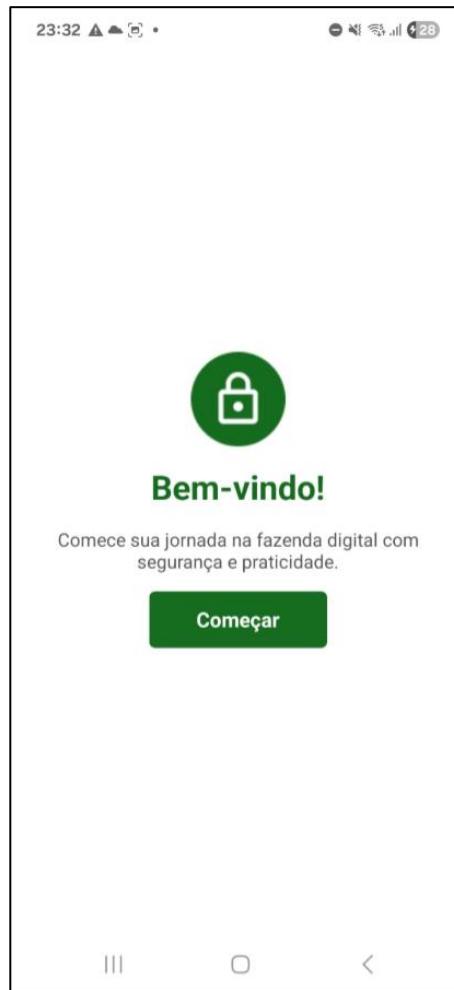
Fonte: De Autoria Própria, 2025.

- Const: “const WelcomeScreen = () => {}” cria um componente funcional e o coloca em uma constante.
- Return: “return ()” é usado para devolver o estilo do componente exibido na tela.
- Import: “import {} from 'react-native'” usado para inserir os componentes do React Native usados para montar a tela.-

- View: “<View style={}>” tem como função criar um contêiner para auxiliar na organização dos elementos na tela.
- Image: “<Image source={require("")} style={} />” usado para mostrar a imagem definida pelo caminho.
- Text: “<Text style={}></Text>” é usado para inserir um texto na tela.
- TouchableOpacity: “<TouchableOpacity style={}” tem a função de acionar o botão quando clicado.
- StyleSheet.create(): “const styles = StyleSheet.create({})” é usado para definir os estilos, como o tamanho e cores.
- Export default: “export default WelcomeScreen” faz o componente ser importado na tela.

O código acima ficará da seguinte forma:

Figura 5 – Exemplo de aplicação React Native



Fonte: De Autoria Própria, 2025.

### **2.2.2.1 EXPO**

Expo é um framework de código aberto utilizado para facilitar o desenvolvimento de aplicações multiplataforma com React Native, oferecendo algumas funcionalidades como o roteamento baseado em arquivos e biblioteca padrão de módulos nativos (Expo).

Para Escudelario e Pinho (2021), o Expo é um conjunto de ferramentas que fornece todo o ambiente necessário para desenvolver as aplicações, possibilitando criar, testar e rodar o código de uma maneira fácil e simples.

De acordo com Falcão (2022), o Expo busca diminuir a complexidade do código, simplificando o processo de construção dos aplicativos, ocultando etapas mais complexas, permitindo assim que o desenvolvedor foque apenas no código JavaScript do ambiente de desenvolvimento.

### **2.2.3 C++**

A linguagem de programação C++ foi criada na década de 1980 pelo Bjarne Stroustrup, nos laboratórios da Bell Labs, com o objetivo de ampliar recursos da linguagem C, adicionando conceitos de programação orientada a objetos (DEITEL; DEITEL, 2005).

De acordo com Villas-Boas (2001), o C++ é uma linguagem muito versátil que pode ser utilizada para desenvolvimento em diversos ambientes e plataformas, desde sistemas operacionais de baixo nível até sistemas embarcados.

Figura 6 – Exemplo de C++ utilizado no ESP32



```
1 #define LED_PIN 2
2
3 void setup() {
4     pinMode(LED_PIN, OUTPUT);
5 }
6
7 void loop() {
8
9     digitalWrite(LED_PIN, HIGH);
10    delay(1000);
11
12    digitalWrite(LED_PIN, LOW);
13    delay(1000);
14
15 }
```

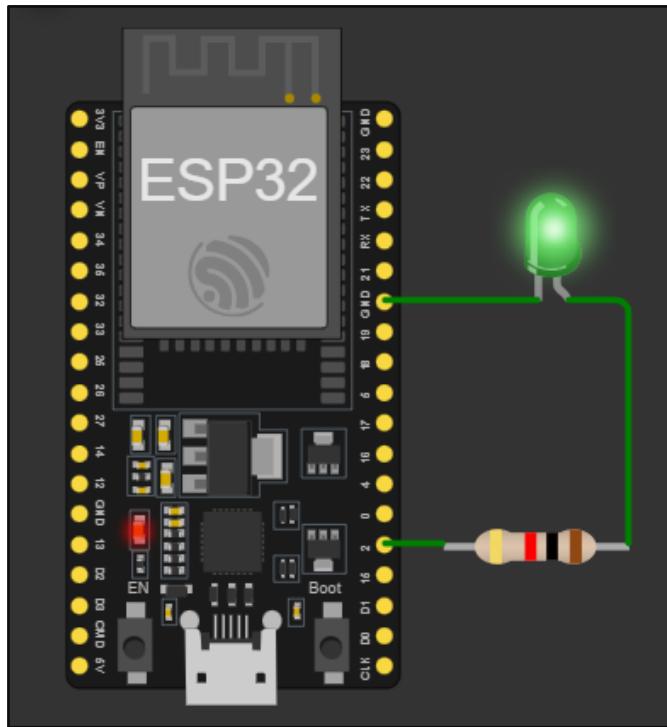
Fonte: De Autoria Própria, 2025.

Explicação do código:

- `#define LED_PIN 2`: É uma diretiva do pré-processador em C/C++ ela cria um nome simbólico para um valor constante. Antes do código ser enviado para o ESP32, o pré-processador troca todas as aparições de `LED_PIN` por 2.
- `void setup()`: É uma função que é executada uma única vez quando o ESP32 ligar ou reiniciar, utilizada para definir modo de um pino, iniciar sensores, displays etc.
- `void loop()`: É uma função que sempre se repete, tudo que estiver dentro desta função roda em ciclo infinito enquanto o ESP32 estiver ligado.
- `digitalWrite()`: É uma função que envia um valor alto ou baixo para um pino digital.
- `digitalWrite(LED_PIN, HIGH)`: `LED_PIN` é o pino 2 (Definido com `#define`), `HIGH` manda o valor alto (3.3V), ou seja, acende o LED.
- `delay(1000)`: É uma função que pausa o programa por um tempo, neste caso em 1000 milissegundos que é equivalente à 1 segundo, serve para deixar o LED aceso por 1 segundo.
- `digitalWrite(LED_PIN, LOW)`: `LOW` manda um valor baixo (0V) para o pino 2, desligando o LED.
- `delay(1000)`: Espera mais 1 segundo com o LED apagado.
- O processo se repete enquanto o ESP estiver ligado.

Abaixo uma imagem que demonstra o código em execução:

Figura 7 – Circuito ESP32 com LED



Fonte: De Autoria Própria, 2025.

#### 2.2.4 Node.js e NPM

Lançado no final de 2009 e criado pelo Ryan Dahl e sua equipe, o Node.js surgiu devido a ineficiência da arquitetura clássica, que gastava grande parte do tempo em uma fila ociosa enquanto era usado para executar um I/O (PEREIRA, 2014).

Segundo Aguiar (2015), o Node.js é uma plataforma não bloqueante orientada a eventos, usada para desenvolver aplicações de redes escaláveis, resolvendo o problema da antiga arquitetura com o grande volume de dados.

O NPM é um gerenciador de pacotes que tem como função principal a instalação de pacotes, e já vem instalado junto do node (BROWN, 2020).

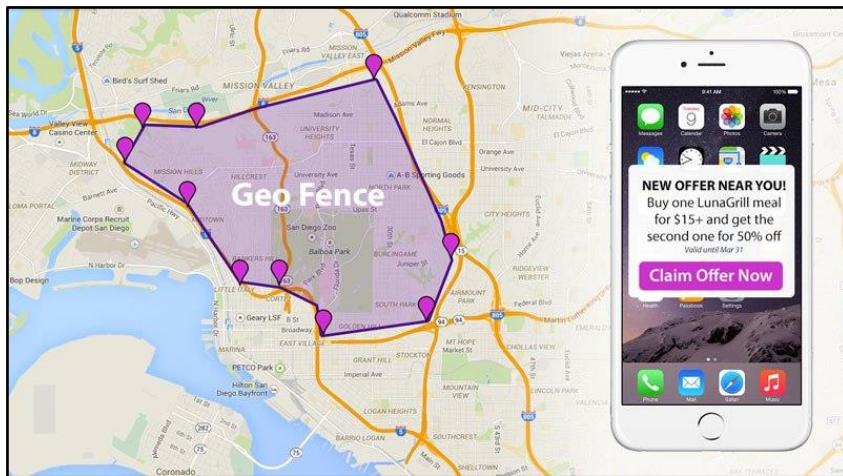
#### 2.2.5 Google Maps API

A API Maps JavaScript é um serviço do Google que possibilita a utilização e personalização de mapas interativos em aplicações web. Com ela é possível exibir localizações, criar marcadores personalizados e até traçar linhas que representam o trajeto por onde um objeto se deslocou. A utilização desta API pode ser enriquecedora para projetos que tenham funções como geolocalização e monitoramento em tempo real (GOOGLE DEVELOPERS, 2025).

## 2.2.6 Geofencing

O *Geofencing* é uma tecnologia que permite criar uma cerca virtual que delimita um espaço geográfico. Essa cerca pode ter qualquer formato e serve para identificar se um objeto entrou ou saiu da área delimitada, com base em sua localização GPS (WAWRZYNIAK; HYLA, 2016, **apud** LOBÔ, 2022).

Figura 8 – Imagem Geofencing



Fonte: Expert Digital, 2025.

## 2.2.7 MongoDB

Conforme a expansão da internet, aumentaram-se as exigências cobradas de um banco de dados. Diante dessa necessidade de mudança, Paniz (2023) afirma que surgiu o conceito de NoSQL (Not Only SQL), buscando apresentar uma nova maneira de armazenar dados.

MongoDB é um sistema de gerenciamento de banco de dados (SGBD) não relacional que utiliza documentos em formato JSON (JavaScript Object Notation) para armazenar informações, simplificando a gestão dos dados pelos desenvolvedores (IBM).

Boaglio (2020) afirma que, diferente das maneiras convencionais, com MongoDB, os dados são organizados conforme a necessidade do sistema, obedecendo às regras da aplicação, sem possuir as limitações de uma tabela relacional.

## 2.2.8 Wireframes

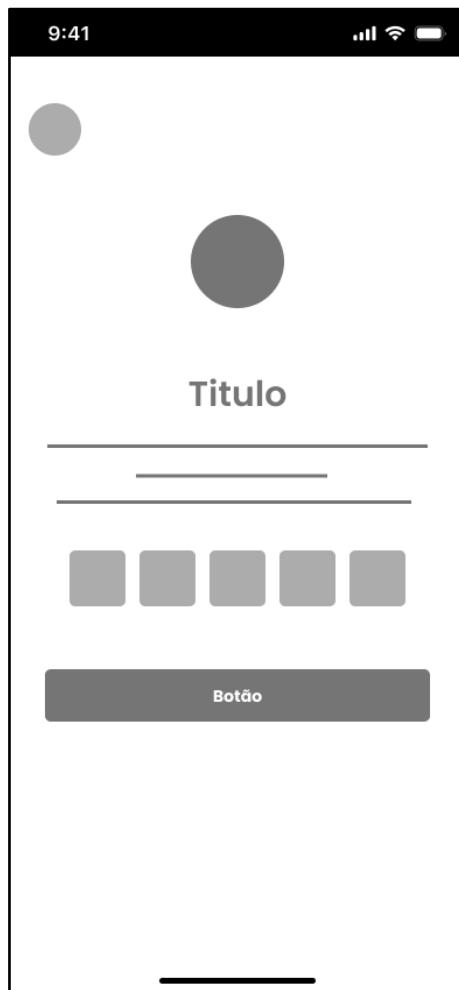
De acordo com Teixeira (2014) o Wireframe é o desenho de uma interface que indica onde cada elemento ficará apresentando de forma intuitiva como será o resultado,

além de ser usado como guia para ter uma ideia sobre o layout final, agilizando o processo.

#### 2.2.8.1 Wireframes baixa fidelidade

Para Rodrigues (2017) os Wireframes de baixa fidelidade são esboços simples usados como uma forma rápida de mostrar uma ideia do produto, mesmo não sendo fiel ao layout final. No entanto, os Wireframes de baixa fidelidade possuem diversas vantagens, como baixo custo e a rápida produção, sendo modificados mais facilmente quando necessário.

Figura 9 – Exemplo de wireframes de baixa fidelidade



Fonte: De Autoria Própria, 2025.

#### 2.2.8.2 Wireframes alta fidelidade

Os Wireframes de alta fidelidade usam elementos que serão colocados no estado final do produto, constituindo, consequentemente, em uma versão mais fiel do que os

wireframes de baixa fidelidade, no entanto, acabam por ser mais caro e demandar mais tempo para ser produzido (Rodrigues, 2017).

Figura 10 – Exemplo de wireframes de alta fidelidade



Fonte: De Autoria Própria, 2025.

### 2.2.9 UX (User Experience)

UX é um termo utilizado para se referir a experiência que um usuário tem ao utilizar certa aplicação. Grant (2019) aponta que para trabalhar com essa questão de experiência, deve se ter empatia e objetividade para poder entender as necessidades e frustrações dos usuários.

Como afirma Teixeira (2014), UX é uma área bem ampla, possuindo várias faces que são utilizadas para aprimorar a interação do usuário, como: organizar um conteúdo dentro da tela, garantir que as interfaces sejam fáceis de usar, definir o fluxo de comportamentos de uma aplicação, entender o público-alvo do produto, entre outros fatores.

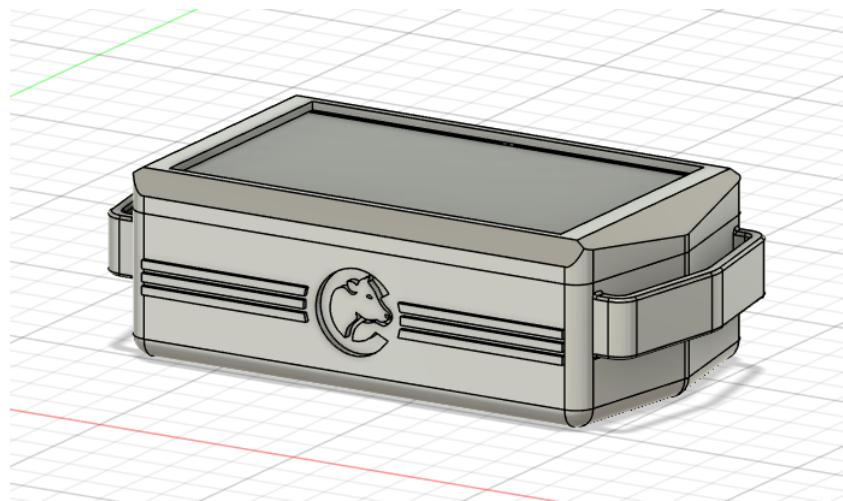
### **2.2.10 LoRa**

Segundo Teixeira e Almeida (2017) O LoRa (Long Range - Longo alcance) desenvolvido pela Semtech Corporation, é uma tecnologia de comunicação sem fio por radiofrequência e baixo consumo de energia. Ela permite a comunicação de dispositivos em uma mesma rede a grandes distâncias sem a necessidade de um cabeamento. Devido a essas características se torna o ideal para aplicações de internet das coisas (IoT) em locais remotos, onde não existe uma conexão Wi-Fi ou rede de celular.

### **2.2.11 Modelagem 3D**

Muito utilizado pela indústria, a modelagem 3D é o processo de desenvolvimento de objetos, personagens, cenários e formas complexas em três dimensões e pode ser utilizada em diversas áreas, desde o design de móveis e objetos até a fabricação de peças automotivas (LOPES, 2023). Abaixo um exemplo de modelagem 3D:

Figura 11 – Exemplo de modelagem 3D T-Fence



Fonte: De Autoria Própria, 2025.

### **2.2.12 UML (Unified Modeling Language)**

De acordo com Guedes (2018), UML é uma linguagem de modelagem utilizada internacionalmente, buscando padronizar o desenvolvimento dos softwares se baseando na programação orientada a objetos. Surgiu da unificação de várias linguagens de modelagem que existiam nos anos 80 e 90, trazendo um padrão para o segmento na época (FOWLER, 2005).

Consoante Booch (2006), a modelagem nos permite compreender o funcionamento de um sistema por completo. No entanto, para se obter esse entendimento é necessários vários modelos conectados entre si. Assim, é possível utilizar a UML para a visualização, especificação, construção e documentação de um projeto.

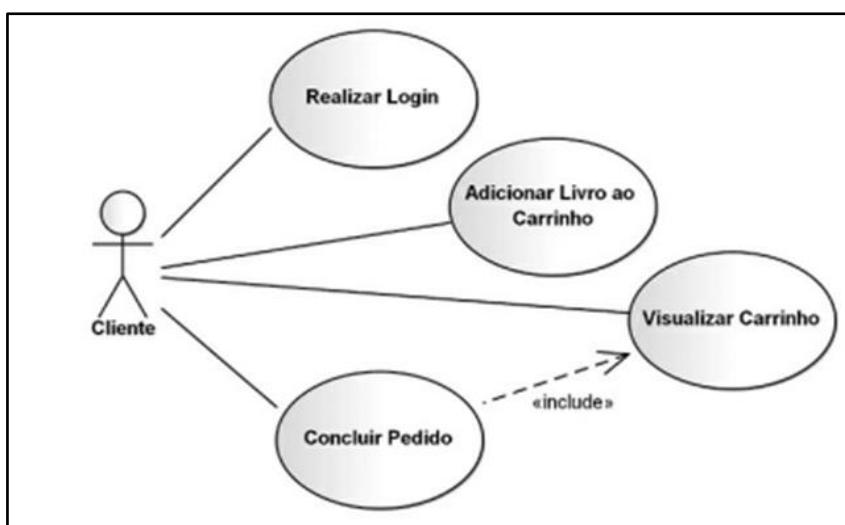
Completando essa ideia, Guedes (2018) destaca que a UML tem diversos diagramas, com objetivo de fornecer várias visões do sistema, analisando diversos aspectos diferentes. O autor também reforça que a utilização desses diagramas é importante, já que ele auxilia na identificação de falhas, diminuindo a possibilidade de erros futuros.

Nos próximos capítulos iremos apresentar os diagramas que utilizamos para a modelagem do nosso sistema.

#### **2.2.12.1 UML- Diagrama de Caso de Uso**

Tem como objetivo apresentar uma visão geral do sistema, exibindo as funcionalidades que devem ser oferecidas. No diagrama, a representação se dá por um conjunto de casos de uso, retratando as funcionalidades, e os atores, sendo tudo aquilo que interage com o sistema (GUEDES, 2018).

Figura 12 – Exemplificação de um diagrama de caso de uso



Fonte: Guedes, 2018.

#### **2.2.12.2 UML- Documentação de Caso de Uso**

Segundo Guedes (2018), a documentação do caso de uso é realizada de maneira simples, trazendo informações básicas, como as funcionalidades do caso de uso, atores e as etapas que devem ser executadas. Ainda sobre esse assunto, o autor diz

que não possui um formato padronizado para se documentar, sendo maleável e de sua escolha.

Figura 13 – Exemplificação de um diagrama de caso de uso

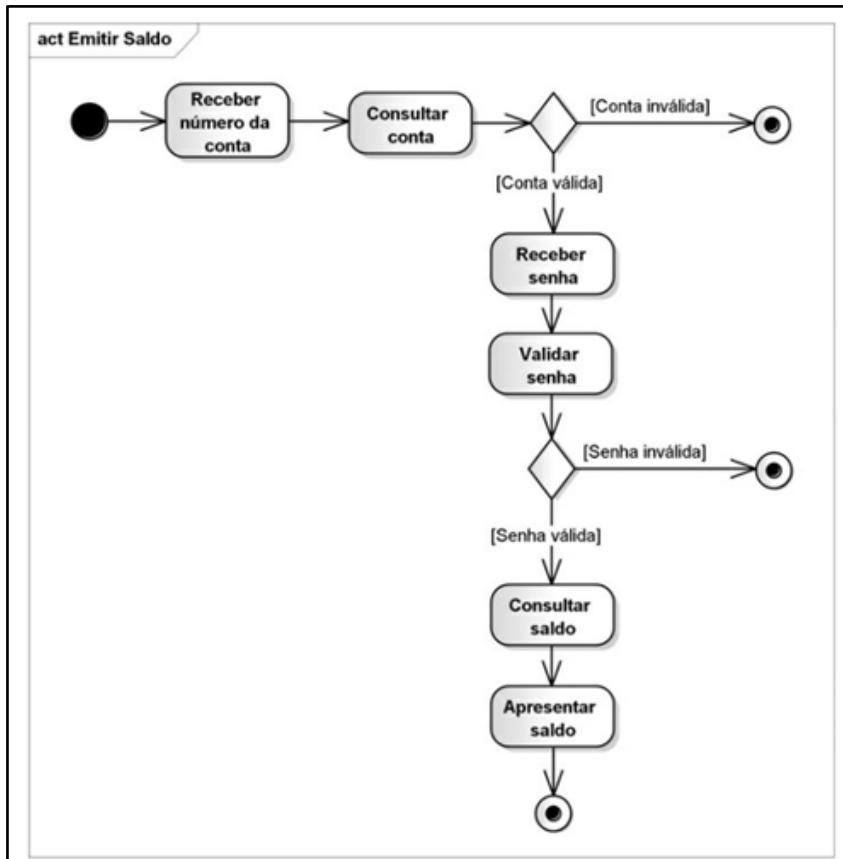
Nome do Caso de Uso		UC06 – Emitir Saldo
Ator Principal	Cliente	
Atores Secundários		
Resumo	Descreve os passos necessários para um cliente obter o saldo referente a uma determinada conta	
Pré-condições		
Pós-condições		
Cenário Principal		
Ações do Ator	Ações do Sistema	
1. Informar o número da conta	2. Verificar a existência da conta 3. Solicitar a senha da conta	
4. Informar a senha	5. Verificar se a senha está correta 6. Emitir o saldo	
Restrições/Validações	1. A conta precisa existir e estar ativa 2. A senha precisa estar correta	
Cenário de Exceção I – Conta não encontrada		
Ações do Ator	Ações do Sistema	
	1. Comunicar ao cliente que o número da conta informada não foi encontrado	
Cenário de Exceção II – Senha inválida		
Ações do Ator	Ações do Sistema	
	1. Comunicar ao cliente que a senha fornecida não combina com a senha da conta	

Fonte: Guedes, 2018.

### 2.2.12.3 UML- Diagrama de Atividade

O diagrama de atividade busca modelar a ordem de atividades em um processo, exibindo a sua ocorrência, contendo ações, nós de atividade, fluxos e valores de objeto (BOOCH, 2006).

Figura 14 – Exemplificação de um diagrama de atividade

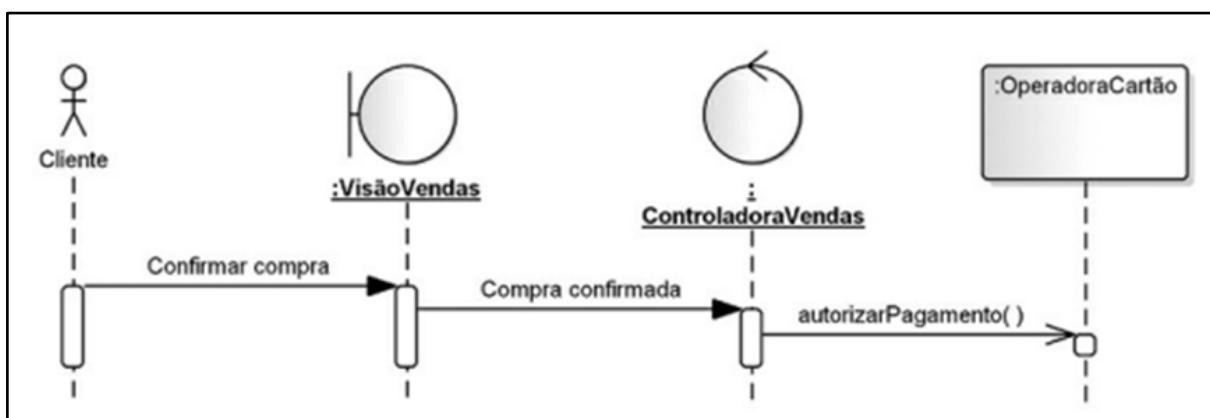


Fonte: Guedes, 2018.

