

Projeto Interdisciplinar Fatec

Grupo 4: (Inova Society)

Envolvidos: Gustavo Henrique Andrade
Davi Xavier
Igor Daniel Cintra
Audinalberth Mendes Carvalho

SUMÁRIO

1- DOCUMENTAÇÃO ENGENHARIA

- 1.1-** TAP (Termo de Aberto do Projeto)
- 1.2-** EAP (Estrutura Analítica do Projeto)
- 1.3-** Tabela SWOT
- 1.4-** 2W1H
- 1.5-** BPMN (Business Process Modeling Nation)
- 1.6-** DOC. DE REQUISITOS
- 1.8-** DIAGRAMA DE CASO DE USO

2- DICIONÁRIO DE DADOS

- 2.1** PRD
- 2.2** MER
- 2.3** DER

3- TESTES E INTEGRAÇÃO

- 3.1** IMPLEMENTAÇÃO E TESTES DA SOLUÇÃO

1.1- TAP

Criação de Sistema para “Associação Só Patinhas de Rua”

1. Situação Atual

Atualmente a Organização não há nenhum tipo de padrão na aquisição de novos voluntários, a partir disso é possível notar após uma entrevista com a CEO da ong é notável que isso pode gerar uma dificuldade no funcionamento da organização, pois ela precisa de novos voluntários para a limpeza dos canis, voluntários que possam doar desde a sua força de trabalho até certa quantia monetária.

Outro ponto negativo é a falta de um local para doações, verificamos que no único canal de comunicação da ong não fica especificado um site para doar, conhecer a história da ong, até mesmo para a inscrição de novos voluntários. Isso pode acarretar numa possível perda dos quais, como podem desistir do processo a partir do momento que não haja um local específico para doações e inscrição de voluntariados.

Um ponto positivo é o Networking e o Trabalho da criadora Lindsey, ela é bastante conhecida na cidade sendo frequentemente chamada em diversas localidades dá para fazer o resgate de novos animais com o apoio de pessoas próximas a ONG, um exemplo é a veterinária “Sabrina” que faz descontos em procedimentos veterinários nos animais. Seu trabalho favorece sua organização, pois com o seu trabalho de vereadora consegue apoio

em certos momentos da prefeitura, como o apoio do VETMOVEL que é uma “Ambulância” apenas para animais que faz resgates pela cidade todos os dias das 8:00 até as 16:00.

2. Justificativa do Projeto

Os cenários visto pela nossa equipe foi de um projeto abandonado em questões precárias de tecnologia. A ideia inicial seria desenvolver um sistema que poderia auxiliar a organização dos canis, fazer com o que tenha novos voluntários e doadores para ONG.

3. Propósito do Projeto e Metas

O projeto tem como objetivo desenvolver um sistema que possa auxiliar na arrecadação de fundos com o objetivo de deixar a ONG mais sustentável através de um maior número de doações vindas dos usuários.

Outro ponto a ser melhorado é a facilidade de se filiar a ONG contendo dentro do próprio site um local onde os usuários poderão entrar e se cadastrar para fazer parte da ONG, onde deverá ter um formulário com suas habilidades e como ela pode ajudar de alguma forma na organização.

Deverá conter um local onde haverá um botão “Quero Doar”, onde o usuário conseguirá fazer uma doação diretamente para a ONG.

Na página também terá um local destinado apenas para doações, onde teria fotos de cachorros e um local onde ele poderá agendar uma visita para escolher qual animal gostaria de adotar.

Também está dentro das metas a construção de uma planilha para a anotação de gastos com a ONG, uma espécie de transparência com quem doa e com os voluntariados.

Por fim a página deverá ter um local que conte a história da ONG onde o usuário poderá entrar e entender os princípios, os desafios e como criada a organização.

4. Descrição do Projeto

O Projeto foi construído a partir da observação das fraquezas da ong e tem como objetivo apoiar a causa de animais de rua, por meio do desenvolvimento de uma solução tecnologia que otimize as operações e consigo conseguir ampliar o alcance que a organização queira chegar.