#### 2.2.12.4 UML- Diagrama de Sequência

O diagrama de sequência tem como objetivo determinar a ordem dos eventos em um determinado processo, baseando-se no caso de uso. É normal que cada caso de uso tenha um diagrama de sequência, sendo um processo disparado por um ator (GUEDES, 2018).

Figura 15 – Exemplo de diagrama de Sequência

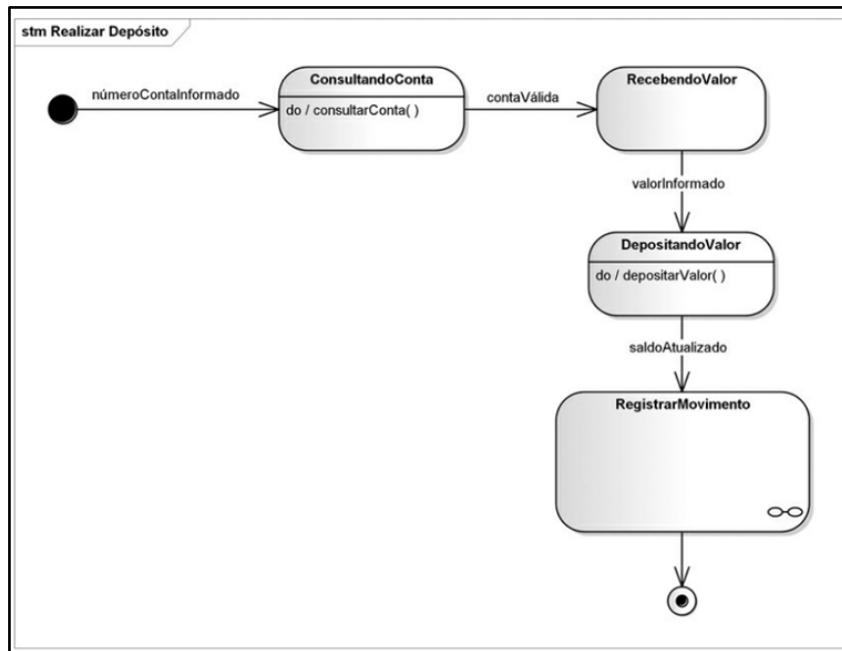


Fonte: Guedes, 2018.

### 2.2.12.5 UML- Diagrama de Máquina de Estado

Para Booch (2006), esse diagrama especifica a sequência de estados que um objeto passa durante seu tempo de vida, levando em consideração seus eventos. O autor também afirma que o estado é a situação de vida de um objeto, enquanto evento é a especificação de uma ocorrência.

Figura 16 – Exemplo de diagrama de Máquina de Estados



Fonte: Guedes, 2018.

### 2.2.13 IoT

Conforme afirma Magrani (2018), IoT (Internet of Things - Internet das Coisas) se baseia em objetos físicos que estejam conectados com a internet e que compartilhem informações por meio de sensores, com a finalidade de auxiliar ou automatizar tarefas.

Segundo Oliveira (2021), o IoT já é um conceito que existe há muito tempo, começou a ter destaque com o avanço da internet, surgiu com a ideia de interligar dispositivos por meio da internet, assim possibilitando não apenas a comunicação entre eles, mas também torná-los inteligentes melhorando a forma com que interagem com o ambiente.

#### 2.2.13.1 ESP32

Lançado em 2016 pela Espressif Systems, o ESP32 é um microcontrolador avançado, considerado o melhor em comparação a outros microcontroladores do mercado,

devido ao seu poder de processamento e capacidade de memória, o ESP32 conta com um processador dual core com 2 núcleos físicos, 530Kb de memória RAM, podendo chegar a um clock de até 240Mhz e ainda possui conectividade com Bluetooth e Wi-Fi, sendo assim uma ótima opção para aplicações IoT (SANTOS; LARA JUNIOR, 2019).

Figura 17 – Imagem do ESP32



Fonte: De Autoria Própria, 2025.

### 2.2.13.2 Módulo LoRa

O módulo LoRa é um transmissor sem fio baseado na tecnologia LoRa (Long Range), equipado desenvolvido pela SEMTECH, este módulo permite a comunicação a longas distâncias com um baixo consumo de energia, muito utilizado em aplicações IoT (Internet das coisas) (SARAVATI, 2025).

Figura 18 – Módulo Transmissor LoRa SX1276



Fonte: De Autoria Própria, 2025.

### 2.2.13.3 Módulo controlador de carga

Como destaca Oliveira (2021) o controlador de carga é componente essencial de um IoT já que garante o bom funcionamento de sua bateria. Um controlador de carga possui diversas funções, dentre elas estão, evitar descarga total, que pode danificar a bateria diminuindo sua vida útil, limitar a carga máxima, evitando uma sobrecarga o que pode causar um superaquecimento ou até mesmo à explosão da bateria.

Figura 19 - Módulo controlador de carga TP4057



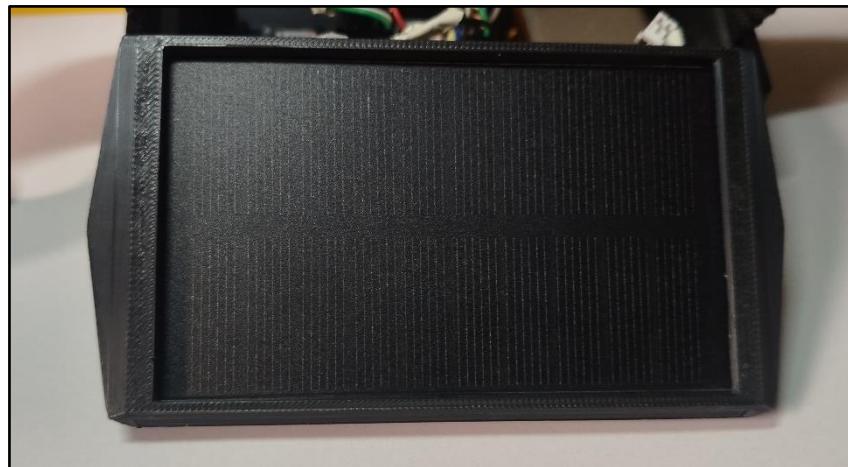
Fonte: De Autoria Própria, 2025.

### 2.2.13.4 Mini painel solar

De acordo com Braga (2008) os painéis solares são capazes de converter radiação solar em energia elétrica a partir do efeito fotovoltaico. Descoberto pelo físico francês Edmond Becquerel em 1839, que ao observar que certos materiais semicondutores, expostos à luz, geram uma pequena corrente elétrica.

Segundo Oliveira (2021) a energia fotovoltaica tem se mostrado uma ótima solução para alimentar energeticamente dispositivos de Internet das coisas (IoT), principalmente quando se trata de aplicações externas e remotas, onde a infraestrutura elétrica pode ser limitada ou até mesmo inexistente.

Figura 20 – Mini painel solar



Fonte: De Autoria Própria, 2025.

#### 2.2.13.5 Módulo GPS Neo-M8N

O módulo GPS Neo-M8N é um dispositivo capaz de determinar coordenadas geográficas com alta precisão, se comunica com até três sistemas de navegação por satélite (GNSS), GPS, GLONASS e BeiDou, garantindo menor taxa de erro e alta confiabilidade (CURTOCIRCUITO, 2025).

Figura 21 – Módulo GPS Neo-M8N



Fonte: De Autoria Própria, 2025.

### 2.2.13.6 Bateria Li-Po

Existe uma variedade de baterias, dentre elas está a bateria de lítio-polímero (Li-Po), sendo uma variação da bateria de lítio, ela se destaca pela sua capacidade de se adequar em aplicações com espaço limitado, sendo possível encontrar baterias Li-Po de diferentes formas e tamanhos, além de serem mais leves e seguras (MAKERHERO, 2025).

Figura 22 – Bateria de lítio-polímero

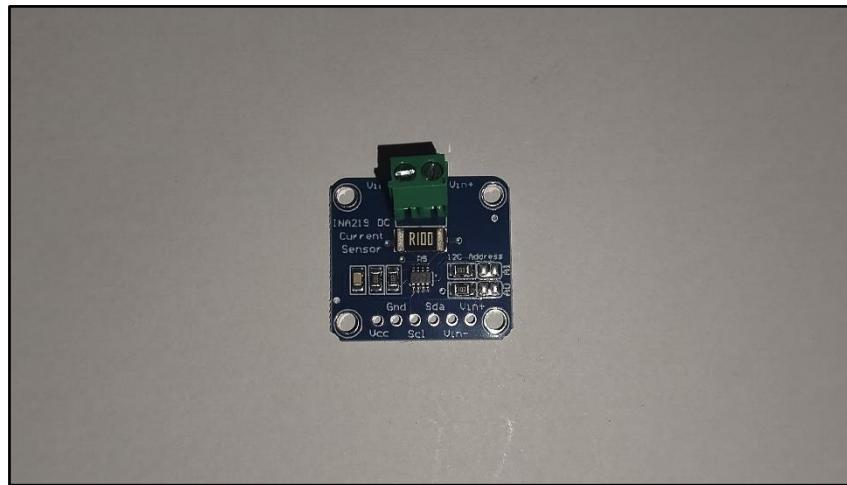


Fonte: De Autoria Própria, 2025.

### 2.2.13.7 Módulo MCU-219

O módulo MCU-219 foi projetado para monitoramento de corrente e tensão em sistemas DC, ele opera entre 3V e 5V, mede tensões de um barramento de até 26VDC e correntes de até +3,2 A (CURTOCIRCUITO, 2021). Esse módulo será utilizado para estimar a porcentagem da bateria da coleira.

Figura 23 – Imagem do MCU-219

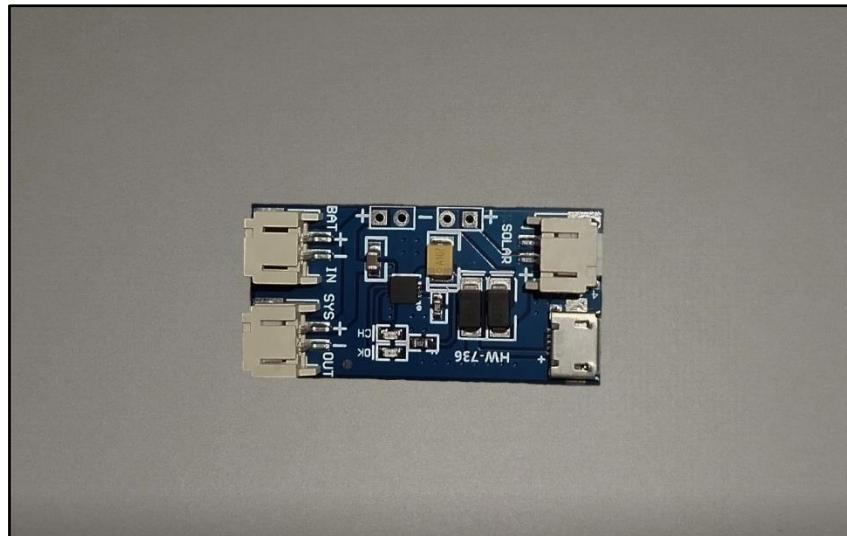


Fonte: De Autoria Própria, 2025.

#### 2.2.13.8 Módulo CN3065

O módulo CN3065 é um carregador solar que tem como principal função a administração da energia das baterias de lítio, operando no controle caso essa bateria precise de carga (TEIXEIRA, 2025).

Figura 24 - Diagrama de Caso de Uso: T-Fence



Fonte: De Autoria Própria, 2025.

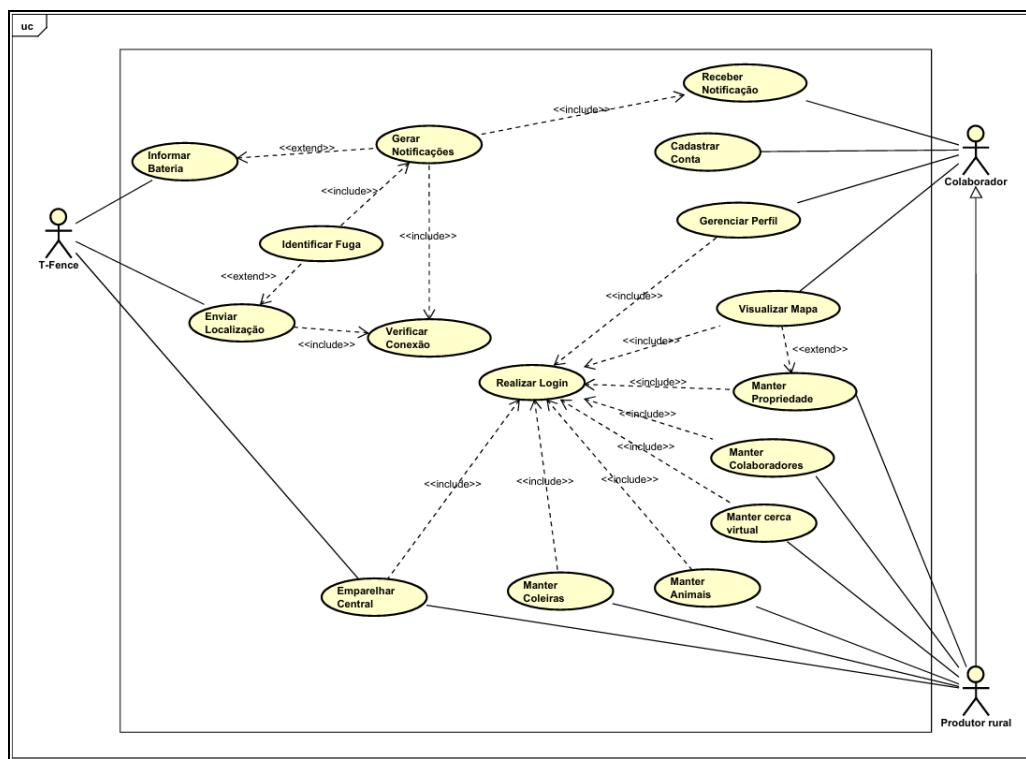
### 3 DESENVOLVIMENTO

Neste capítulo serão apresentados os processos utilizados durante o desenvolvimento do sistema T-Fence, partindo da utilização de métodos de documentação UML até a construção efetiva de nosso aplicativo e dos dispositivos IoT.

#### 3.1 Diagrama de Casos de Uso

Este capítulo permite visualizar detalhadamente o Diagrama de Caso de Uso do sistema T-Fence. Pode-se observar a existência de três atores que interagem diretamente com o sistema: Dispositivo T-Fence, Colaborador e Produtor Rural. O diagrama apresenta as ações que poderão ser executadas por cada ator, representadas em 16 elipses de caso de uso.

Figura 25 - Diagrama de Caso de Uso: T-Fence



Fonte: De Autoria Própria, 2025.

#### 3.2 Documentação Caso de Uso

Com base no diagrama apresentado no capítulo anterior, será detalhada a seguir a documentação dos casos de uso de nosso sistema. Para o desenvolvimento de nossa documentação, utilizamos uma estrutura dividida em 3 tópicos principais: Requisitos Funcionais, Requisitos Não Funcionais e Regras de Negócio.

Os requisitos funcionais evidenciam as funcionalidades que estão incluídas em nosso sistema, sendo essenciais para o funcionamento adequado. Os requisitos não funcionais são características atribuídas ao projeto, sendo essenciais para definir a qualidade e as restrições do projeto, buscando muitas vezes atender expectativas relacionadas a performance, usabilidade, segurança entre outras. Por fim, as regras de negócio são políticas e princípios baseados em condições e restrições que o sistema obrigatoriamente deve seguir, visando atender as necessidades reais do usuário.

#### Requisito Funcionais:

- RF01: O produtor rural e o colaborador podem se cadastrar no sistema utilizando seus dados;
- RF02: O produtor rural e o colaborador podem gerenciar o seu perfil no sistema;
- RF03: O produtor rural e o colaborador podem visualizar um mapa interativo com a localização da cerca virtual e de cada animal com coleira ativa;
- RF04: O sistema deve permitir o cadastro e o gerenciamento de dispositivos IoT (coleiras) com identificação única, permitindo associar a um animal já cadastrado;
- RF05: O sistema deve permitir que o produtor rural autorize solicitações e gerencie pessoas para acessar as informações da propriedade (ex: funcionários da fazenda);
- RF06: O produtor rural deve ser capaz de desenhar, salvar e alterar um perímetro virtual sobre o mapa, delimitando a área segura para os animais, e determinar quais coleiras pertencem a determinado perímetro;
- RF07: O sistema deve permitir o cadastro e o gerenciamento de animais com identificação única;
- RF08: O sistema deve permitir ao produtor cadastrar uma propriedade com nome, localização, área aproximada e uma breve descrição;
- RF09: O sistema deve permitir ao usuário emparelhar o dispositivo IoT (Central) a uma rede Wi-Fi;
- RF10: O produtor rural e o colaborador devem ser capazes de realizar login na plataforma, caso já tenham se registrado, podendo assim obter acesso às funcionalidades do sistema;

- RF11: O sistema deve enviar a localização da coleira para o aplicativo, permitindo o monitoramento dos animais;
- RF12: O sistema deve verificar a porcentagem de bateria do dispositivo, informando no aplicativo;
- RF13: O sistema deve identificar caso algum animal saia do perímetro delimitado pelas cercas virtuais;
- RF14: O sistema deve gerar notificações caso identifique a fuga de algum animal ou se a bateria estiver baixa, enviando-as para o aplicativo.

#### Requisitos Não Funcionais:

- RNF01: O sistema deve funcionar corretamente em dispositivos móveis Android, com interface responsiva.
- RNF02: O sistema deve garantir a segurança e integridade dos dados cadastrados pelo produtor.
- RNF03: O sistema deve possuir uma interface intuitiva e de fácil interação, buscando facilitar a utilização do aplicativo pelo usuário.
- RNF04: O sistema deve possuir um rápido tempo de resposta, enviando os alertas críticos com agilidade, prevenindo roubos ou problemas nos dispositivos.
- RNF05: O sistema deve priorizar um baixo consumo energético, gastando o mínimo de bateria e recarregador de forma autônoma com energia solar.

#### Regras de Negócio:

- RN01: Um colaborador só pode acessar o sistema caso o produtor rural aceite a sua solicitação.
- RN02: O sistema deve gerar notificações no aplicativo, quando a bateria dos dispositivos estiver abaixo de 20%.
- RN03: O sistema deve gerar notificações no aplicativo, caso a coleira perca o sinal por 10 minutos.
- RN04: O sistema deve gerar notificações no aplicativo, caso a coleira seja violada e retirada do animal.

Tabela 1 - Documentação do caso de uso: Cadastrar Conta

<b>Nome do Caso de Uso</b>	Cadastrar Conta
<b>Autor Principal</b>	Produtor Rural/Colaborador
<b>Resumo</b>	Este caso de uso descreve a sequência de passos que se realiza ao se cadastrar no sistema
<b>Pós-condições</b>	Os dados são cadastrados no sistema e o produtor deve se vincular ou cadastrar uma propriedade
<b>Cenário Principal</b>	
<b>Ações do Ator</b>	<b>Ações do Sistema</b>
1. Acessa a área de cadastro	
	2. Exibe um formulário requisitando algumas informações
3. Insere dados solicitados pelo sistema e envia o formulário	
	4. Valida os dados e registra o produtor rural no banco de dados
<b>Cenário Alternativo – Dados incorretos</b>	
<b>Ações do Ator</b>	<b>Ações do Sistema</b>
	1. O produtor é notificado que os dados estão incorretos
	2. O sistema pede para que o produtor preencha os dados novamente

Fonte: De Autoria Própria, 2025.

Tabela 2 - Documentação do caso de uso: Manter Propriedade

<b>Nome do Caso de Uso</b>	Manter propriedade
<b>Ator Principal</b>	Produtor Rural
<b>Resumo</b>	Este caso de uso descreve a sequência de passos que ele realiza ao realizar o cadastro da propriedade
<b>Pré-condições</b>	O produtor deve estar logado no sistema
<b>Cenário Principal</b>	
<b>Ações do Ator</b>	<b>Ações do Sistema</b>
	1. Solicita informações da propriedade
2. Insere dados solicitados pelo sistema	
3. Confirma o cadastro	
	3. Verifica se os campos são válidos
	4. Efetua o cadastro da propriedade
<b>Cenário Alternativo – Dados incorretos</b>	
<b>Ações do Ator</b>	<b>Ações do Sistema</b>

	1. O produtor é notificado que os dados não são válidos
	2. Pede que o produtor reinsira os dados

Fonte: De Autoria Própria, 2025.

Tabela 3 - Documentação do caso de uso: Realizar Login

<b>Nome do Caso de Uso</b>	Realizar Login
<b>Autor Principal</b>	Produtor Rural
<b>Resumo</b>	Este caso de uso descreve a sequência de passos que o produtor rural/colaborador realiza ao efetuar o login
<b>Pré-condições</b>	Para efetuar login, é preciso ter um cadastro no sistema
<b>Cenário Principal</b>	
<b>Ações do Ator</b>	<b>Ações do Sistema</b>
	1. Solicitar e-mail e senha
2. Insere dados solicitados pelo sistema	
	3. Verificar se os dados já foram cadastrados

	4. Efetuar o login
<b>Cenário Alternativo – Dados incorretos</b>	
Ações do Ator	Ações do Sistema
	1. O produtor é notificado que os dados estão incorretos ou não estão cadastrados 2. O sistema recusa a solicitação de login

Fonte: De Autoria Própria, 2025.

Tabela 4 - Documentação do caso de uso: Emparelhar Central

<b>Nome do Caso de Uso</b>	Emparelhar central
<b>Autor Principal</b>	Produtor Rural
<b>Resumo</b>	Este caso de uso descreve a sequência de passos que o produtor rural realiza emparelhar a Central
<b>Pré-condições</b>	Para efetuar o emparelhamento é necessário ter uma propriedade cadastrada
<b>Cenário Principal</b>	
Ações do Ator	Ações do Sistema

	1. Solicita nome da rede, senha
2. Insere dados solicitados pelo sistema	
	3. Verifica se os dados são válidos
	4. Efetua o emparelhamento
<b>Cenário Alternativo – Dados incorretos</b>	
Ações do Ator	Ações do Sistema
	1. O produtor é notificado que os dados estão incorretos
	2. Pede que o produtor reinsira os dados

Fonte: De Autoria Própria, 2025.

Tabela 5 - Documentação do caso de uso: Manter Coleiras

<b>Nome do Caso de Uso</b>	Manter Coleiras
<b>Autor Principal</b>	Produtor Rural
<b>Resumo</b>	O produtor pode cadastrar e gerenciar uma coleira no sistema, além de vincular o dispositivo a um animal já cadastrado.
<b>Pré-condições</b>	O produtor deve estar logado no sistema. A central deve estar emparelhada.

<b>Cenário Principal</b>	
<b>Ações do Ator</b>	<b>Ações do Sistema</b>
	1. Solicitar dados Sobre a Coleira
2. Insere dados solicitados pelo sistema	
	3. Verificar se os dados já foram cadastrados
	4. É efetuado o cadastro da coleira
5. Opcionalmente, o produtor pode editar ou excluir sua coleira	
<b>Cenário Alternativo – Dados incorretos</b>	
<b>Ações do Ator</b>	<b>Ações do Sistema</b>
	1. O produtor é notificado que há dados inválidos
	2. O sistema recusa o cadastro da coleira

Fonte: De Autoria Própria, 2025.

Tabela 6 - Documentação do caso de uso: Manter Animais

<b>Nome do Caso de Uso</b>	Manter animais
----------------------------	----------------

<b>Ator Principal</b>	Produtor Rural
<b>Resumo</b>	O produtor pode cadastrar e gerenciar um animal no sistema.
<b>Pré-condições</b>	O produtor deve estar logado no sistema. A central deve estar emparelhada.
<b>Cenário Principal</b>	
<b>Ações do Ator</b>	<b>Ações do Sistema</b>
	1. Solicitar dados sobre o animal
2. Insere dados solicitados pelo sistema	
	3. Verifica a validez dos dados
	4. É efetuado o cadastro do animal
5. Opcionalmente, o produtor pode editar ou excluir os dados do animal	
<b>Cenário Alternativo – Dados incorretos</b>	
<b>Ações do Ator</b>	<b>Ações do Sistema</b>
	1. O produtor é notificado que os dados estão incorretos

	2. O sistema recusa o cadastro do animal
--	--

Fonte: De Autoria Própria, 2025.

Tabela 7 - Documentação do caso de uso: Manter Cerca Virtual

<b>Nome do Caso de Uso</b>	Manter Cerca Virtual
<b>Autor Principal</b>	Produtor Rural
<b>Resumo</b>	O produtor pode cadastrar e gerenciar uma cerca virtual no sistema.
<b>Pré-condições</b>	O produtor deve estar logado no sistema. A central deve estar emparelhada.
<b>Cenário Principal</b>	
<b>Ações do Ator</b>	<b>Ações do Sistema</b>
	1. Solicitar Dados da cerca virtual
2. Insere dados solicitados pelo sistema	
	3. Verifica se os dados são válidos
	4. É efetuado o cadastro da cerca
5. Opcionalmente, o produtor pode editar ou excluir os dados da cerca virtual	

<b>Cenário Alternativo – Dados incorretos</b>	
<b>Ações do Ator</b>	<b>Ações do Sistema</b>
	1. O sistema identifica que os dados estão inválidos
	2. O sistema recusa os dados e notifica produtor

Fonte: De Autoria Própria, 2025.

Tabela 8 - Documentação do caso de uso: Manter Colaboradores

<b>Nome do Caso de Uso</b>	Manter Colaborador
<b>Ator Principal</b>	Produtor Rural
<b>Resumo</b>	O produtor gerencia um colaborador no sistema.
<b>Pré-condições</b>	O produtor deve estar logado no sistema.
<b>Cenário Principal</b>	
<b>Ações do Ator</b>	<b>Ações do Sistema</b>
	1. Sistema mostra se existem pedidos de colaboradores
2. O produtor, opcionalmente, aceita o pedido	

3. O produtor pode, opcionalmente, excluir colaboradores existentes	
	4. É efetuado a exclusão e notificado o produtor

Fonte: De Autoria Própria, 2025.

Tabela 9 - Documentação do caso de uso: Enviar Localização

<b>Nome do Caso de Uso</b>	Enviar Localização
<b>Autor Principal</b>	T-Fence
<b>Resumo</b>	Descreve a sequência de passos que o sistema faz ao enviar uma localização
<b>Pré-condições</b>	É preciso estar emparelhado com a central
<b>Cenário Principal</b>	
<b>Ações do Ator</b>	<b>Ações do Sistema</b>
1. Envia dados da localização	
	2. Verifica os dados enviados
	4. É apresentado a localização definida
<b>Cenário Alternativo – Localização Errada</b>	

Ações do Ator	Ações do Sistema
1. Envia localização errada	
	2. Valida os dados enviados
	3. É apresentado a localização errada

Fonte: De Autoria Própria, 2025.

Tabela 10 - Documentação do caso de uso: Informar Bateria

<b>Nome do Caso de Uso</b>	Informar Bateria
<b>Ator Principal</b>	T-Fence
<b>Resumo</b>	Este caso de uso descreve a sequência de passos para verificar a bateria
<b>Pré-condições</b>	É preciso estar com a central emparelhada
<b>Cenário Principal</b>	
<b>Ações do Ator</b>	<b>Ações do Sistema</b>
1. Envia informações da bateria	
	2. Calcula a porcentagem da bateria
	3. Apresenta a porcentagem da bateria

<b>Nome do Caso de Uso</b>	Informar Bateria
<b>Autor Principal</b>	T-Fence
<b>Resumo</b>	Este caso de uso descreve a sequência de passos para verificar a bateria
<b>Pré-condições</b>	É preciso estar com a central emparelhada
<b>Cenário Alternativo – Bateria abaixo de 20%</b>	
<b>Ações do Ator</b>	<b>Ações do Sistema</b>
1. Informa porcentagem da bateria	
	2. Verifica a porcentagem da bateria
	3. Notifica o produtor da baixa porcentagem
	4. Apresenta a porcentagem da bateria

Fonte: De Autoria Própria, 2025.

Tabela 11 - Documentação do caso de uso: Identificar Fuga

<b>Nome do Caso de Uso</b>	Identificar Fuga
<b>Ator Principal</b>	T-fence
<b>Resumo</b>	Este caso de uso descreve a sequência de passos identificar uma fuga
<b>Pré-condições</b>	Para identificar uma fuga, é preciso estar emparelhado com a central
<b>Cenário Principal</b>	
<b>Ações do Ator</b>	<b>Ações do Sistema</b>
1. Envia a Localização do animal	
	2. Compara a localização do animal com o perímetro
	3. Atualiza a posição do animal no mapa
<b>Cenário Alternativo – Animal fora do Perímetro</b>	
<b>Ações do Ator</b>	<b>Ações do Sistema</b>
	1. Compara a localização do animal com o perímetro
	2. Envia um alerta em forma de notificação

	3. Atualiza a posição do animal no mapa
--	---

Fonte: De Autoria Própria, 2025.

Tabela 12 - Documentação do caso de uso: Gerar Notificações

<b>Nome do Caso de Uso</b>	Gerar notificações
<b>Autor Principal</b>	T-Fence
<b>Resumo</b>	Este caso de uso descreve a sequência de passos para gerar notificações
<b>Pré-condições</b>	Para gerar notificações, é preciso estar emparelhado com a central
<b>Cenário Principal</b>	
<b>Ações do Ator</b>	<b>Ações do Sistema</b>
1. Envia dados gerais	
	2. Verifica se há algum problema
	3. Envia uma notificação referente ao problema encontrado

Fonte: De Autoria Própria, 2025.

Tabela 13 - Documentação do caso de uso: Visualizar Mapa

<b>Nome do Caso de Uso</b>	Visualizar mapa
<b>Autor Principal</b>	Produtor Rural

<b>Resumo</b>	Este caso de uso descreve a sequência de passos que o produtor rural realiza para visualizar mapa
<b>Pré-condições</b>	É preciso estar logado no sistema. Emparelhado a central.
<b>Cenário Principal</b>	
<b>Ações do Ator</b>	<b>Ações do Sistema</b>
	1. Enviar mapa atualizado
2. Escolher algum animal para monitorar ou ver seus dados	

Fonte: De Autoria Própria, 2025.

Tabela 14 - Documentação do caso de uso: Gerenciar Perfil

<b>Nome do Caso de Uso</b>	Gerenciar Perfil
<b>Autor Principal</b>	Colaborador/Produtor Rural
<b>Resumo</b>	O produtor/colaborador gerencia seu perfil no sistema.
<b>Pré-condições</b>	O produtor deve estar logado no sistema. A central deve estar emparelhada.
<b>Cenário Principal</b>	

Ações do Ator	Ações do Sistema
	1. Exibir dados do perfil
2. Editar informações do perfil	
	3. Verificar se os dados são válidos
	4. Os dados são atualizados
<b>Cenário Alternativo – Excluir Perfil</b>	
Ações do Ator	Ações do Sistema
1. Solicita exclusão do perfil	
	2. O sistema confirma a exclusão
	3. exibe mensagem de sucesso

Fonte: De Autoria Própria, 2025.

Tabela 15 - Documentação do caso de uso: Receber Notificação

<b>Nome do Caso de Uso</b>	Receber Notificação
<b>Autor Principal</b>	Colaborador/Produtor Rural
<b>Resumo</b>	O produtor recebe as notificações enviadas pelo sistema.
<b>Cenário Principal</b>	

Ações do Ator	Ações do Sistema
	1. O sistema gera notificações
	2. O sistema envia as notificações para o aplicativo
3. O produtor/colaborador recebe as notificações	

Fonte: De Autoria Própria, 2025.

Tabela 16 - Documentação do caso de uso: Verificar Conexão

<b>Nome do Caso de Uso</b>	Verificar Conexão
<b>Ator Principal</b>	T-Fence
<b>Resumo</b>	O sistema verifica a conexão antes de enviar a localização e gerar notificações para o aplicativo T-Fence.
<b>Cenário Principal</b>	
Ações do Ator	Ações do Sistema
1. O sistema inicia verificação da conexão	
	2. O sistema envia as notificações para o aplicativo

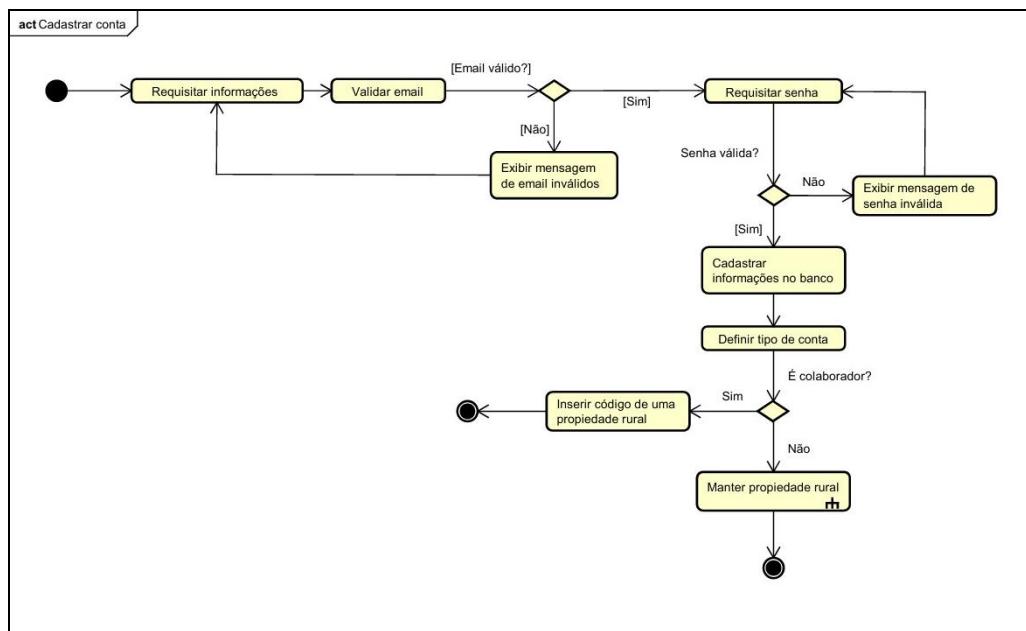
3. O produtor/colaborador recebe as notificações

Fonte: De Autoria Própria, 2025.

### 3.3 Diagrama de Atividade

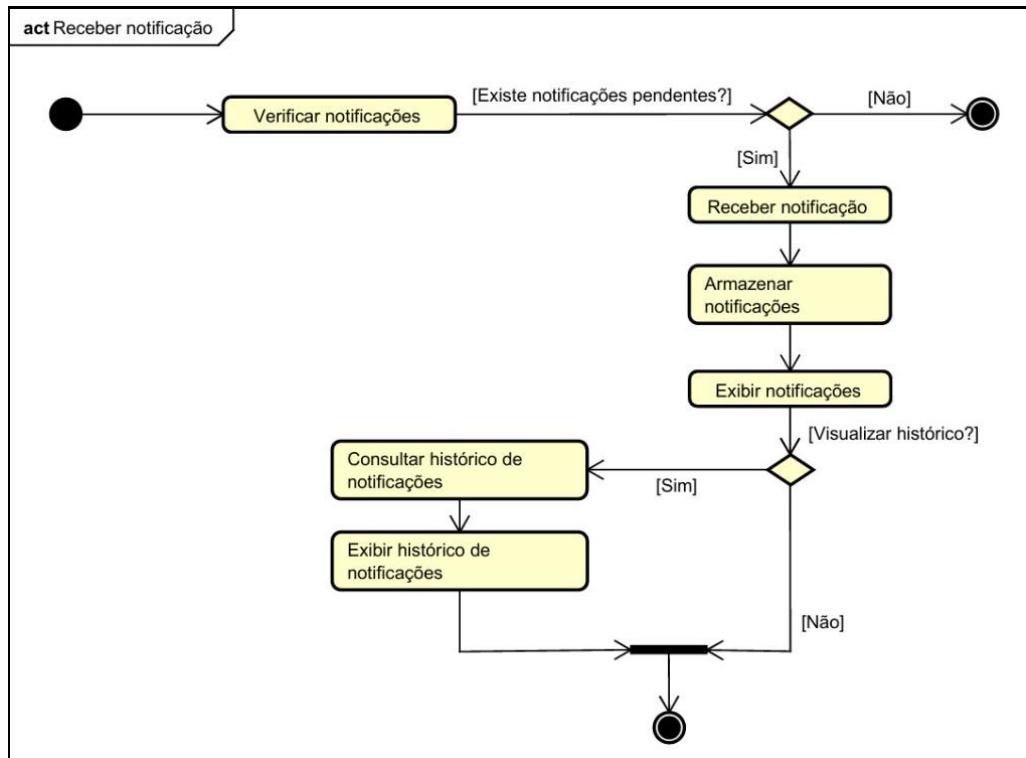
Para nosso projeto, desenvolvemos alguns diagramas para detalhar o fluxo de ações das atividades exercida durante os processos do sistema. A seguir pode se observar todos os diagramas de atividade do sistema T-Fence.

Figura 26 – Diagrama de Atividade: Cadastrar Conta



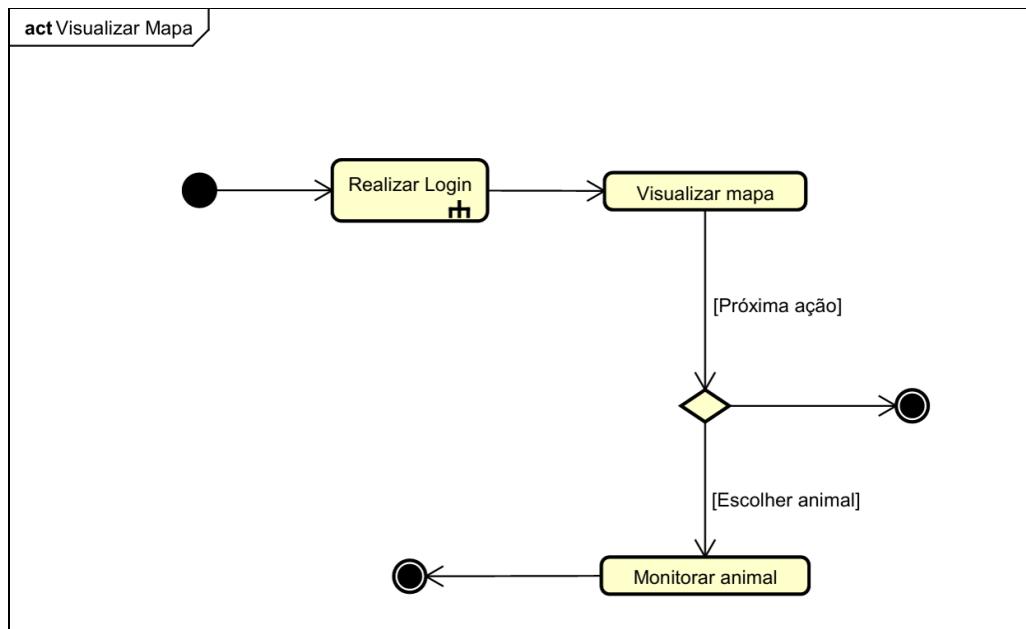
Fonte: De Autoria Própria, 2025.

Figura 27 – Diagrama de Atividade: Receber Notificação



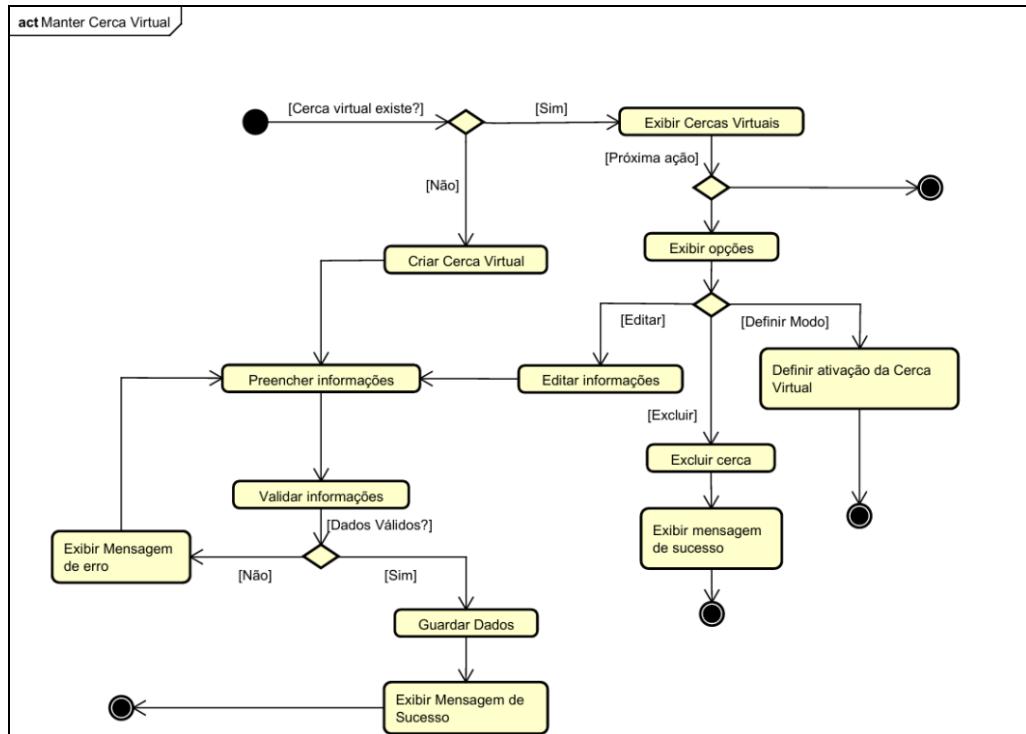
Fonte: De Autoria Própria, 2025.

Figura 28 – Diagrama de Atividade: Visualizar Mapa



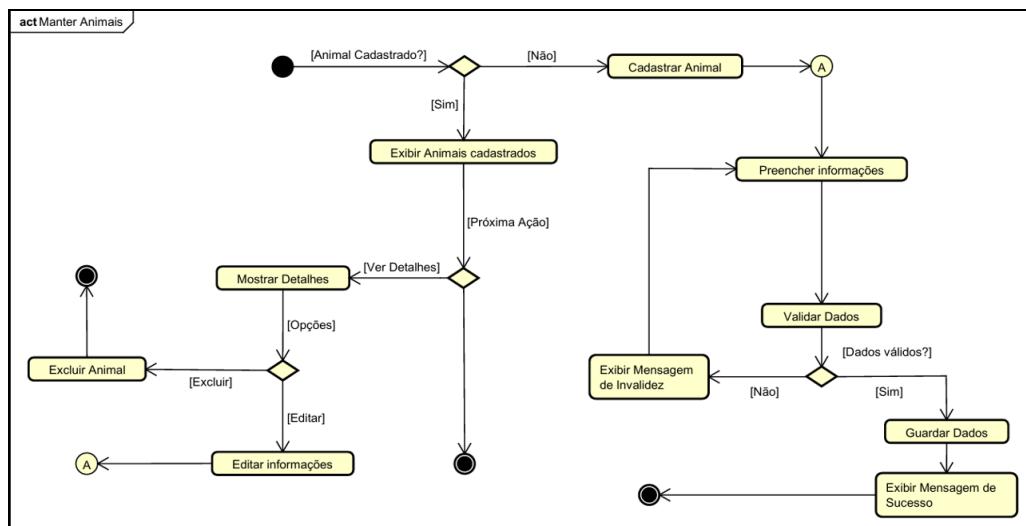
Fonte: De Autoria Própria, 2025.

Figura 29 – Diagrama de Atividade: Manter Cerca Virtual



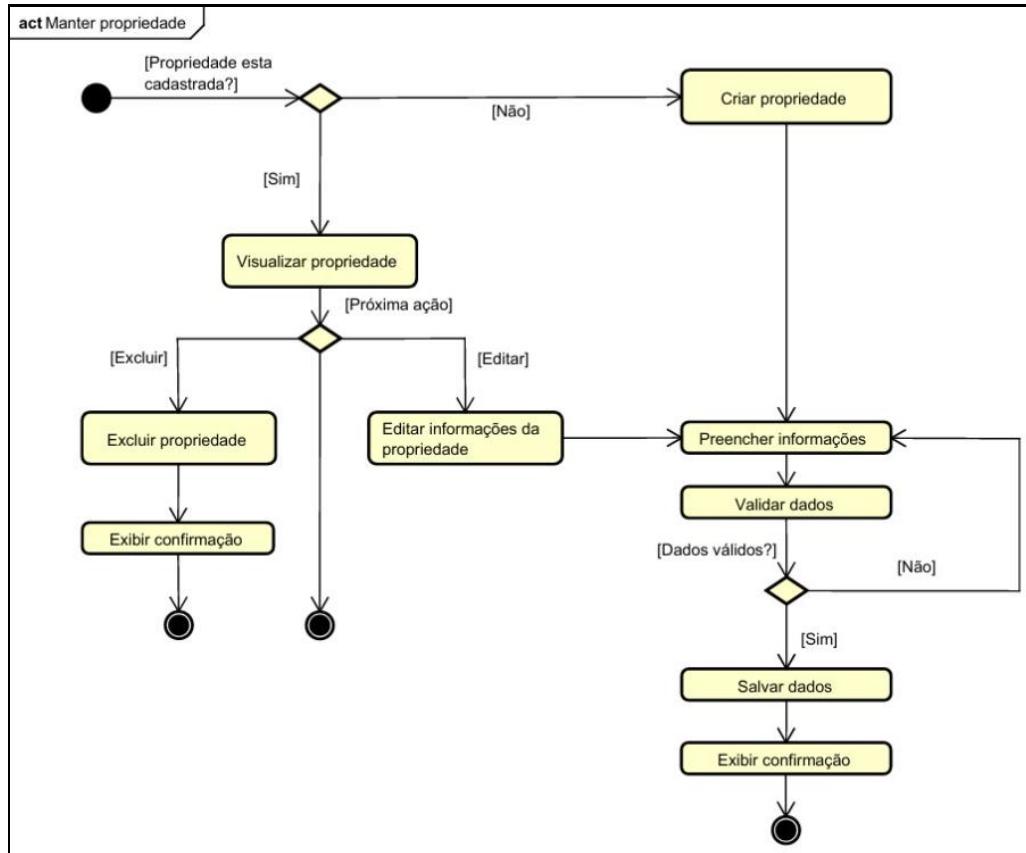
Fonte: De Autoria Própria, 2025.