A partir da visitação na associação foi possível observar as fraquezas e as forças da empresa, partindo desse ponto fizemos as tabelas SWOT e 5W2H, com o objetivo de analisar o que seria de interessante para inserir no sistema que está sendo desenvolvido.

Serão desenvolvidas ações para adoções de cães e gatos dentro página, um sistema para cadastro de voluntários, campanhas de arrecadação de recursos e u sistema de controle de animais que faria a função de contabilizar quantos animais foram adotados em determinado mês, quantos animais tem na ONG e a quantidade de gastos e receitas do movimento.

5. Premissas

A ONG fornecerá fotos dos Animais e o conteúdo necessário dentro da página.

Os desenvolvedores terão a autonomia para escolher as ferramentas adequadas de acordo com o projeto

O objetivo da página (informar, captar doações, divulgar adoções, etc.) será previamente definido junto à ONG.

O objetivo principal da landing page (divulgação institucional, incentivo à adoção, arrecadação de doações, captação de voluntários etc.) será alinhado previamente com a ONG, orientando toda a estrutura e o conteúdo da página.

Desenvolver o projeto visando a fácil implantação e manutenção, tendo em vista que não só a CEO o utilizará.

6. Restrições

Prazo de entrega próximo.

Falta de tempo para todos visitarem as instalações do Projeto.

Tudo é feito a partir de celulares ou computadores dos voluntários, a Associação não possui equipamentos.

Conhecimento dos desenvolvedores sobre Javascript.

Não há verba o suficiente no Projeto social.

7. Stakeholders

Voluntários, pois eles podem contar experiencias vividas dentro do Projeto Social que podem ajudar no sistema.

Público-alvo, Doadores, adotantes em potencial e simpatizantes da causa.

Instituições apoiadoras/parceiras, Clínicas veterinárias, pet shops, empresas locais, porque podem ser citadas ou ter interesse na visibilidade do projeto.

8. Riscos

Segurança dos dados

Alto fluxo de informações, pois a iniciativa possui altos número de cães e Gatos

Questão financeira da ONG

9. Marco

O cronograma foi estipulado para as entregas iniciais, desde sua concepção, até a data de entrega do documento, a entrega estipulada das responsabilidades individuais foi para o dia 12/04/2025 e a data de entrega das documentações foi para 25/04/2025