Figura 30 – Diagrama de Atividade: Manter Animais



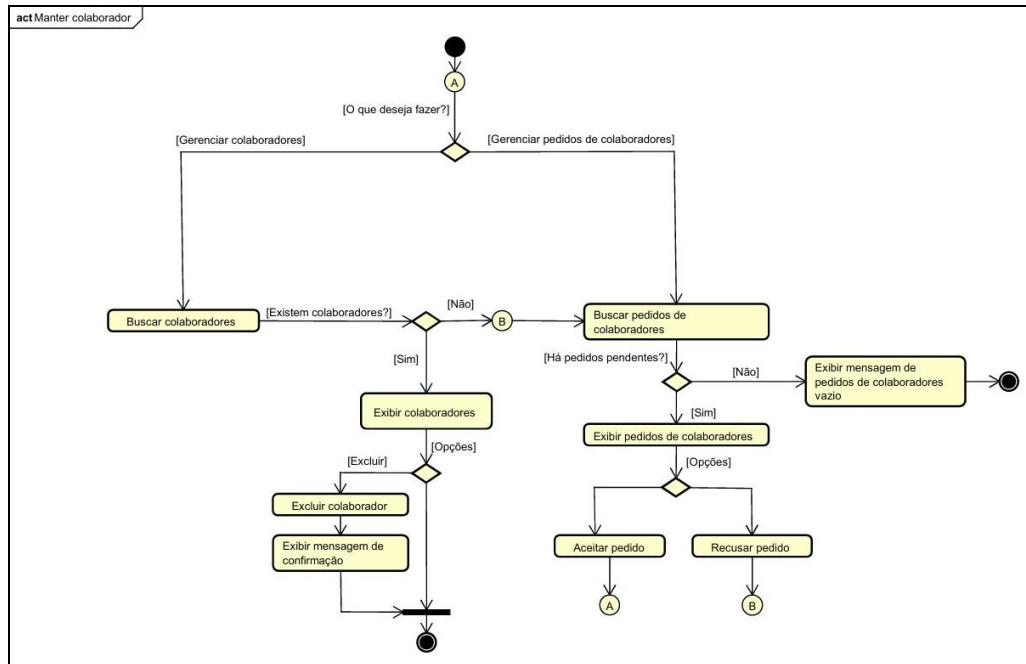
Fonte: De Autoria Própria, 2025.

Figura 31 – Diagrama de Atividade: Manter propriedade



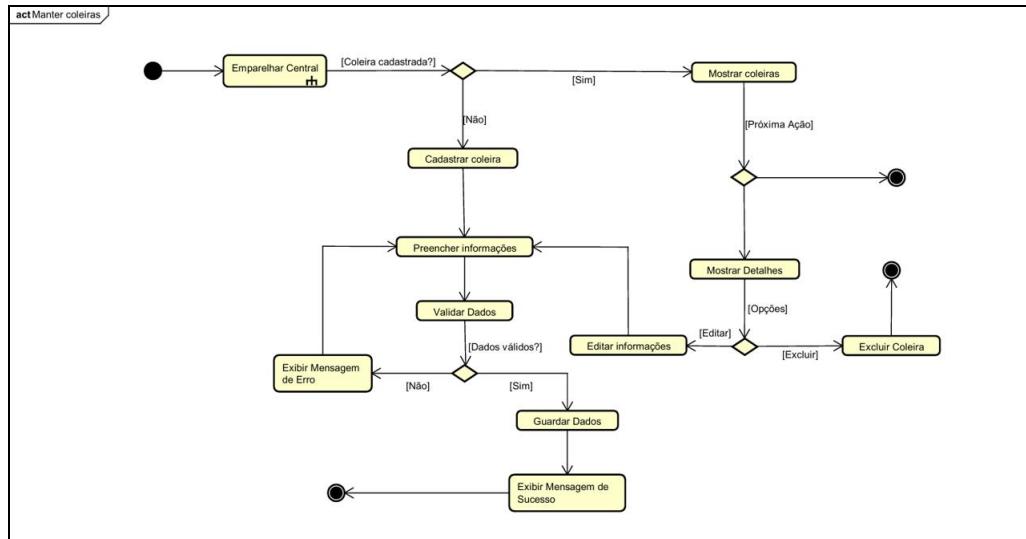
Fonte: De Autoria Própria, 2025.

Figura 32 – Diagrama de Atividade: Manter Colaborador



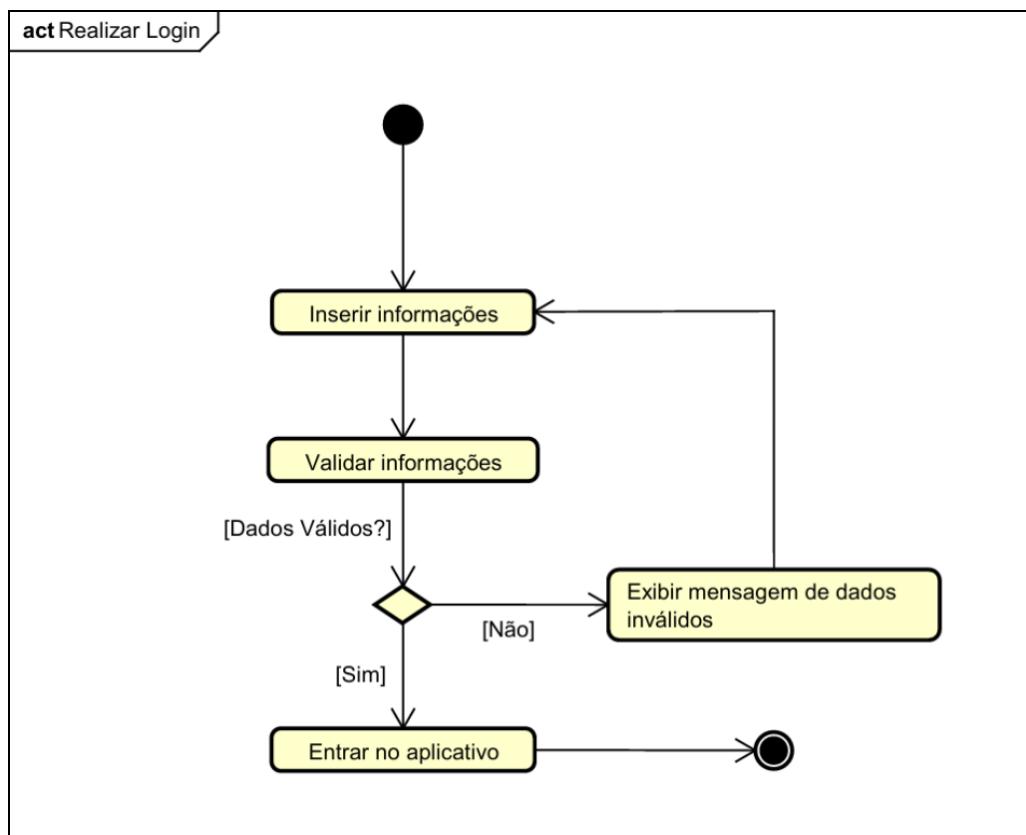
Fonte: De Autoria Própria, 2025.

Figura 33 – Diagrama de Atividade: Manter Coleiras



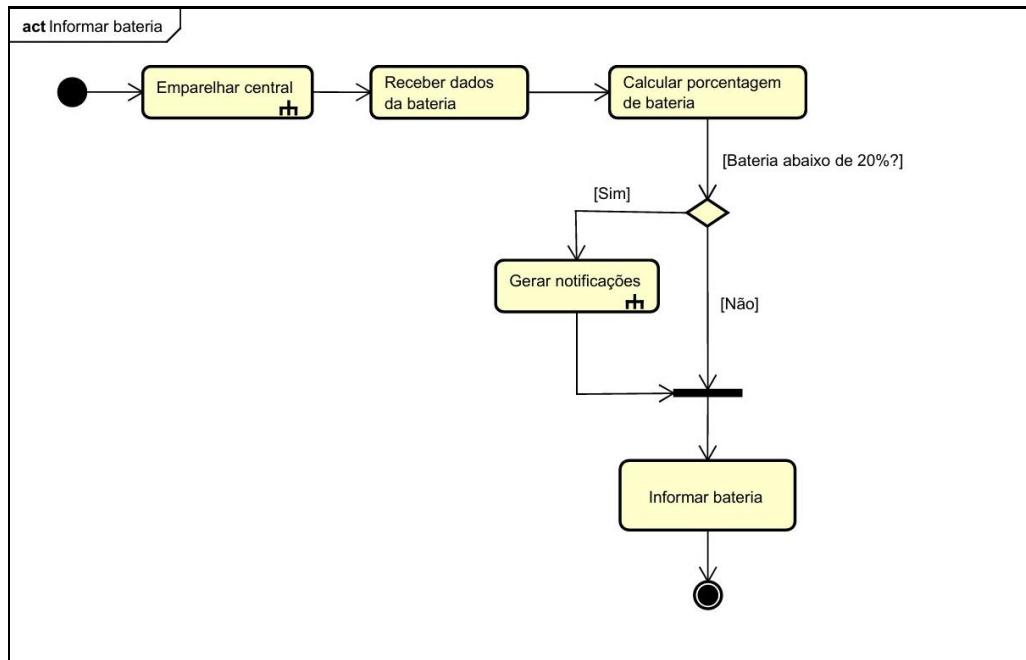
Fonte: De Autoria Própria, 2025.

Figura 34 – Diagrama de Atividade: Realizar Login



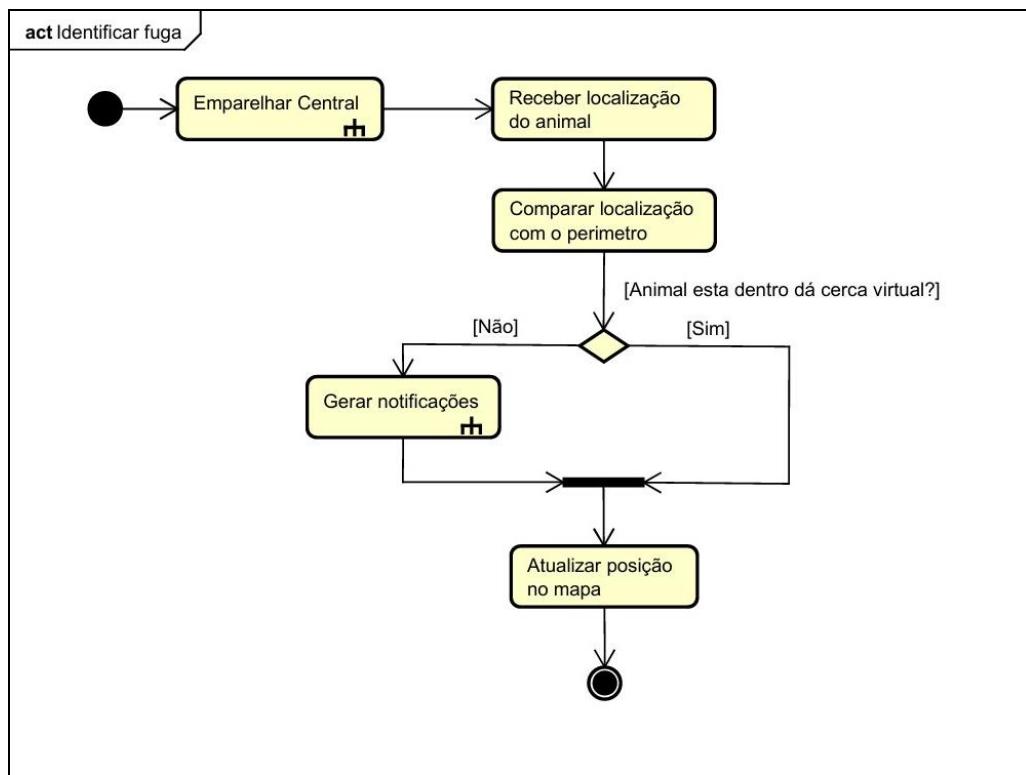
Fonte: De Autoria Própria, 2025.

Figura 35 – Diagrama de Atividade: Informar bateria



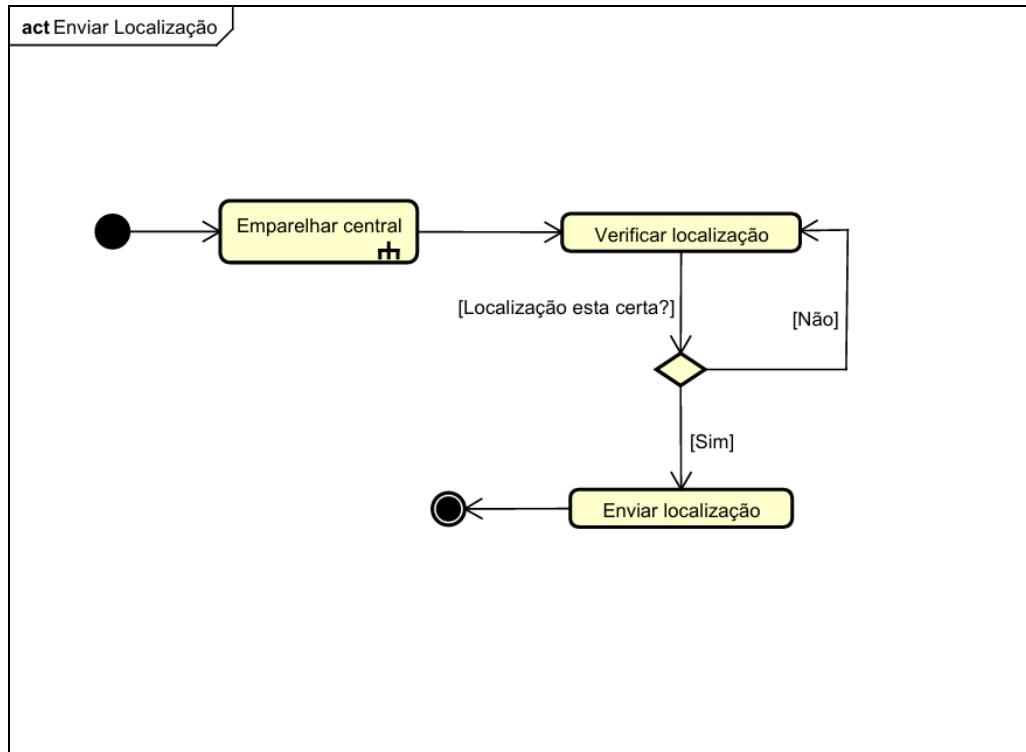
Fonte: De Autoria Própria, 2025.

Figura 36 – Diagrama de Atividade: Identificar Fuga



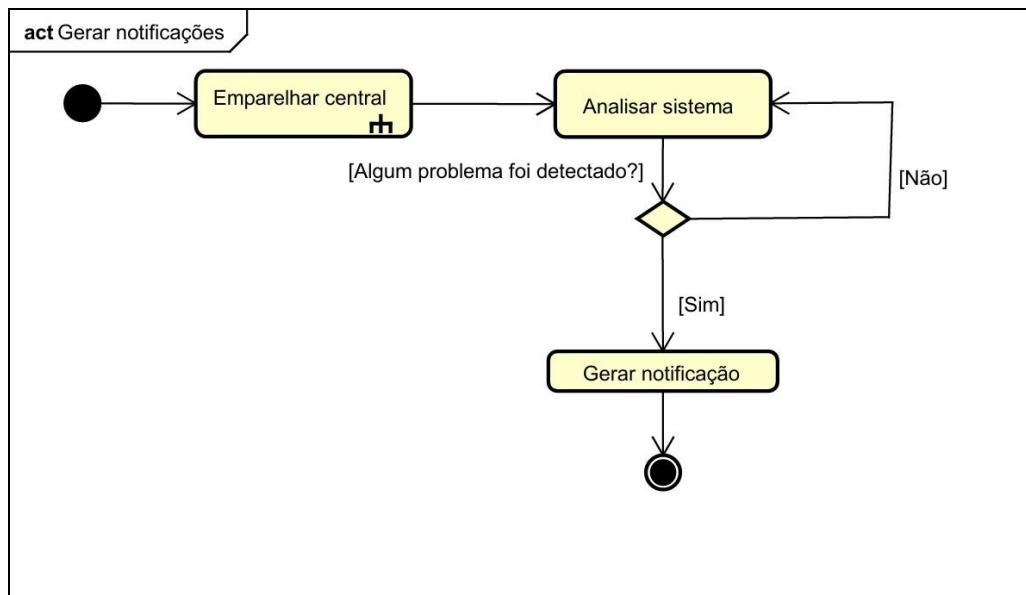
Fonte: De Autoria Própria, 2025.

Figura 37 – Diagrama de Atividade: Enviar Localização



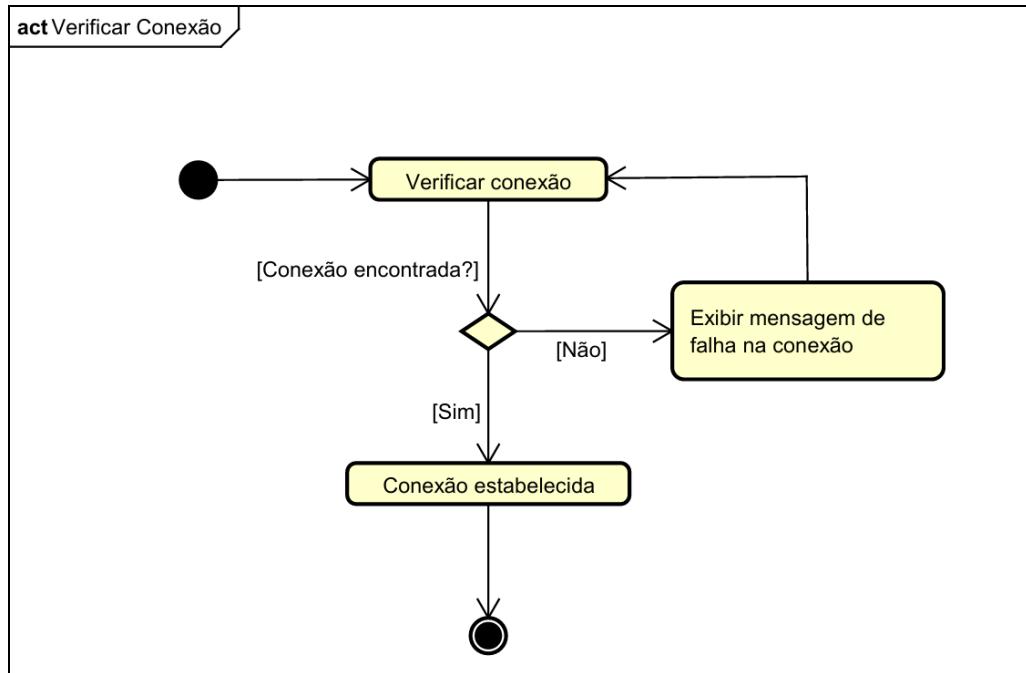
Fonte: De Autoria Própria, 2025.

Figura 38 – Diagrama de Atividade: Gerar Notificações



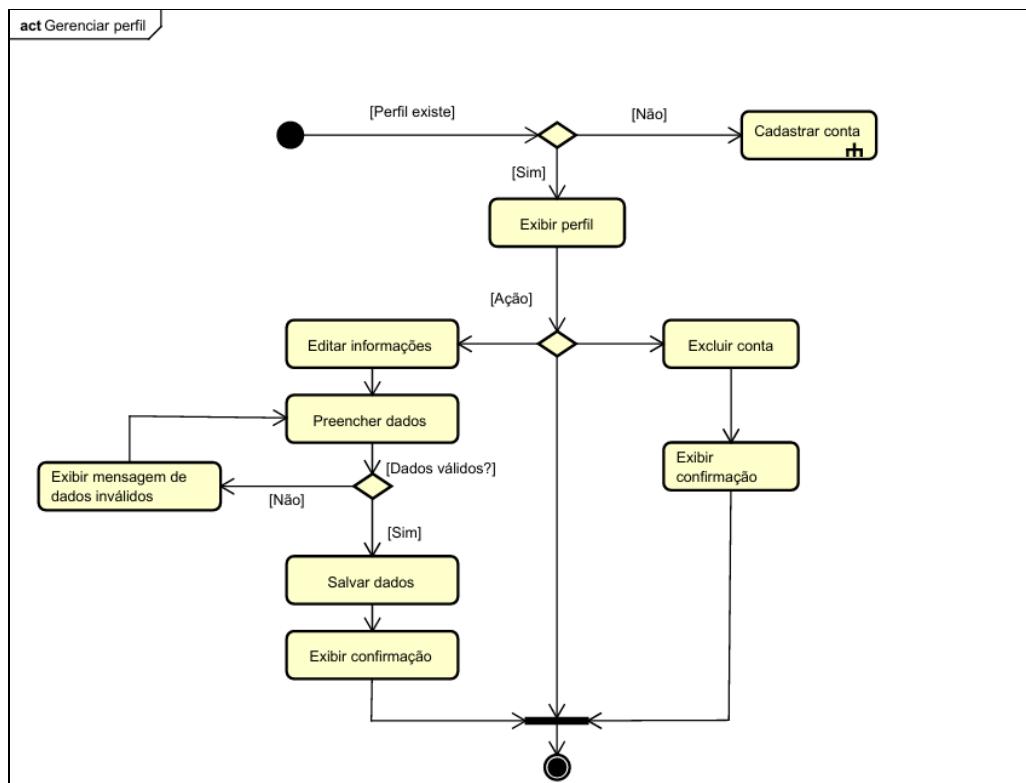
Fonte: De Autoria Própria, 2025.

Figura 39 – Diagrama de Atividade: Verificar Conexão



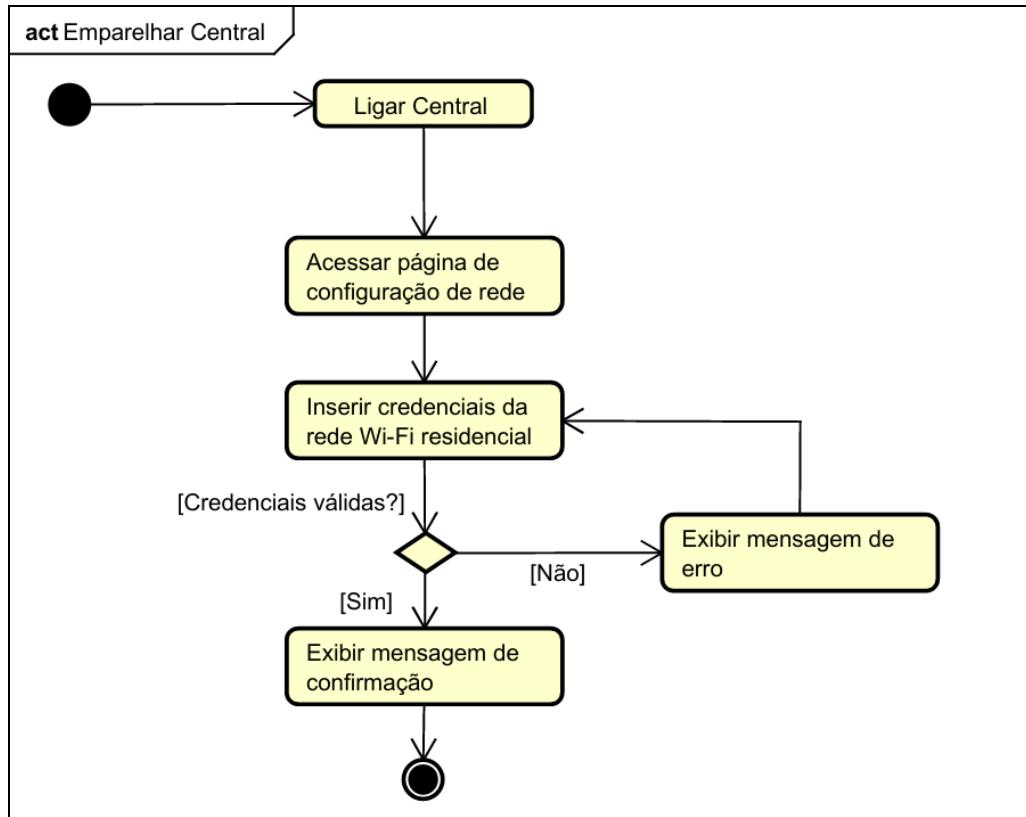
Fonte: De Autoria Própria, 2025.

Figura 40 – Diagrama de Atividade: Gerenciar Perfil



Fonte: De Autoria Própria, 2025.

Figura 41 – Diagrama de Atividade: Emparelhar Central

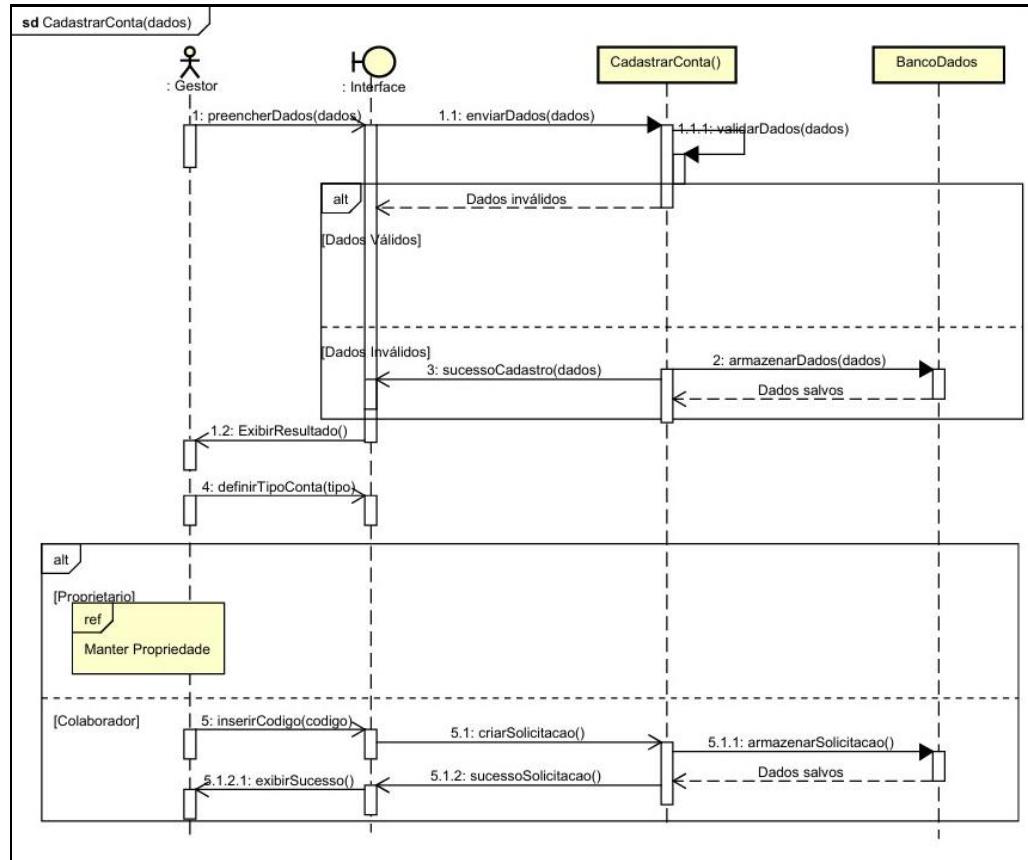


Fonte: De Autoria Própria, 2025.

### 3.4 Diagrama de Sequência

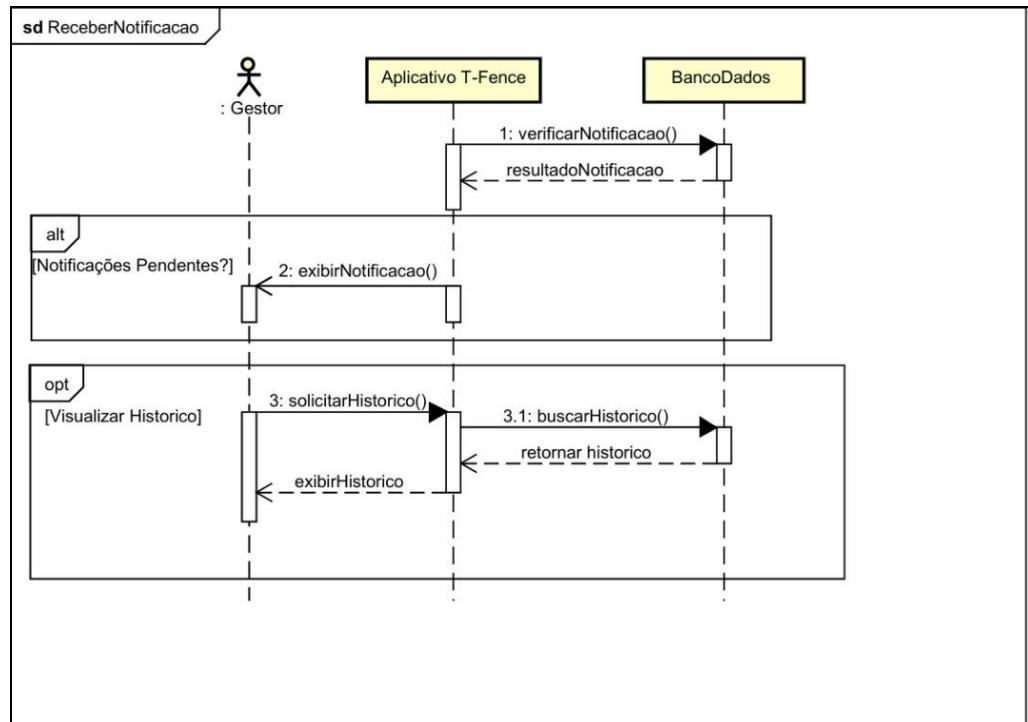
Os diagramas de sequência têm como objetivo detalhar com uma ordem temporal, a sequência de interações de um objeto durante sua linha de vida. Nesse diagrama é evidente a utilização do diagrama de casos de uso como base.

Figura 42 – Diagrama de Sequência: Cadastrar Conta



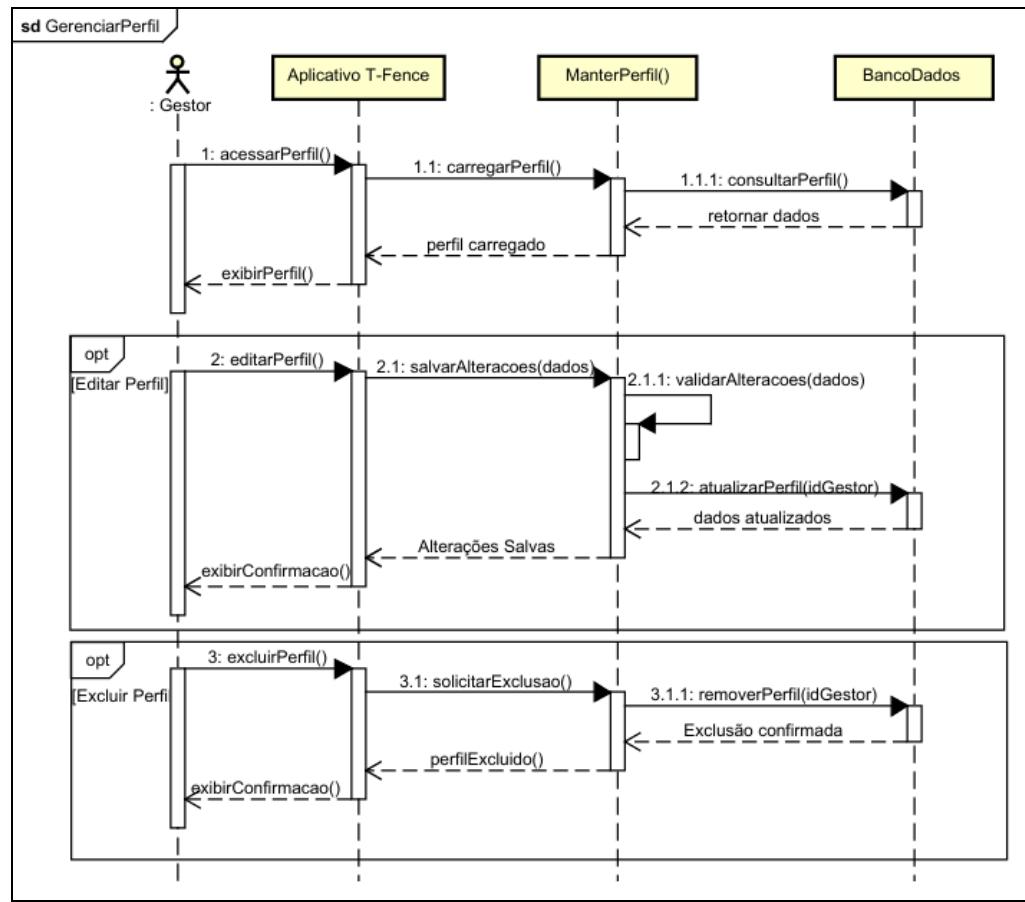
Fonte: De Autoria Própria, 2025.

Figura 43 – Diagrama de Sequência: Receber Notificação



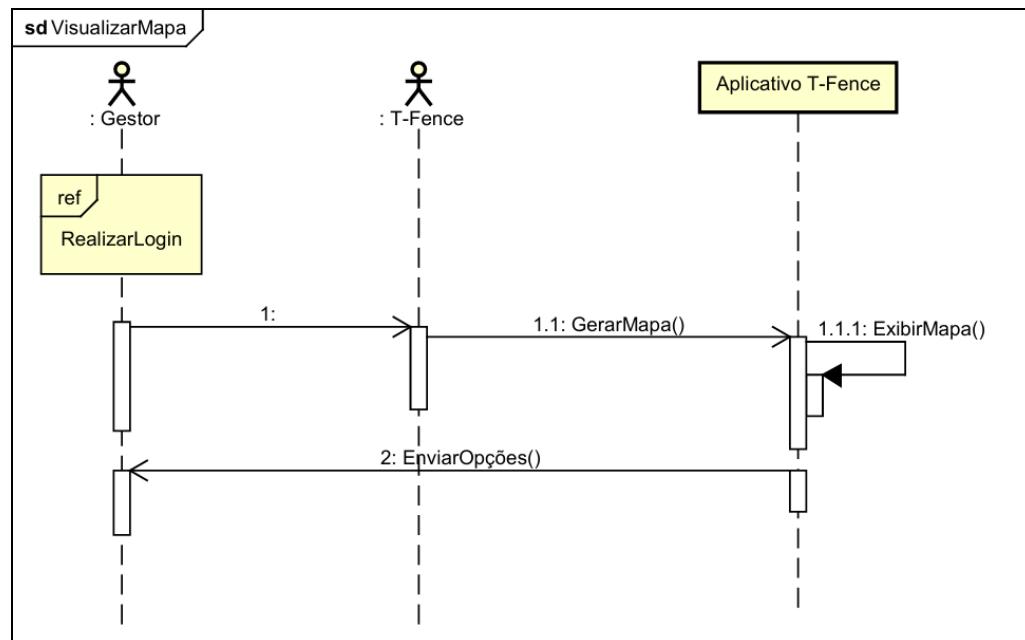
Fonte: De Autoria Própria, 2025.

Figura 44 – Diagrama de Sequência: Gerenciar Perfil



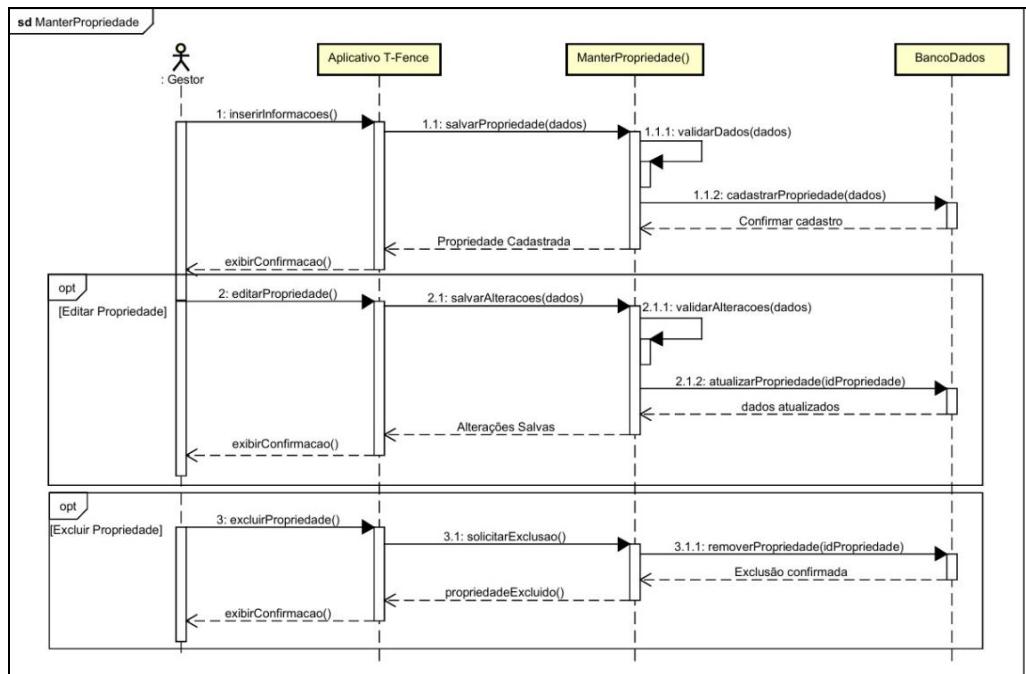
Fonte: De Autoria Própria, 2025.

Figura 45 – Diagrama de Sequência: Visualizar Mapa



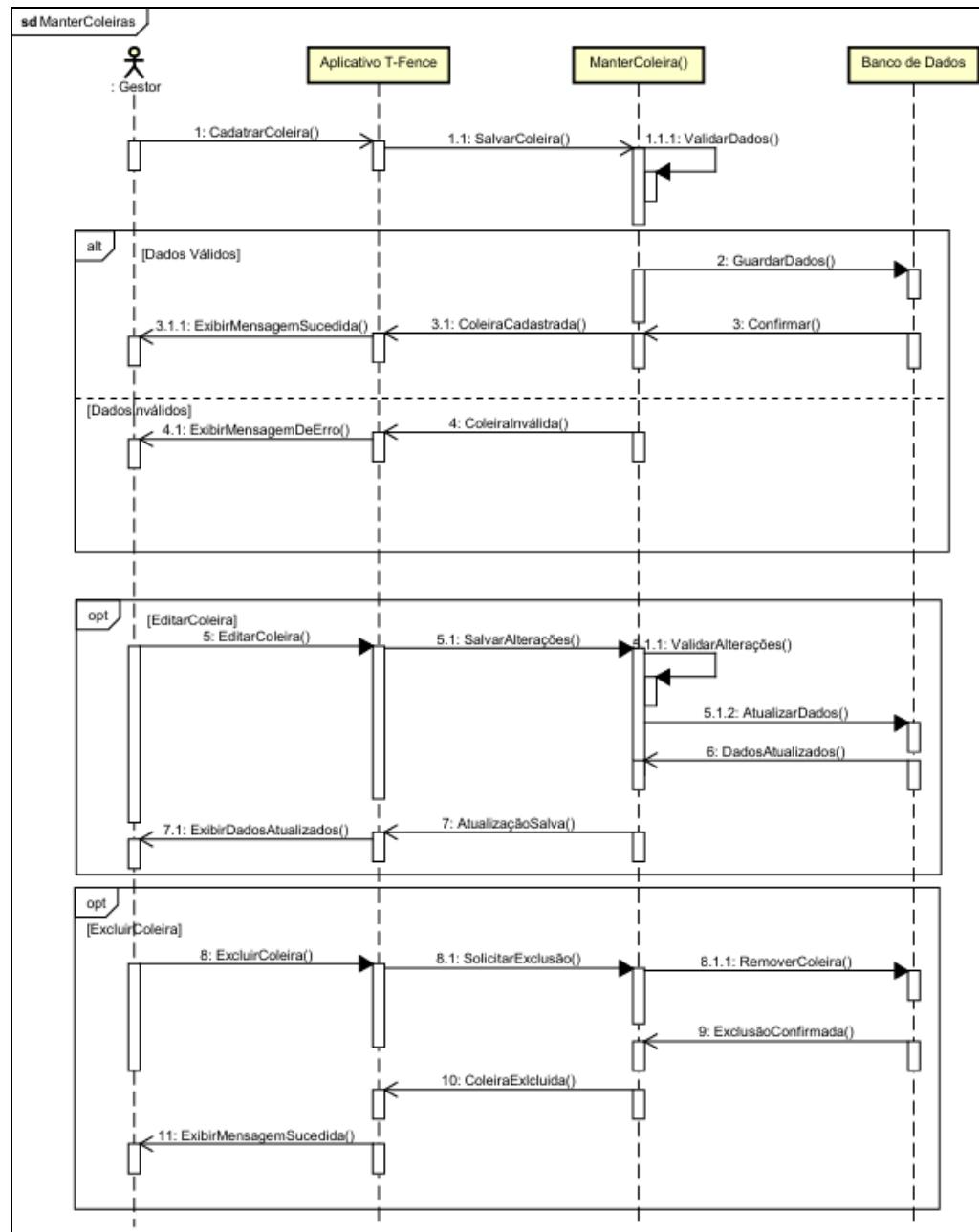
Fonte: De Autoria Própria, 2025.

Figura 46 – Diagrama de Sequência: Manter Propriedade



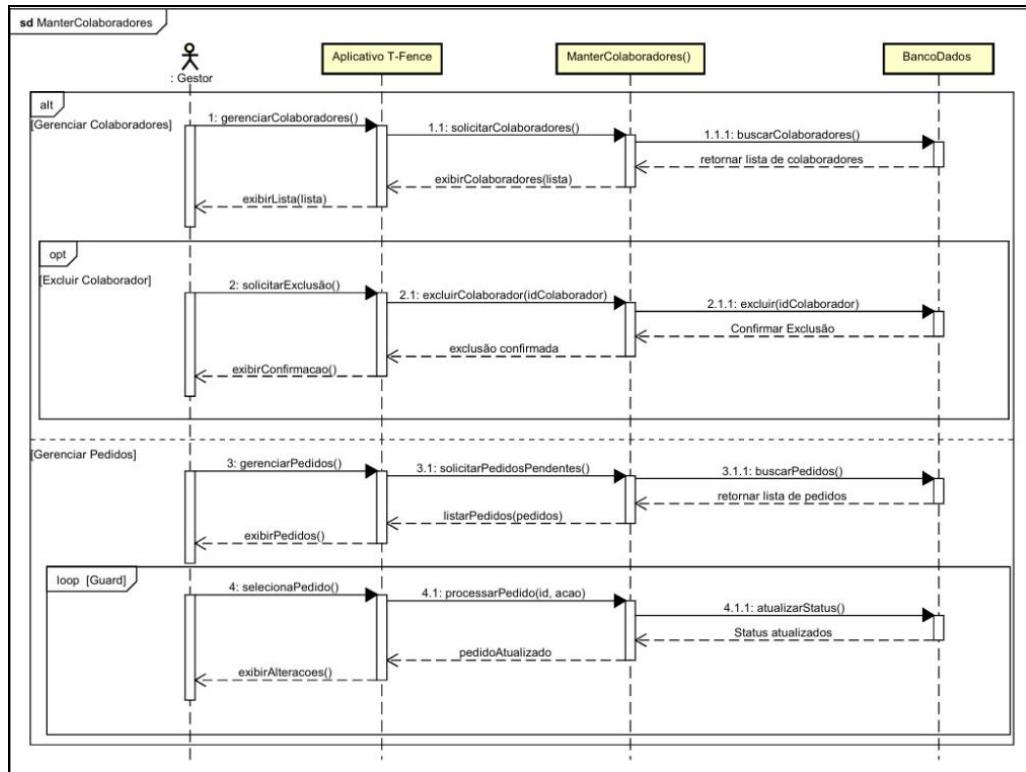
Fonte: De Autoria Própria, 2025.

Figura 47 – Diagrama de Sequência: Manter Coleiras



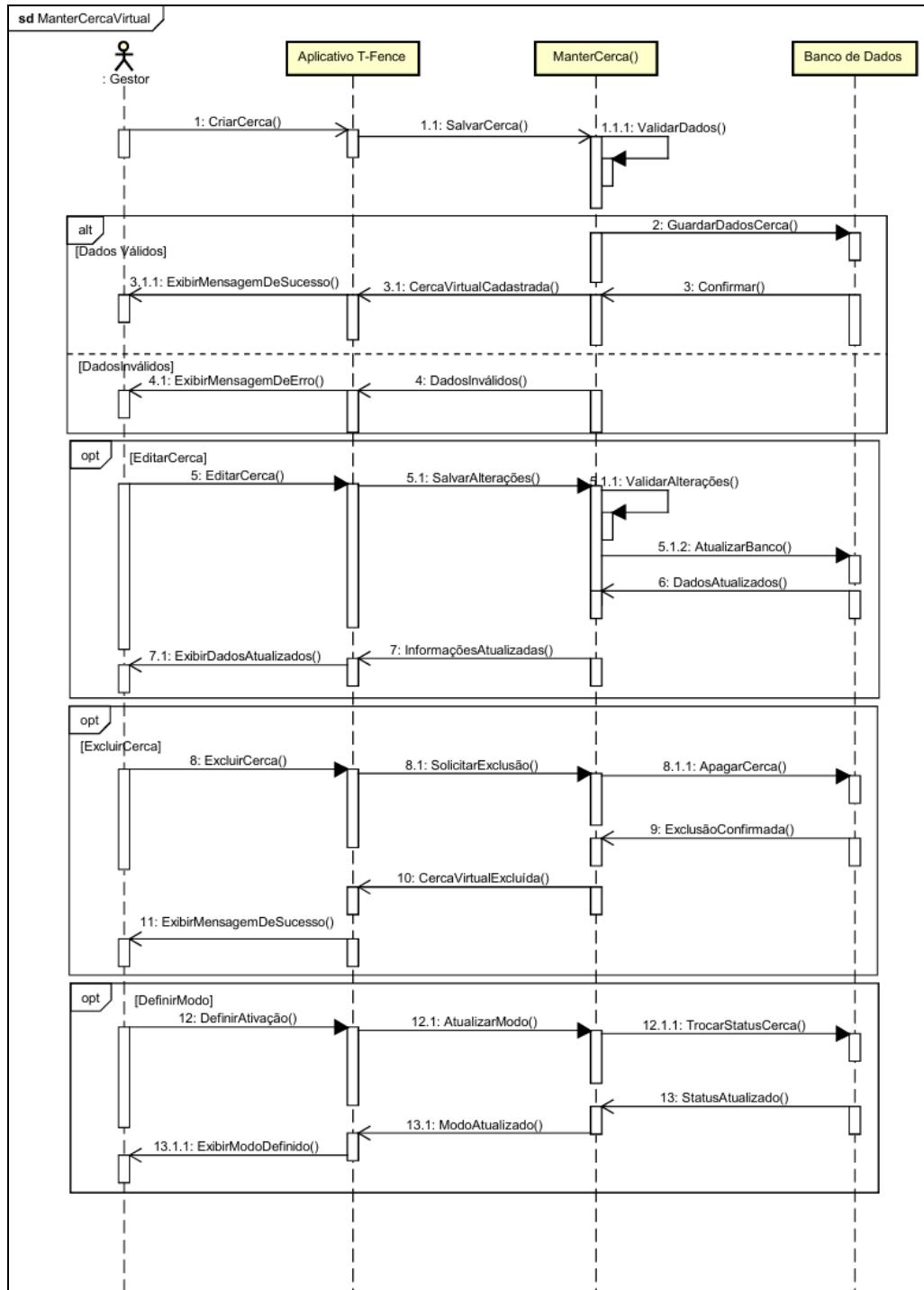
Fonte: De Autoria Própria, 2025.