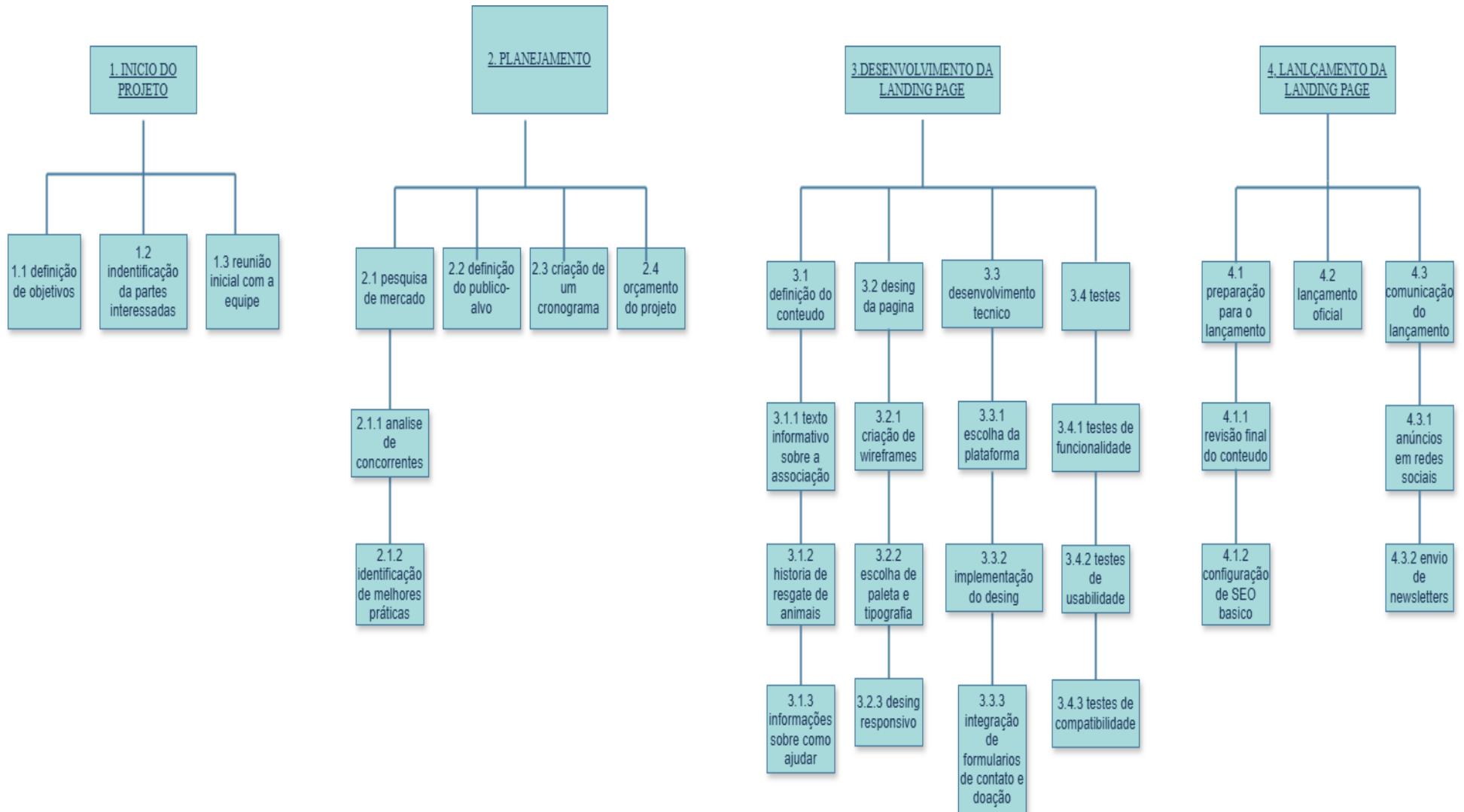
10. Responsabilidades

Gerente: Gustavo Henrique

Diagrama de EAP: Igor Cintra

Documentação de Requisitos: Pedro Henrique

1.2- EAP



1.3- Tabela SWOT

| PONTOS FORTES + | PONTOS FRACOS - |
|---|---|
| <ul style="list-style-type: none">• Trabalho pessoal da diretora, auxilia na ONG• Parcerias: Nutrinorte, Adota Franca e Veterinário Sabrina• Devoção da diretora• Castração fornecida pela ONG• “Lei cão comunitário” | <ul style="list-style-type: none">• Falta de voluntários• Falta de adoção/doação• Falta de um sistema funcional• Sem registro e contagem dos animais |

| OPORTUNIDADES + | AMEAÇAS - |
|--|---|
| <ul style="list-style-type: none">• Campanhas, rifas• Sorteio (para arrecadação)• Parcerias com Pet Shops e Veterinários e Faculdades• Divulgação | <ul style="list-style-type: none">• Despejo• Falta de orçamento• Dificuldade para conseguir voluntários• Imprevistos de trabalho com animais que geram gastos excessivos e necessários |

1.4- 2W1H

Falta de voluntários:

| O que? | Por quê? | Como? |
|--|---|---|
| Gerar pontuação para os voluntários. | A fim de oferecer benefícios para o voluntário. | Desenvolver um sistema de pontuação e prêmio para voluntários. |
| Divulgar na LP a necessidade de voluntários e realizar o contato entre ONG e voluntario através da página. | A fim de obter identificação com a ONG e recrutar colaboradores | Inserindo na LP informações para contato e a história da associação |

Falta de doação:

| O que? | Por quê? | Como? |
|---|--|---|
| Inserir barra de metas de doação na LP. | A fim de chamar atenção de usuários para a doação. | Inserindo uma barra de doação que se completa de acordo com as metas. |
| Oferecer divulgação para parcerias dentro da LP em troca de alimento. | A fim de fortalecer parcerias e conseguir alimentos. | Exibindo parcerias da ONG na LP afim de divulgar. |