Figura 48 – Diagrama de Sequência: Manter Colaboradores



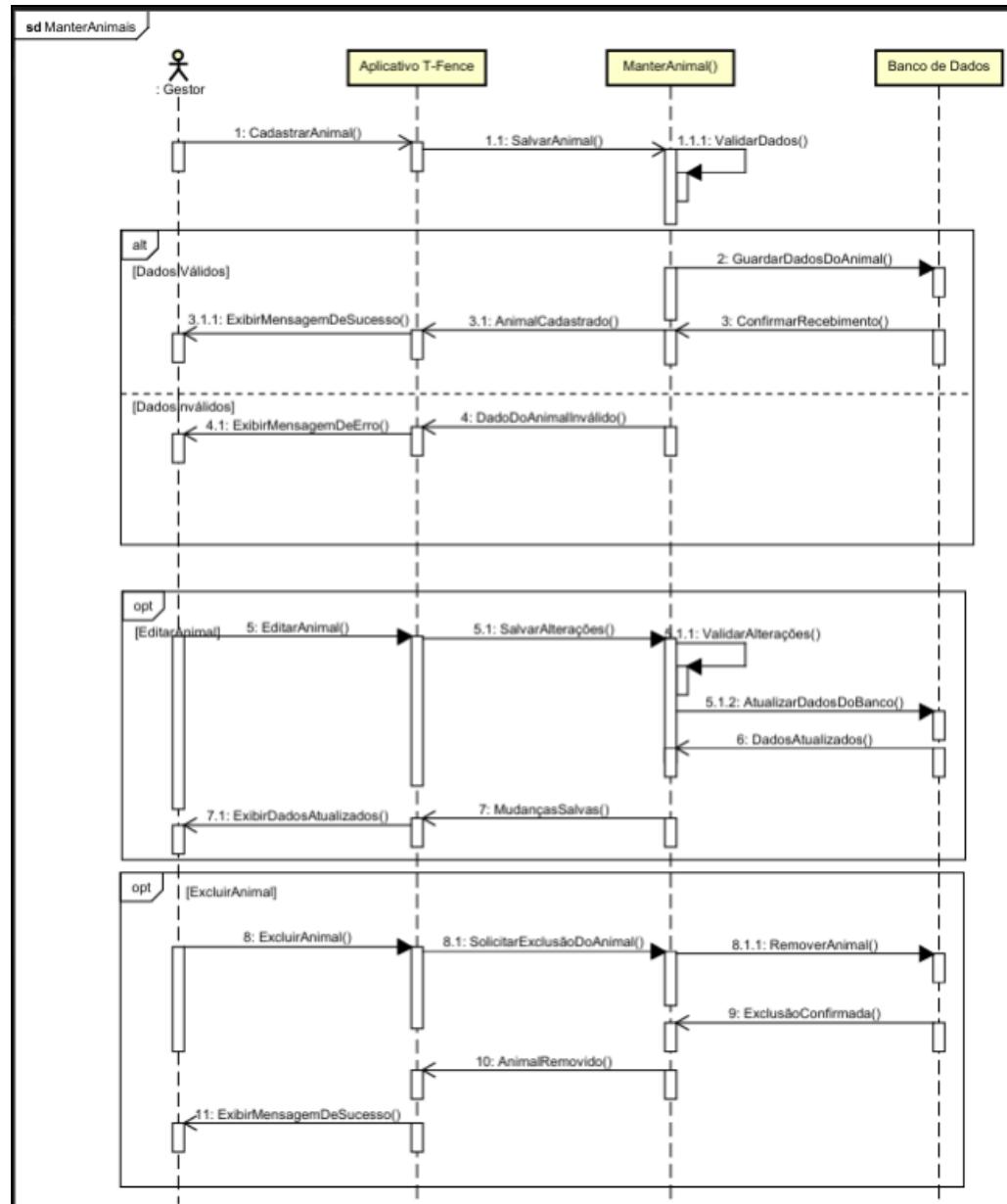
Fonte: De Autoria Própria, 2025.

Figura 49 – Diagrama de Sequência: Manter Cerca Virtual



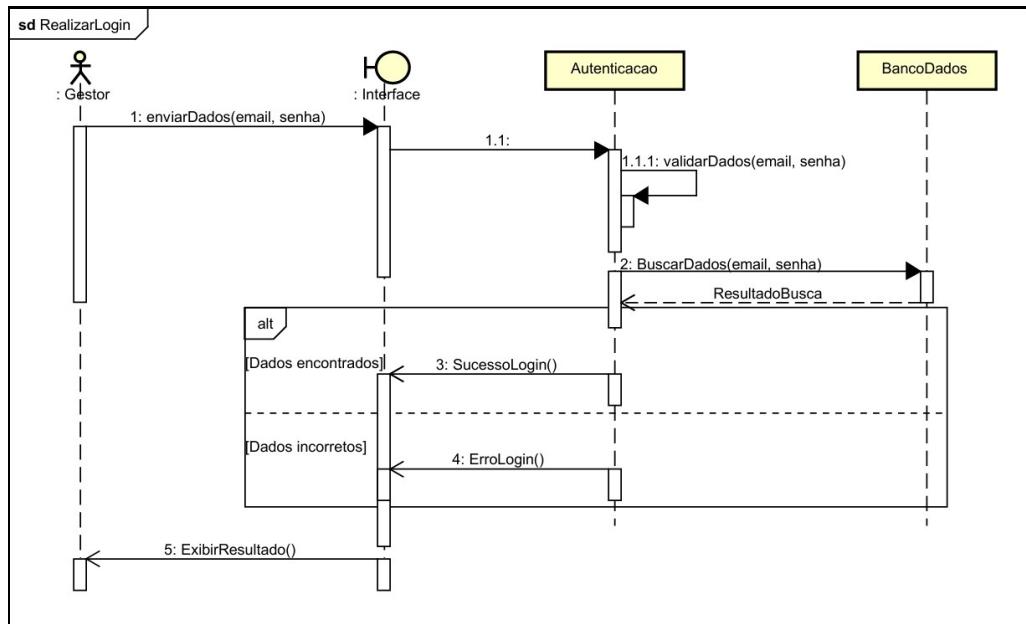
Fonte: De Autoria Própria, 2025.

Figura 50 – Diagrama de Sequência: Manter Animais



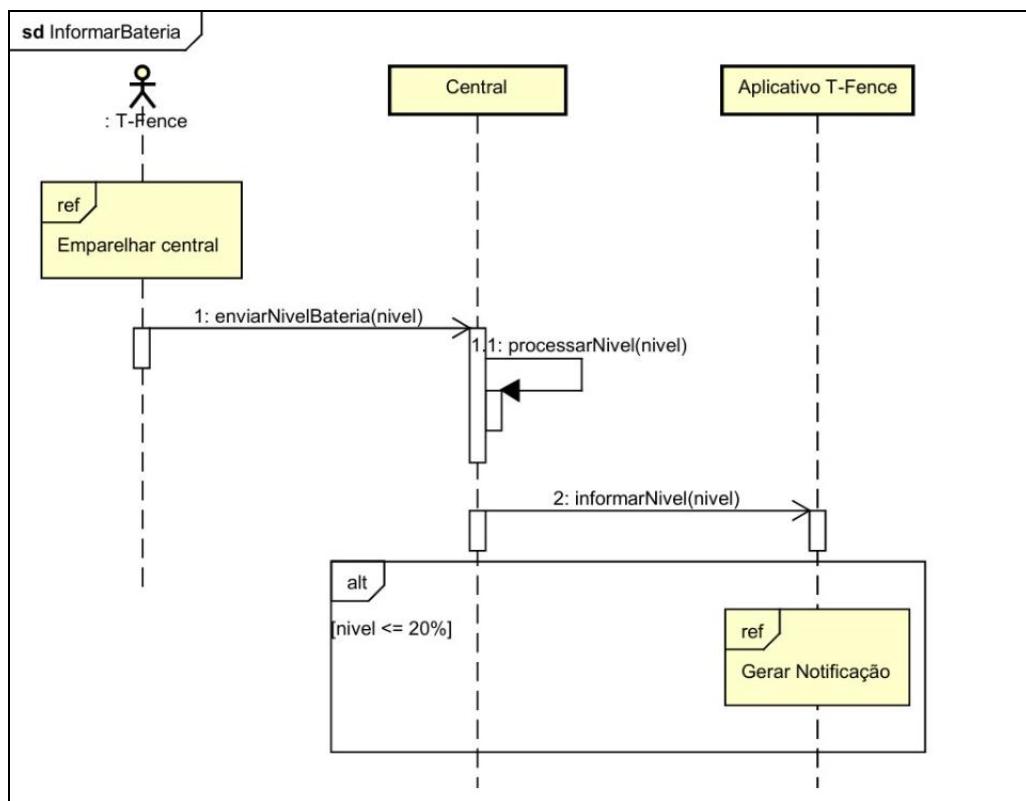
Fonte: De Autoria Própria, 2025.

Figura 51 – Diagrama de Sequência: Realizar Login



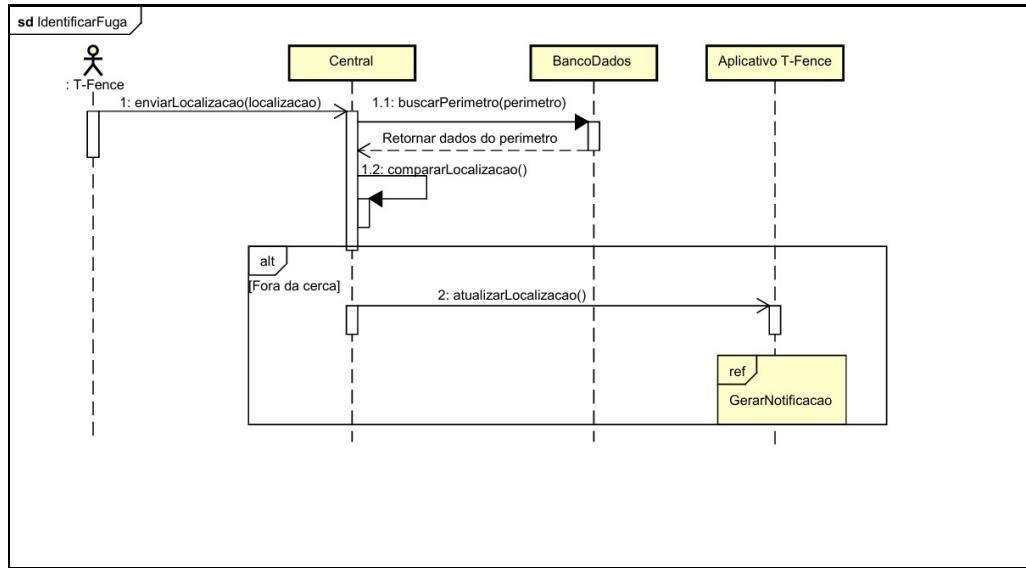
Fonte: De Autoria Própria, 2025.

Figura 52 – Diagrama de Sequência: Informar Bateria



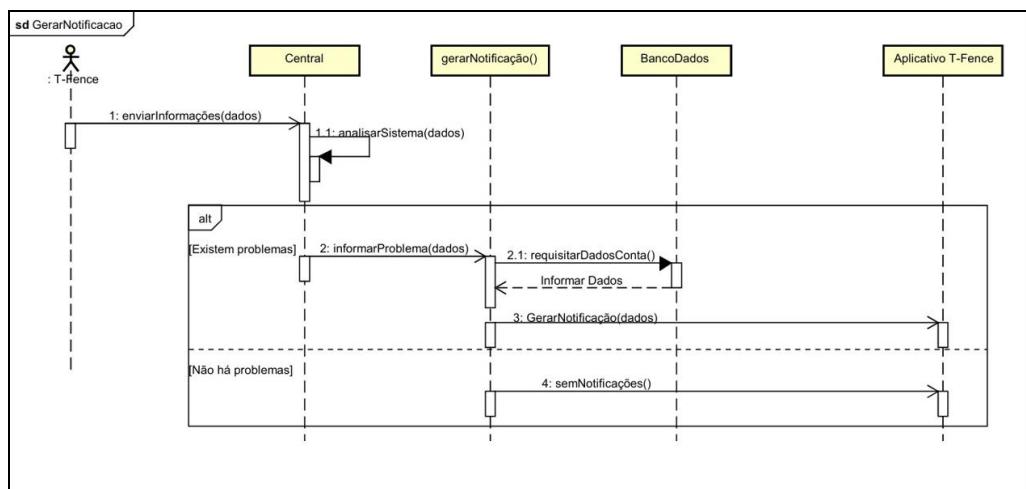
Fonte: De Autoria Própria, 2025.

Figura 53 – Diagrama de Sequência: Identificar Fuga



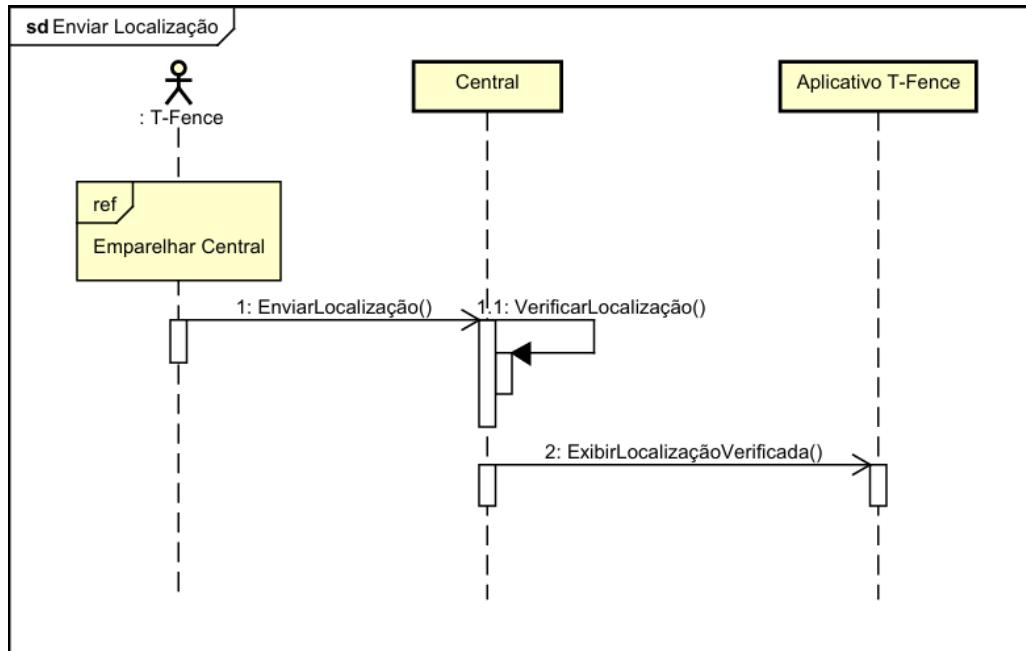
Fonte: De Autoria Própria, 2025.

Figura 54 – Diagrama de Sequência: Gerar Notificação



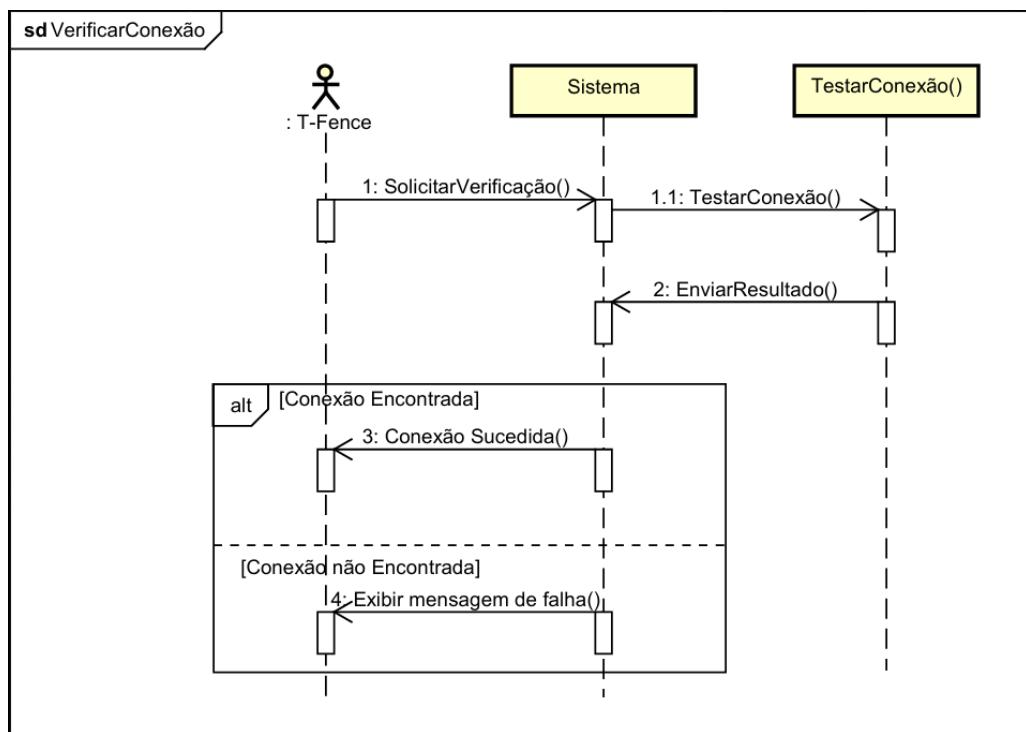
Fonte: De Autoria Própria, 2025.

Figura 55 – Diagrama de Sequência: Enviar Localização



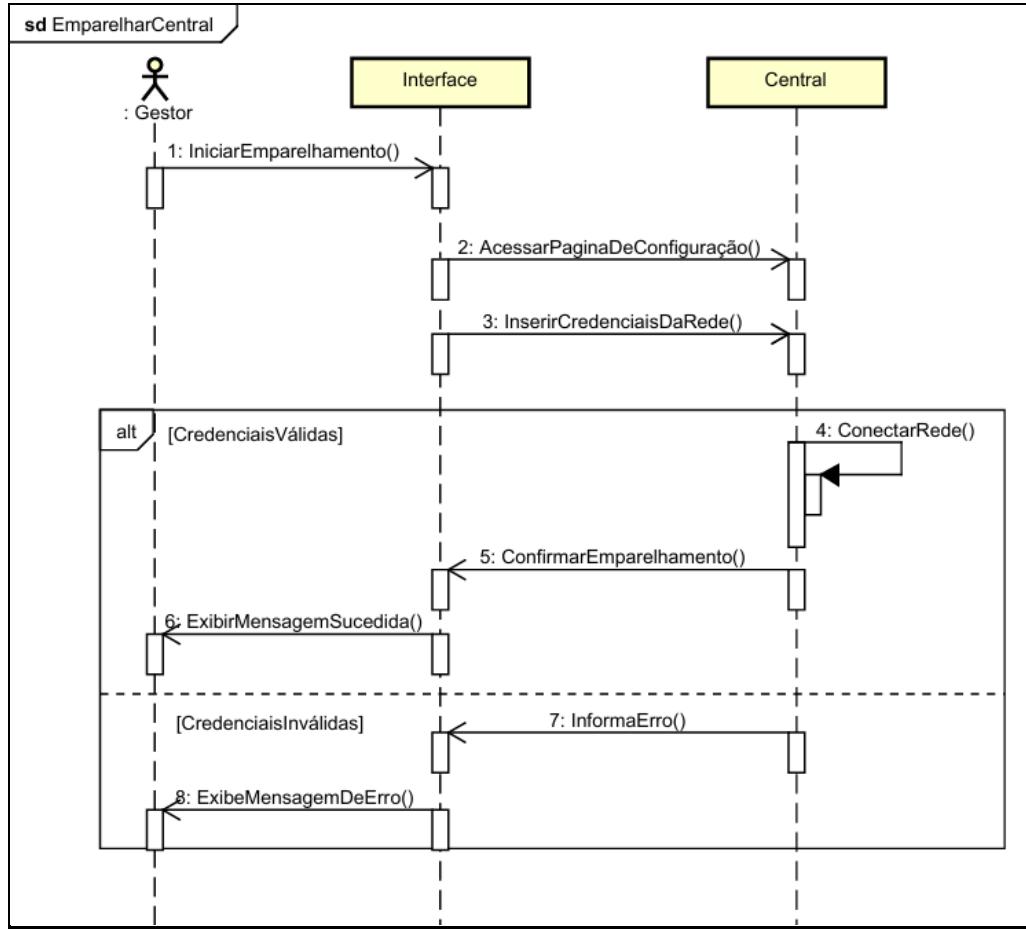
Fonte: De Autoria Própria, 2025.

Figura 56 – Diagrama de Sequência: Verificar Conexão



Fonte: De Autoria Própria, 2025.

Figura 57 – Diagrama de Sequência: Emparelhar Central

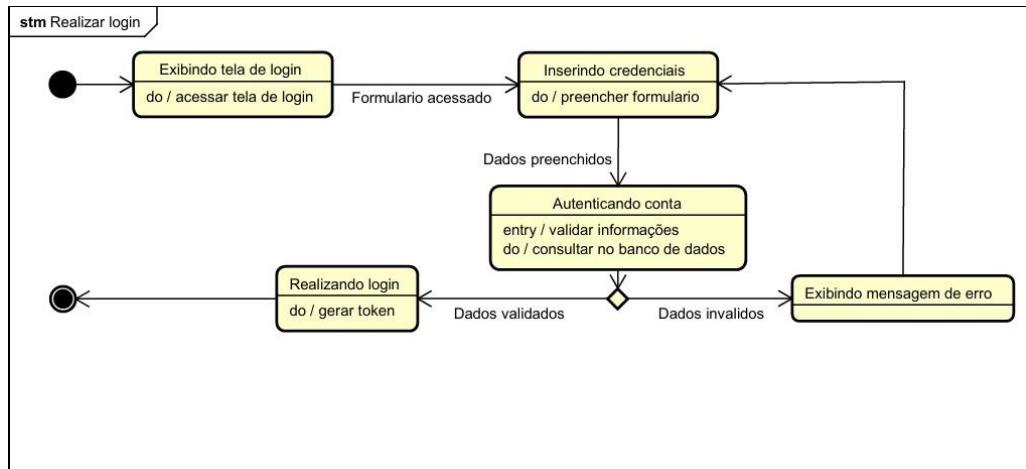


Fonte: De Autoria Própria, 2025.

### 3.5 Diagrama de Máquina de Estados

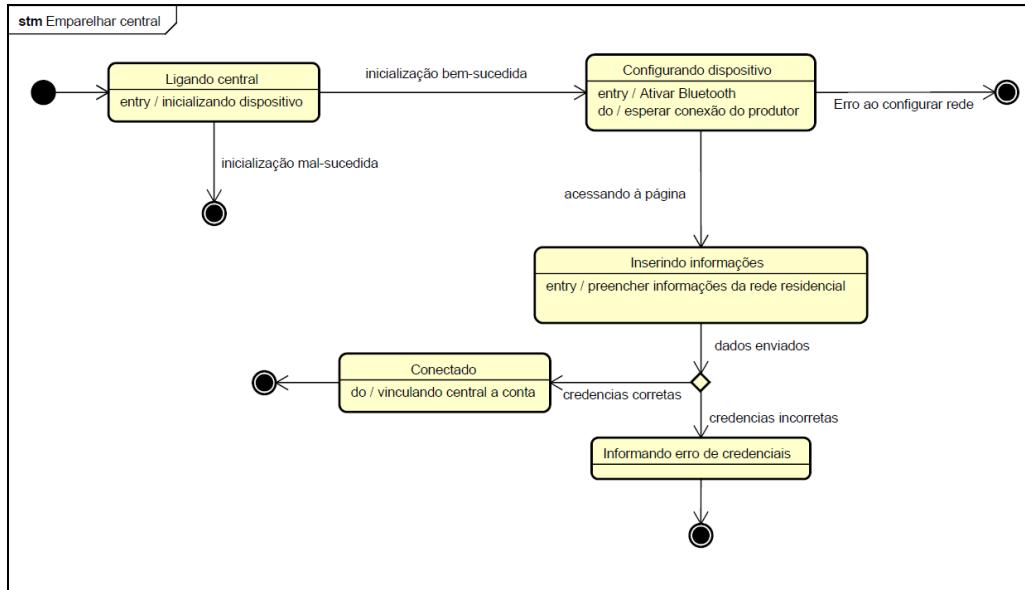
O diagrama de máquina de estado é utilizado para representar as alterações de estado de um elemento, por meio de eventos. Abaixo será apresentado os diagramas de máquina de estado do sistema T-Fence.

Figura 58 – Diagrama de Máquina de Estado: Realizar Login



Fonte: De Autoria Própria, 2025.

Figura 59 – Diagrama de Máquina de Estado: Emparelhar Central



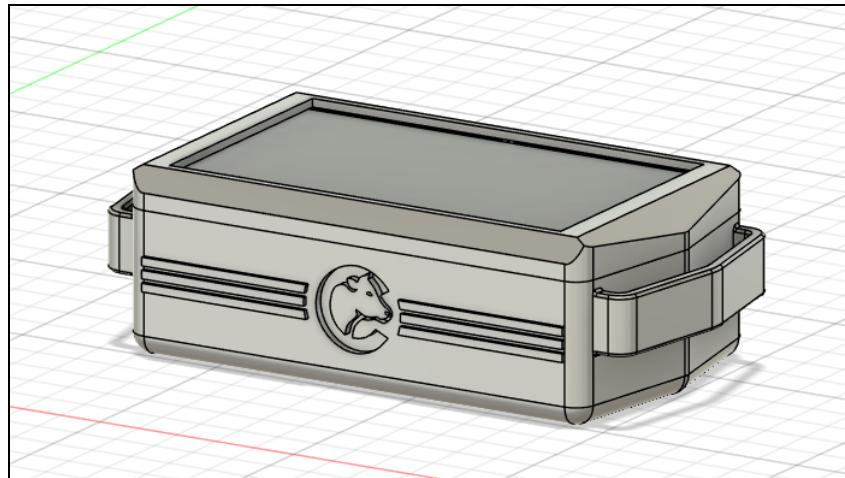
Fonte: De Autoria Própria, 2025.

### 3.6 Impressão 3D

Nesse capítulo será apresentada a modelagem e impressão 3D dos dispositivos que compõe o projeto. O sistema T-Fence necessitou da modelagem de dois modelos tridimensionais para atender as necessidades do projeto, sendo eles: Uma coleira localizada ao entorno do pescoço do animal e um dispositivo na residência do proprietário para transmitir as informações até a base de dados.

O primeiro modelo elaborado foi o da coleira. Sua estrutura foi pensada para ser o mais compacta possível, visando garantir o conforto do animal durante o uso, ela possui alças para o cinto, um compartimento interno destinado à acomodação do microcontrolador ESP32 e de seus módulos, como o INA219, módulo GPS, módulo LoRa e Bateria. Além disso, a tampa da coleira foi projetada para comportar um painel solar essencial para garantir sua autonomia energética.

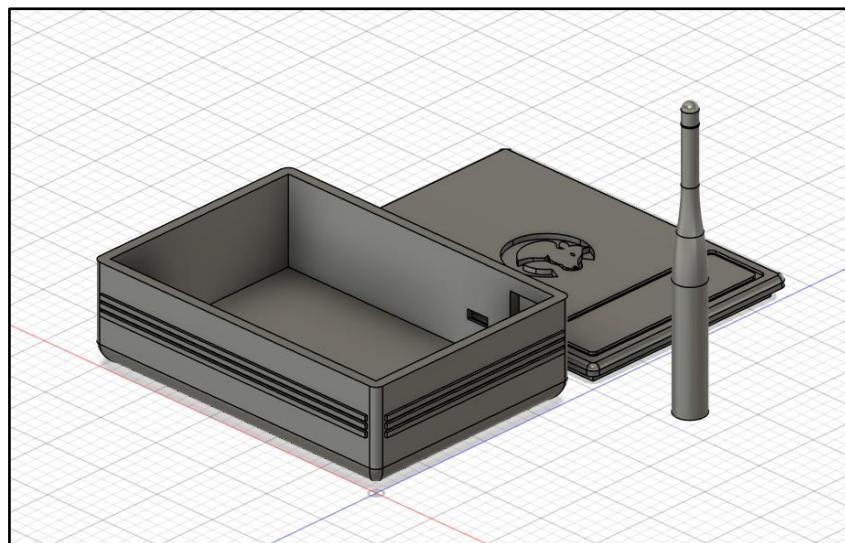
Figura 60 – Protótipo Final Coleira T-Fence



Fonte: De Autoria Própria, 2025

Já o segundo modelo corresponde à central receptora, diferentemente da coleira, trata-se de um modelo mais simples. Ele possui um compartimento interno destinado aos módulos ESP32, módulo LoRa, módulo de carregamento, e duas cavidades: uma para o botão liga/desliga, e outra para a entrada do carregador USB.

Figura 61 – Protótipo Final Central Receptora



Fonte: De Autoria Própria, 2025

### 3.7 Montagem do circuito dos dispositivos IoT

O sistema desenvolvido tem como objetivo o monitoramento dos animais por meio de sua localização aproximada, permitindo acompanhar sua movimentação e fornecer informações que auxiliem na identificação de possíveis fugas ou furtos. Para possibilitar esse funcionamento, o sistema foi dividido em dois dispositivos principais:

a coleira, responsável pela coleta e transmissão de informações e a central encarregada de receber as informações e enviá-las para um banco de dados, permitindo que possam ser acessadas em um aplicativo pelo criador rural.

O primeiro circuito corresponde à coleira, que foi projetada com a finalidade de capturar a localização e transmitir esses dados para a central.

Para isso foi utilizado o ESP32, microcontrolador responsável por executar toda a lógica do circuito, garantindo o gerenciamento dos demais módulos bem como o processamento dos dados fornecidos.

Com o objetivo de obter a localização, a coleira conta com um módulo Neo-M8N, que é capaz de utilizar múltiplos sistemas de navegação por satélite, proporcionando maior precisão geográfica da coleira.

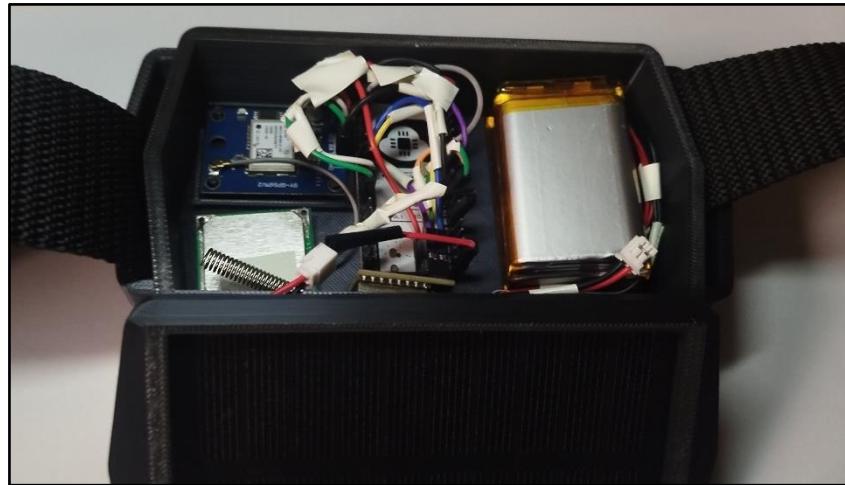
Para viabilizar a transmissão dos dados foi escolhido um módulo LoRa, tecnologia caracterizada pelo seu longo alcance de transmissão via rádio e baixo consumo de energia, ideal para nossa aplicação.

Por se tratar de um dispositivo móvel, ele foi equipado com uma bateria de longa duração e um painel solar permitindo autonomia energética, desta forma reduzindo a necessidade de carregamento manual.

O módulo de carregamento CN3065 é componente essencial do projeto, sendo responsável por fazer o carregamento eficiente das baterias por meio da energia fornecida pelo painel solar.

Além disso foi utilizado o módulo MCU-219, que tem como função medir a voltagem da bateria, sendo fundamental para estimar a medição da porcentagem da bateria. Desta forma o sistema pode identificar baixo nível de bateria e enviar notificações para o produtor.

Figura 62 – Circuito da Coleira



Fonte: De Autoria Própria, 2025

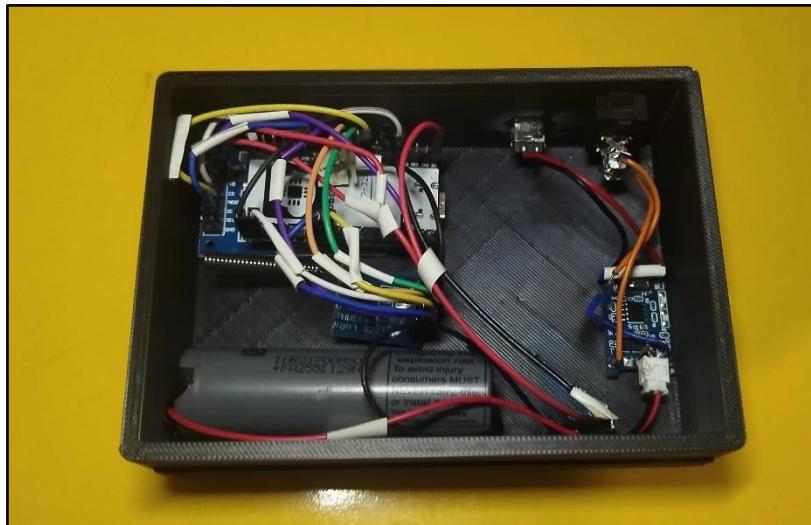
Já o segundo circuito corresponde a central, responsável por receber os dados enviados pela coleira e enviá-los para um servidor.

Neste dispositivo, o microcontrolador principal também é o ESP32, atuando como parte central, gerenciando os módulos, além do mais ele possui conectividade com Wi-Fi, que é essencial para permitir enviar dados pela internet.

Com o objetivo de permitir a comunicação entre coleira e central, foi utilizado um segundo módulo LoRa, encarregada de receber os dados transmitidos pela coleira.

Para a alimentação primária do circuito, foi utilizado uma porta USB-C, permitindo conexão direta a uma fonte de energia externa. Como alimentação secundária foi adicionada uma bateria de 5000mAh junto a um módulo de carregamento TP4056, responsável por garantir um carregamento seguro evitando descarga total e sobrecarga. Além disso ele possui um botão liga/desliga, possibilitando controle manual do funcionamento do dispositivo.

Figura 63 – Circuito da Central



Fonte: De Autoria Própria, 2025

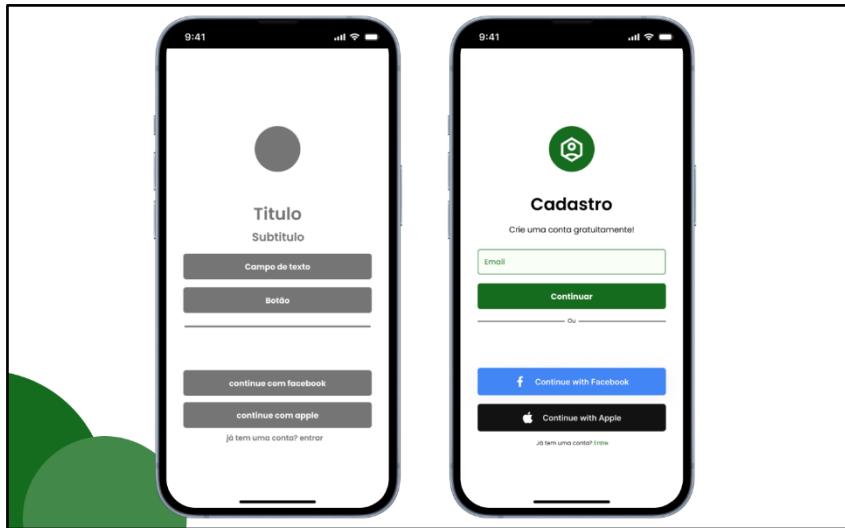
### 3.8 Prototipação das telas

Nesse tópico, será apresentado os wireframes de alta e baixa fidelidade referentes às principais telas do aplicativo desenvolvido para o sistema IoT de monitoramento do rebanho. Essas representações têm como foco mostrar a estrutura da aplicação móvel e como o usuário interage com as informações. O aplicativo foi planejado para entregar uma experiência intuitiva, facilitando ao produtor realizar as tarefas desejadas.

Abaixo, estão, respectivamente, os wireframes de baixa e alta, posicionados lado a lado, facilitando a comparação entre as versões iniciais e finais propostas.

As primeiras telas a serem apresentadas são as telas de Cadastro. Nela o produtor terá que criar uma conta no aplicativo e cadastrar suas informações como Email e senha. A primeira tela é focada em apenas colocar o Email.

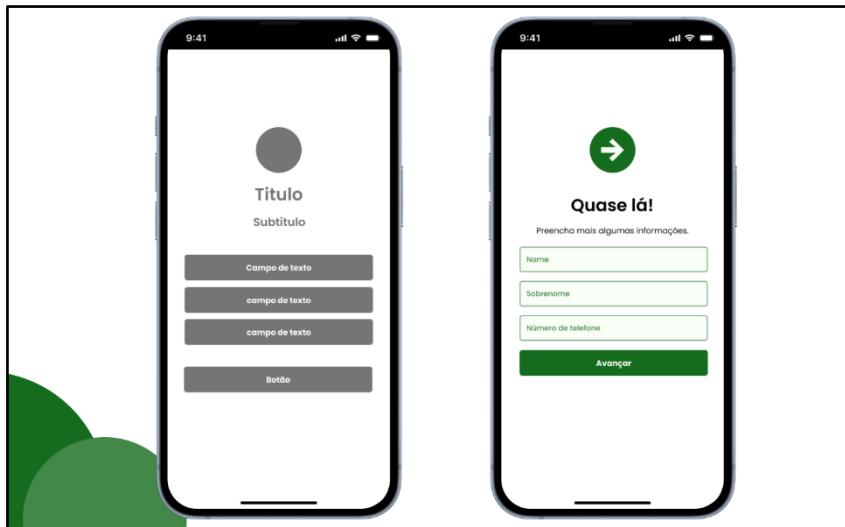
Figura 64 – Wireframes de baixa e alta fidelidade do Cadastro



Fonte: De Autoria Própria, 2025

Em seguida desta página, está o segundo passo do cadastro. O produtor terá que inserir todas as novas informações requeridas pela tela como nome, sobrenome e telefone.

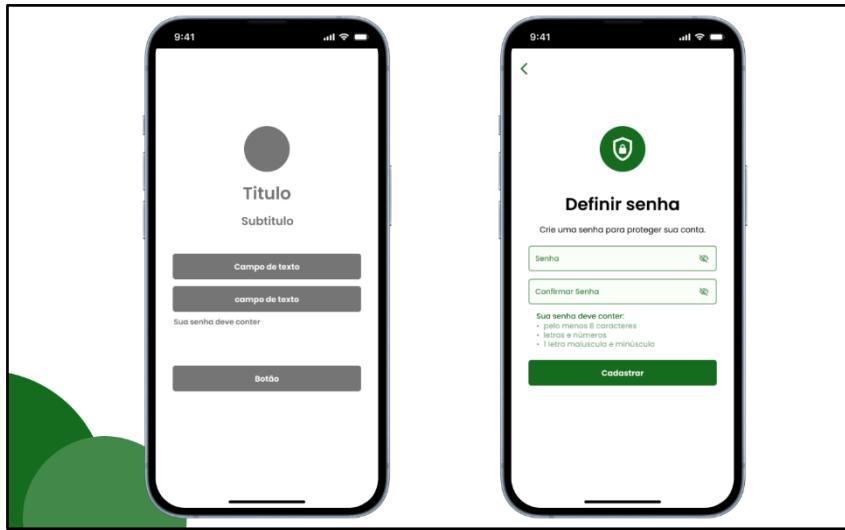
Figura 65 – Wireframes de baixa e alta fidelidade da segunda etapa do Cadastro



Fonte: De Autoria Própria, 2025

Na próxima interface, o usuário terá que estabelecer sua senha, se atentando para os requisitos para que seja uma senha válida

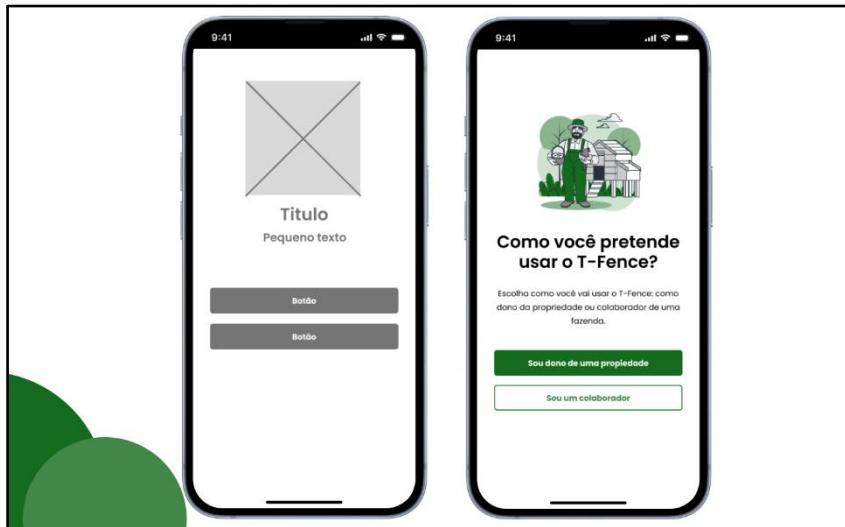
Figura 66 – Wireframes de baixa e alta fidelidade da terceira etapa do Cadastro



Fonte: De Autoria Própria, 2025

Em seguida, o usuário será redirecionado a uma página onde terá que escolher entre duas opções: Colaborador ou Proprietário.

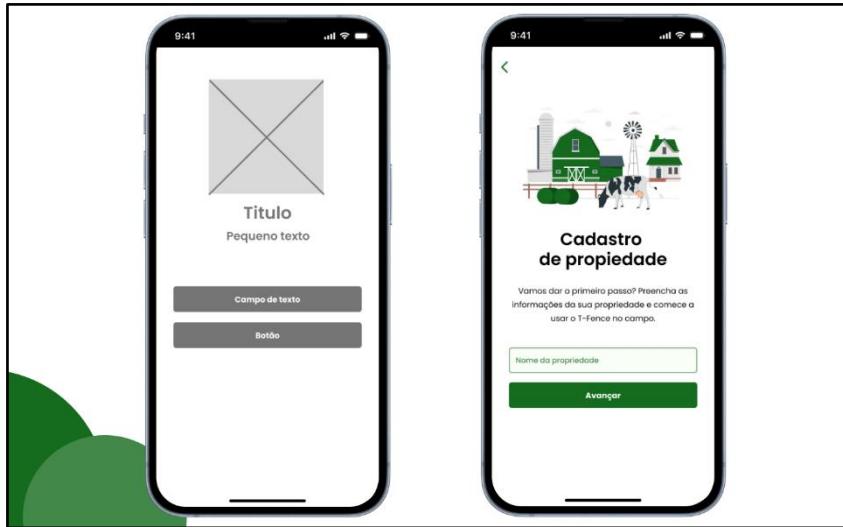
Figura 67 – Wireframes de baixa e alta fidelidade da quarta etapa do Cadastro



Fonte: De Autoria Própria, 2025

Caso o usuário seja proprietário, ele será redirecionado para uma tela no qual terá que cadastrar informações de sua propriedade.

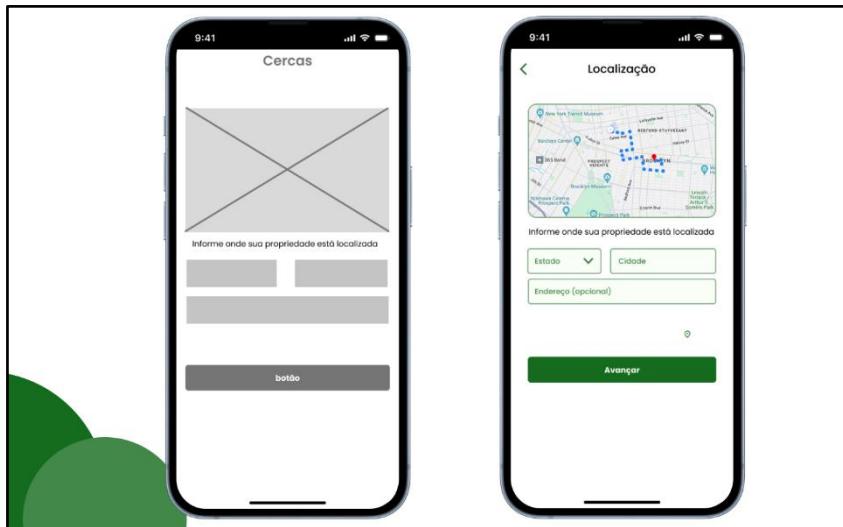
Figura 68 – Wireframes de baixa e alta fidelidade da quinta etapa do Cadastro



Fonte: De Autoria Própria, 2025

Em seguida, é exibida uma continuação da página anterior, solicitando mais algumas informações referentes a localização do terreno.

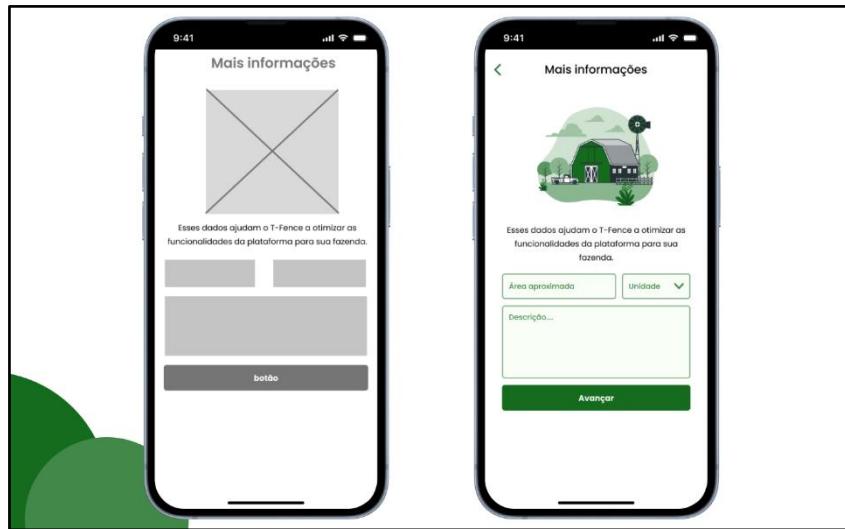
Figura 69 – Wireframes de baixa e alta fidelidade da sexta etapa do Cadastro



Fonte: De Autoria Própria, 2025

Continuando a tela anterior, essa interface exige informações voltadas ao tamanho e descrição da propriedade.

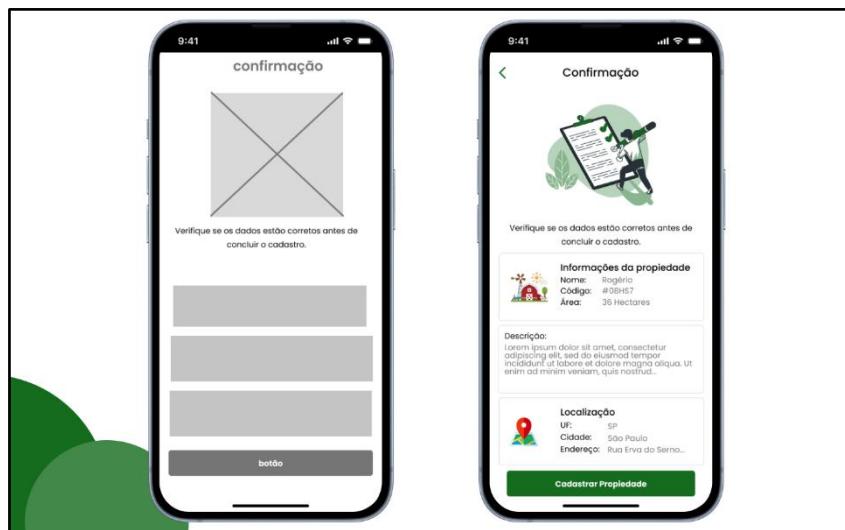
Figura 70 – Wireframes de baixa e alta fidelidade da sétima etapa do Cadastro



Fonte: De Autoria Própria, 2025

Por fim, a última tela de cadastro para o colaborador. Nessa página o usuário pode visualizar um pequeno resumo e conferir se os dados colocados estão corretos.

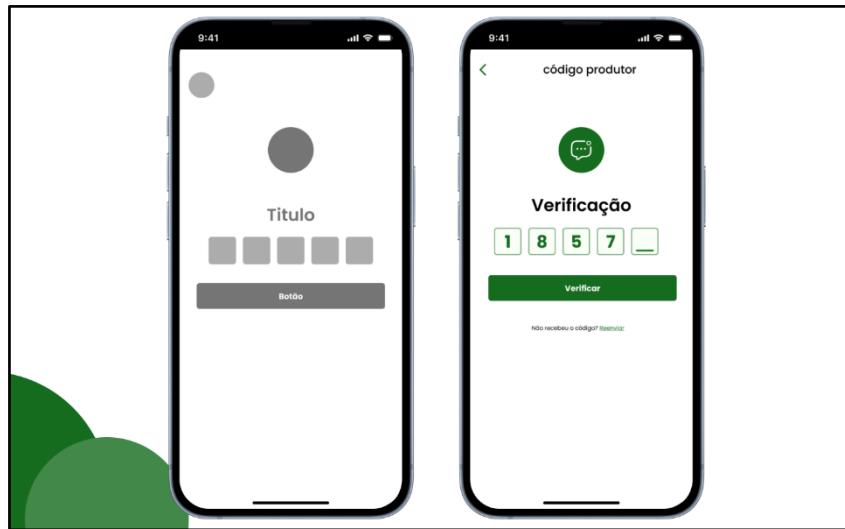
Figura 71 – Wireframes de baixa e alta fidelidade da sétima etapa do Cadastro



Fonte: De Autoria Própria, 2025

Caso o usuário seja colaborador, ele será encaminhado para uma página no qual exige um código de um produtor para ter acesso a propriedade desejada.