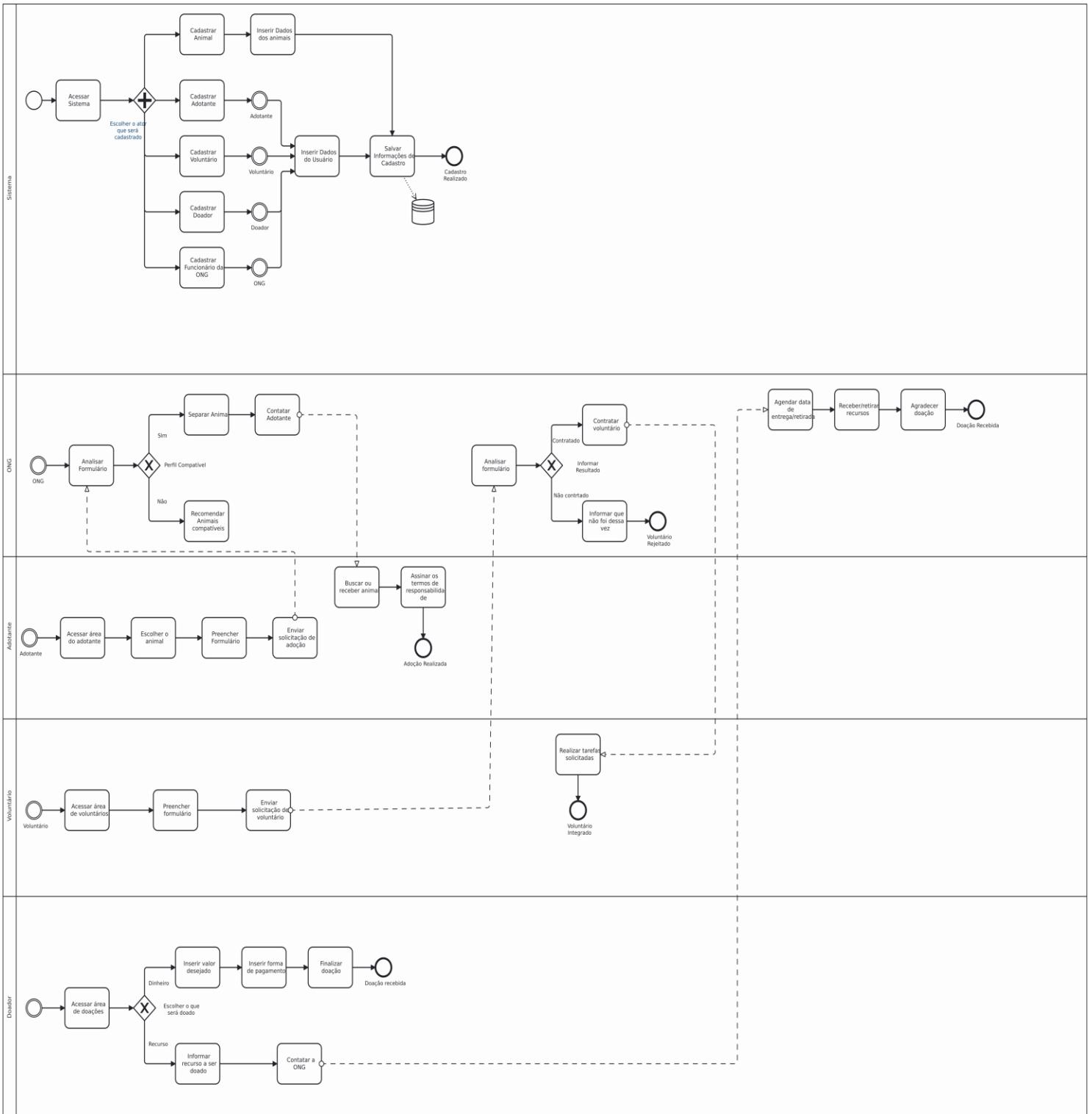
Registro de cães:

| O que? | Por quê? | Como? |
|--|---|---|
| Desenvolver um sistema para registrar manualmente. | A fim de controlar e administrar a ONG. | Desenvolvendo um sistema. |
| Separar por categorias, como cães e gatos. | A fim de facilitar identificação e manuseio de animais. | Registrando por categorias e subcategorias. |