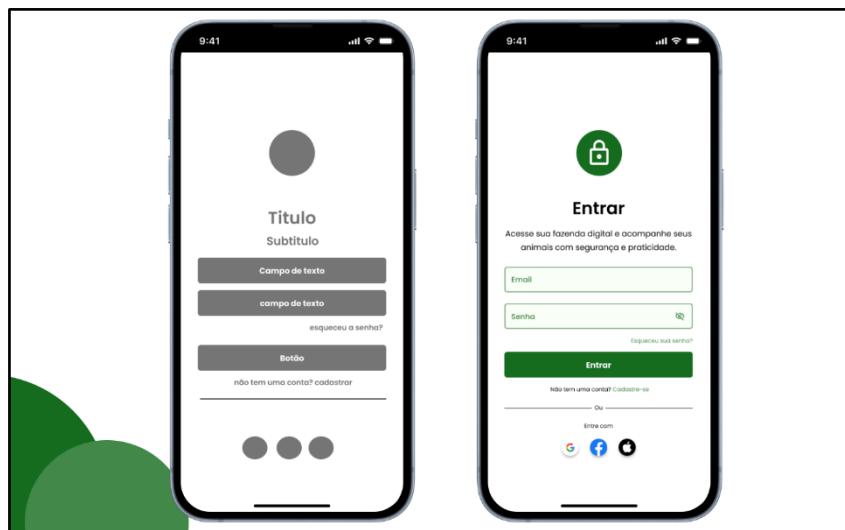
Figura 72 – Wireframes de baixa e alta fidelidade da última etapa do Cadastro



Fonte: De Autoria Própria, 2025

Segue abaixo a tela de Login. Ela é utilizada para o usuário pode logar após colocar suas devidas informações na tela.

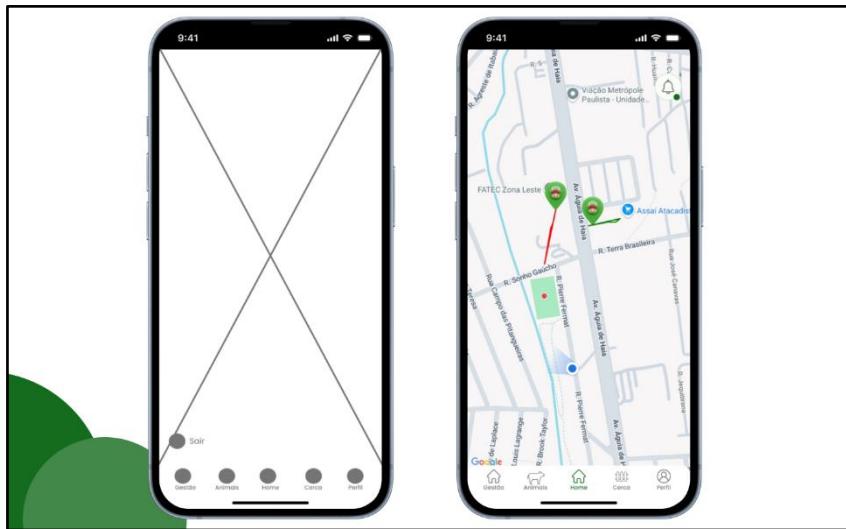
Figura 73 – Wireframes de baixa e alta fidelidade do Login



Fonte: De Autoria Própria, 2025

Já com a propriedade cadastrada, será exibido uma tela com um mapa referente a localização desejada, um menu na parte inferior da página com outras 4 opções e um botão à direita superior do mapa.

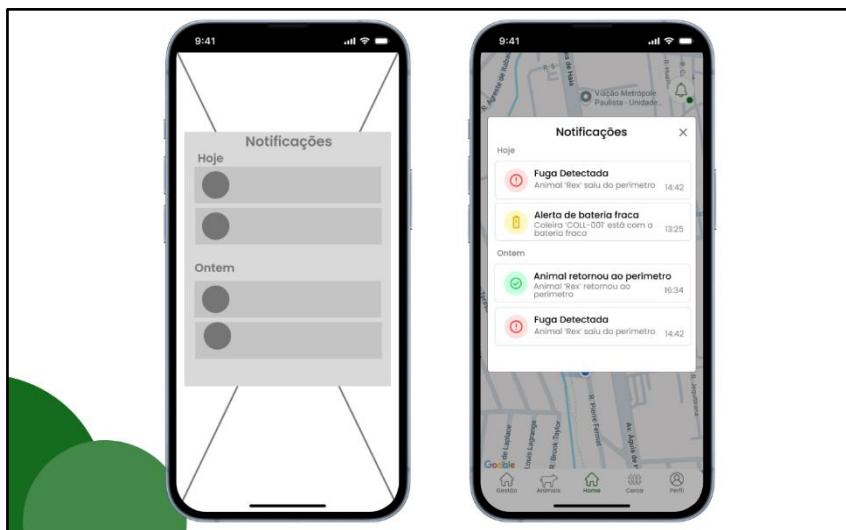
Figura 74 – Wireframes de baixa e alta fidelidade do Mapa



Fonte: De Autoria Própria, 2025

Caso o usuário clique no botão superior, uma pequena tela exibirá o histórico de todas as notificações já enviadas.

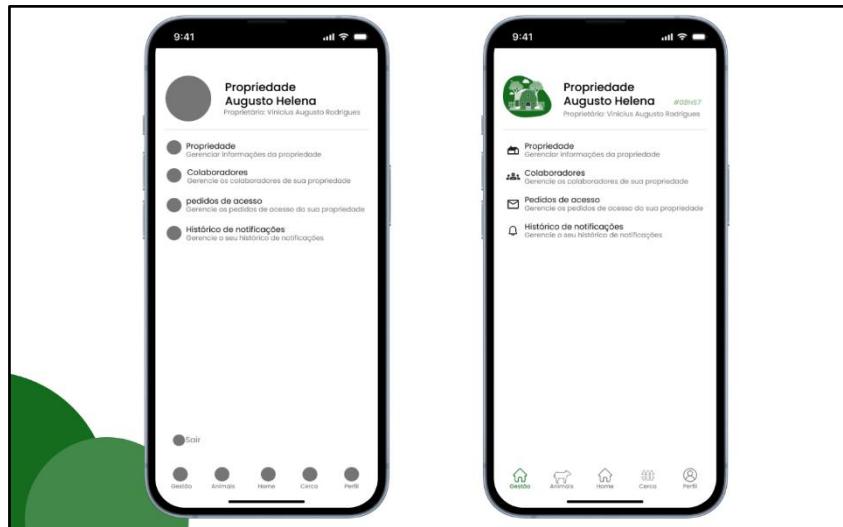
Figura 75 – Wireframes de baixa e alta fidelidade das notificações



Fonte: De Autoria Própria, 2025

Caso o usuário clique em “Propriedade”, ele será redirecionado a uma página contendo todas as informações da propriedade, contendo algumas opções.

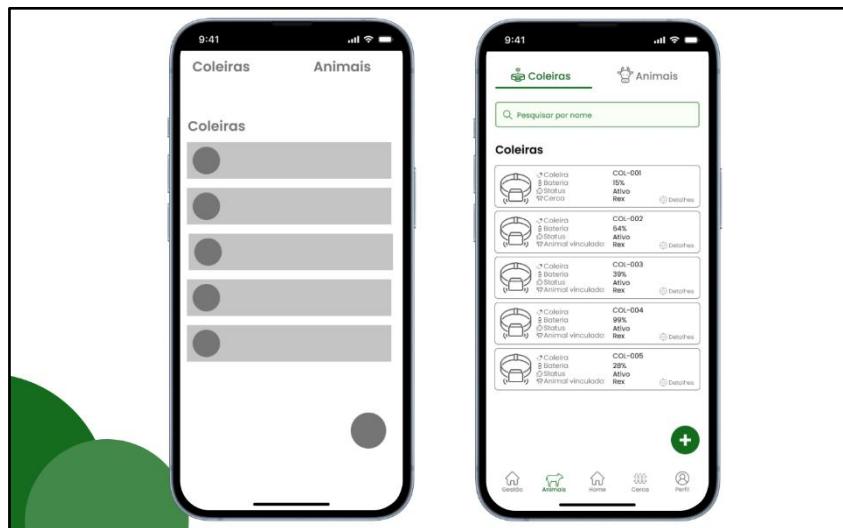
Figura 76 – Wireframes de baixa e alta fidelidade da propriedade



Fonte: De Autoria Própria, 2025

Caso o usuário escolha a página “monitoramento”, ele poderá optar entre duas páginas onde poderá adicionar uma coleira e/ou animal no sistema.

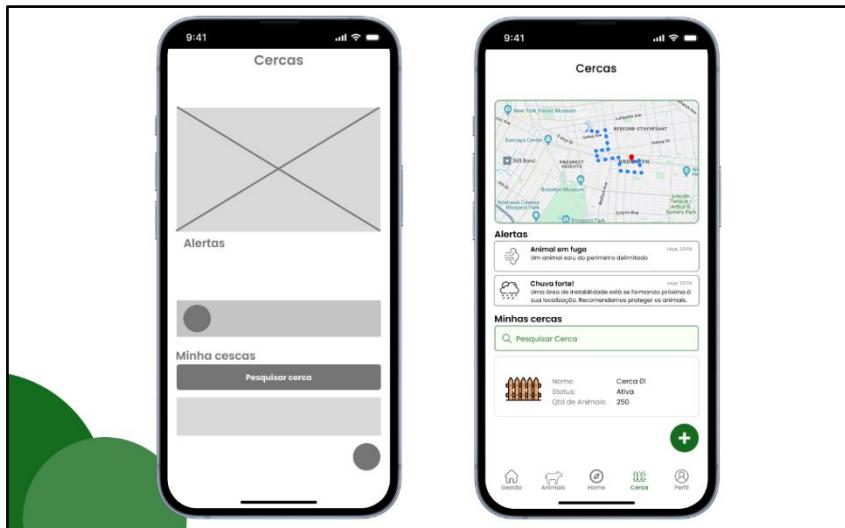
Figura 77 – Wireframes de baixa e alta fidelidade de monitoramento



Fonte: De Autoria Própria, 2025

Se o usuário optar pela página “cercas”, ele conseguirá delimitar um perímetro no mapa e, dessa forma, criar uma cerca virtual.

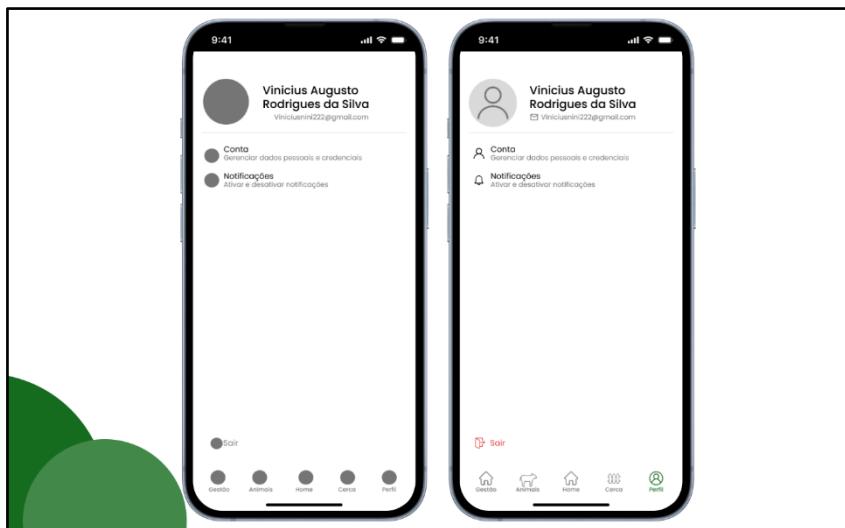
Figura 78 – Wireframes de baixa e alta fidelidade de Cercas



Fonte: De Autoria Própria, 2025

Por fim, se o usuário escolher a aba do Perfil, será exibido uma tela com informações referentes ao usuário. Nela terá a opção de gerenciar o perfil.

Figura 79 – Wireframes de baixa e alta fidelidade do Perfil



Fonte: De Autoria Própria, 2025

#### **4 CONSIDERAÇÕES FINAIS**

No decorrer do desenvolvimento deste projeto, ficou claro o impacto positivo da tecnologia no meio rural e a proposta apresentada buscou oferecer uma ferramenta eficiente e acessível para a administração do rebanho integrando tecnologia LoRa, Geofencing, energia solar e uma aplicação mobile intuitiva. Essa junção permitiu que o produtor tenha maior controle de seus animais e possa agir de forma imediata em casos de movimentação atípica, contribuindo para a segurança e gestão do gado.

Embora o sistema não tenha sido testado em grandes escalas e ainda se encontre em fase experimental, os resultados adquiridos em testes simples com menores distâncias apontam que a T-Fence pode contribuir, de forma significativa, no desenvolvimento tecnológico do setor agropecuário. Esse avanço se mostra capaz de auxiliar na redução de perdas e na proteção de propriedades rurais de larga escala.

Além disso, o projeto reforça que a inovação tecnológica é fundamental para o desenvolvimento contínuo da modernização no setor do agronegócio, comprovando que é possível solucionar problemas da área rural utilizando a tecnologia de forma estratégica. Dessa forma, a T-Fence se mostra um avanço significativo entre a tecnologia e o campo, incentivando soluções cada vez mais criativas e eficientes para ampliar a conectividade na administração do gado.

## REFERÊNCIAS

AGUIAR, Gustavo Stor. **NODE.JS: Estudo Tecnológico e Desenvolvimento Full-stack JavaScript de Plataforma De competições em Problemas Algorítmicos.** 2015. Dissertação (Graduação em Ciência da Computação) -Universidade Federal de Pernambuco, Recife, 2015.

BOAGLIO, Fernando. **MongoDB: Construa novas aplicações com novas tecnologias.** São Paulo, SP: Casa do Código, 2020. ISBN: 978-85-5519-043-8

BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivan. UML: Guia Do Usuário. 2. ed. [S. l.]. GEN LTC, 2006.

BRAGA, Renata Pereira. **Energia solar fotovoltaica: fundamentos e aplicações.** 2008. Trabalho de Conclusão de Curso (Graduação em Engenharia Elétrica) – Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2008. Disponível em: <https://pantheon.ufrj.br/handle/11422/7372>. Acesso em: 25 maio 2025.

BRASIL. MINISTÉRIO DA AGRICULTURA E PECUÁRIA (MAPA). **Crescimento da economia brasileira é impulsionado pela alta de 1,5% da agropecuária em 2023.** Brasília, DF, 2023. Disponível em: <https://www.gov.br/agricultura/pt-br/assuntos/noticias/crescimento-da-economia-brasileira-e-impulsionado-pela-alta-de-15-da-agropecuaria-em-2023>. Acesso em: 9 ago. 2025.

BROWN, Ethan. **Programação Web com Node e Express Beneficiando-se da stack JavaScript.** Rio de Janeiro: Novatec Editora, 2020.

CENTRO DE ESTUDOS AVANÇADOS EM ECONOMIA APLICADA (CEPEA). **PIB do agronegócio brasileiro.** Piracicaba, SP: CEPEA/Esalq-USP, 17 jun. 2025. Disponível em: <https://www.cepea.org.br/br/pib-do-agronegocio-brasileiro.aspx>. Acesso em: 09 ago. 2025.

CONFEDERAÇÃO DA AGRICULTURA E PECUÁRIA DO BRASIL. **Estudo sobre a criminalidade no campo.** Brasília, DF: CNA, 2017. Disponível em: <http://cnabrasil.org.br/publicacoes/estudo-sobre-criminalidade-no-campo>. Acesso em: 9 ago. 2025.

CURTO CIRCUITO. **INA219: como usar.** Disponível em: <https://curtocircuito.com.br/blog/sensores/ina219-como-usar?srsltid=AfmBOool3oGcmGtmR5k2dmJ79ppY5DVDOe6Rv8eZ9HE8Tjjgo6UYuVW2&utm>. Acesso em: 25 maio 2025.

CURTO CIRCUITO. **Módulo GPS Neo-M8N.** Disponível em: <https://curtocircuito.com.br/modulo-gps-neo-m8n.html>. Acesso em: 26 maio 2025.

DEITEL, Harvey M.; DEITEL, Paul J. **C++: Como Programar.** 5. ed. São Paulo: Pearson Prentice Hall, 2005.

EMBRAPA. **Cientistas usam drones com câmeras inclinadas para monitorar gado no pasto.** Brasília, DF: Embrapa, 2021. Disponível em:

<https://www.embrapa.br/acre/busca-de-noticias/-/noticia/58844049/cientistas-usam-drones-com-cameras-inclinadas-para-monitorar-gado-no-pasto>. Acesso em: 9 ago. 2025.

ESCUDELARIO, Bruna; PINHO, Diego. React Native: **Desenvolvimento de Aplicativos Mobile com React**. 3.ed. São Paulo: Casa do Código, 2021. 189 p.

EXPO. **Expo**. Disponível em: <https://expo.dev/>. Acesso em: 24 de maio de 2025.

FALCÃO, Filipe Dourado. **Desenvolvimento do aplicativo Turistando Beberibe utilizando React Native**. 2022. 69 f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Universidade Federal do Ceará, Fortaleza, 2022. Disponível em:  
[https://repositorio.ufc.br/bitstream/riufc/69029/3/2022\\_tcc\\_fdfalc%C3%A3o.pdf](https://repositorio.ufc.br/bitstream/riufc/69029/3/2022_tcc_fdfalc%C3%A3o.pdf). Acesso em: 25 maio 2025.

FLANAGAN, David. JavaScript – **O Guia Definitivo**. 6. ed. Porto Alegre: Bookman, 2013.

FOWLER, Martin. **Um Breve Guia para a Linguagem-padrão de Modelagem de Objetos**. 3. ed. Porto Alegre: Bookman, 2005. 160 p.

GRANT, Will. **UX design: guia definitivo com as melhores práticas de UX**. São Paulo: Novatec, 2019. 208 p.

GUEDES, Gilleanes. **UML 2: Uma Abordagem Prática**. 3. ed. São Paulo: Novatec Editora Ltda., 2018. 496 p.

GOOGLE DEVELOPERS. **Visão geral da API Maps JavaScript**. 2025. Disponível em: <https://developers.google.com/maps/documentation/javascript/overview?hl=pt-br>. Acesso em: 25 maio 2025.

GRONER, Loaine. **Estrutura de Dados e Algoritmos em JavaScript**. 2. ed. Rio de Janeiro: Novatec Editora, 2018.

IBM. **Por que MongoDB?** Disponível em: <https://www.ibm.com/br-pt/think/topics/mongodb> Acesso em: 14 de julho de 2025.

LÔBO, Rafael Miranda. **Geofencing: monitoramento de propriedades rurais**. 2022. Trabalho de Conclusão de Curso (Graduação em Engenharia da Computação) – Pontifícia Universidade Católica de Goiás, Goiânia, 2022. Disponível em:  
<https://repositorio.pucgoias.edu.br/jspui/handle/123456789/4999>. Acesso em: 25 maio 2025.

LOPES, Michele. **Modelagem 3D: o que é e como funciona**. Ebac Online, 29 de setembro de 2023. Disponível em: <https://ebaconline.com.br/blog/modelagem-3d-que-e-e-como-funciona/>. Acesso em: 14 de julho de 2025.

MAGRANI, Eduardo. **A internet das coisas.** 2. ed. Rio de Janeiro: FGV Editora, 2018.

MAKERHERO. **Pilhas e Baterias: Principais Tipos.** Disponível em: <https://www.makerhero.com/blog/pilhas-e-baterias-principais-tipos/>. Acesso em: 26 maio 2025.

MARCONI, Marina de Andrade; LAKATOS, Eva Maria. **Fundamentos de metodologia científica.** 8. ed. São Paulo: Atlas, 2017.

OLIVEIRA, Sérgio de. **Internet das coisas com ESP8266, Arduino e Raspberry Pi.** 2. ed. São Paulo: Novatec, 2021.

OLIVEIRA, Carlos Antonio Ferreira de; MEDINA, Gabriel da Silva; TEIXEIRA, Lana Mara Silva. **Análise dos registros dos crimes de furto e roubo contra propriedades rurais em Goiás nos anos de 2017 e 2018.** Boletim Goiano de Geografia, Goiânia, v. 41, n. 1, 2021. Disponível em: <https://revistas.ufg.br/bgg/article/view/62579>. Acesso em: 9 ago. 2025.

PANIZ, David. **NoSQL: Como armazenar os dados de uma aplicação moderna.** São Paulo, SP: Casa do Código, 2023. ISBN:978-85-5519-192-3

PEREIRA, Caio Ribeiro. **Node.JS Aplicações Web Real-Time com Node.JS.** Casa Do Código, 2014.

POLÍCIA CIVIL DE SANTA CATARINA. **CAOAGRO da Polícia Civil intensifica combate aos crimes contra o agronegócio.** Florianópolis, SC: Polícia Civil de Santa Catarina, 7 ago. 2024. Disponível em: <https://pc.sc.gov.br/?p=19856>. Acesso em: 9 ago. 2025.

REACT NATIVE. **React Native · Learn once, write anywhere.** Disponível em: <https://reactnative.dev/>. Acesso em: 23 de maio de 2025.

RODRIGUES, Paulo Henrique de Araujo. **Aplicação do Conceito Visual Material Design no Desenvolvimento de um Protótipo de Interface Gráfica.** 2017. 51 f. Trabalho de Conclusão de Curso (Graduação) – Tecnologia em Análise e Desenvolvimento de Sistemas. Universidade Tecnológica Federal do Paraná. Ponta Grossa, 2017.

SANTOS, Jean Willian; LARA JUNIOR, Renato Capelin de. **Sistema de automatização residencial de baixo custo controlado pelo microcontrolador ESP32 e monitorado via smartphone.** 2019. 46 f. Trabalho de Conclusão de Curso (Tecnologia em Automação Industrial) - Universidade Tecnológica Federal do Paraná, Ponta Grossa, 2019. Disponível em: <https://repositorio.utfpr.edu.br/jspui/handle/1/16960>. Acesso em: 25 maio 2025.

SARAVATI. **LoRa: sem fio de longo alcance e baixo consumo.** [S.I.], 2025. Disponível em: <https://blog.saravati.com.br/lora-sem-fio-longo-alcance-baixo-consumo/>. Acesso em: 27 maio 2025.

SILVA, Maurício Samy. Javascript, **O Guia Do Programador**. Rio de Janeiro: Novatec Editora, 2010.

TEIXEIRA, Fabrício. **Introdução e boa práticas em UX Design**. 1. ed. São Paulo: Casa do Código, 2014.

TEIXEIRA, Grazielle Bonaldi; ALMEIDA, João Víctor Peroni de. **Rede LoRa® e protocolo LoRaWAN® aplicados na agricultura de precisão no Brasil**. 2017. 76 f. Trabalho de Conclusão de Curso (Graduação em Engenharia Eletrônica) – Universidade Tecnológica Federal do Paraná, Ponta Grossa, 2017. Disponível em: <https://repositorio.utfpr.edu.br/jspui/handle/1/16191>. Acesso em: 24 maio 2025.

TEIXEIRA, Santiago Coelho. **Projeto de irrigação automática com monitoramento remoto e energy harvesting**. 2025. 69 f. Trabalho de Conclusão de Curso (Graduação em Engenharia de Controle e Automação) – Universidade Estadual Paulista “Júlio de Mesquita Filho”, Instituto de Ciência e Tecnologia, Câmpus de Sorocaba, 2025. Disponível em: <https://repositorio.unesp.br/server/api/core/bitstreams/c46e0585-1b6a-450b-b6db-8b2dd5bd7af2/content>. Acesso em: 01 de dezembro de 2025.

VILLAS-BOAS, Sergio Barbosa. **C/C++ e Orientação a Objetos em Ambiente Multiplataforma**. Rio de Janeiro, RJ, Brasil: [s.n.], 2001.