Falta de controle de gastos:

| O que? | Por quê? | Como? |
|---|---|------------------------|
| Desenvolver um sistema de controle de gastos. | A fim de obter controle de gastos e auxiliar na administração. | Desenvolvendo sistema. |
| Dar início a uma planilha no Excel. | Para contabilizar a entrada e saída de gastos. | Criando planilhas. |

1.5-BPMN



1.6-DOCUMENTAÇÃO DE REQUISITOS

| | |
|--------------------------|---|
| ID | RF01 |
| Nome do requisito | Analisar Formulário |
| Descrição | Permitir que os voluntários analisem os formulários preenchidos pelos adotantes interessados. |
| Categoria | Evidente |
| Prioridade | Essencial |
| Informações | O formulário deve exibir os dados preenchidos e opções para aprovar ou recusar. |
| Regras do negócio | Apenas usuários com permissão podem analisar. Após análise, o status do formulário é travado. |

| | |
|--------------------------|--|
| ID | RF02 |
| Nome do requisito | Separar Animal |
| Descrição | Permitir reservar um animal após aprovação do adotante, retirando-o da listagem pública. |
| Categoria | Oculto |
| Prioridade | Essencial |
| Informações | A reserva é automática após aprovação. |
| Regras do Negócio | Um animal só pode ser reservado para um adotante por vez. |

| | |
|--------------------------|---|
| ID | RF03 |
| Nome do Requisito | Recomendar Animais Compatíveis |
| Descrição | Sugerir animais com base no perfil do adotante. |
| Categoria | Evidente |
| Prioridade | Desejável |
| Informações | Recomendações baseadas em dados como porte, ambiente e estilo de vida. |
| Regras do Negócio | O sistema pode utilizar um algoritmo de compatibilidade. Adotante pode ignorar sugestões. |

| | |
|--------------------------|---|
| ID | RF04 |
| Nome do Requisito | Contatar Adotante |
| Descrição | Permitir envio de mensagens ao adotante com o status da adoção. |
| Categoria | Evidente |
| Prioridade | Essencial |
| Informações | Pode ser automático ou manual, via e-mail ou WhatsApp. |
| Regras do Negócio | Apenas usuários autorizados podem enviar mensagens. |

| | |
|---------------------------|---|
| ID: | RF05 |
| Nome do Requisito: | Analisar Formulário |
| Descrição: | Analisar o formulário enviado pelo civil para se tornar voluntário da ONG |
| Categoria: | Evidente |
| Prioridade: | Essencial |

| | |
|---------------------------|---|
| Informações | Formulário com informações sobre o indivíduo que quer se juntar a ONG como voluntário |
| Regras do negócio: | Não possui |

| | |
|--------------------------|--|
| ID | RF06 |
| Nome do Requisito | Contratar Voluntário |
| Descrição | Notificar voluntários sobre tarefas, eventos e mudanças. |
| Categoria | Oculto |
| Prioridade | Importante |
| Informações | Envio via e-mail ou WhatsApp conforme cadastro. |
| Regras do Negócio | Cada notificação deve respeitar o perfil e as funções do voluntário. |

| | |
|--------------------------|--|
| ID | RF07 |
| Nome do Requisito | Informar que Não Foi Dessa Vez |
| Descrição | Enviar mensagens educadas ao adotante não aprovado. |
| Categoria | Evidente |
| Prioridade | Essencial |
| Informações | Mensagem automática, mas com tom humanizado. |
| Regras do Negócio | Só pode ser enviada após análise negativa do formulário. |

| | |
|--------------------------|--|
| ID | RF08 |
| Nome do Requisito | Agendar Data de Entrega/Retirada |
| Descrição | Marcar datas para entrega de animais ou retirada de doações. |
| Categoria | Evidente |
| Prioridade | Essencial |
| Informações | Seleção de data, hora e responsável no sistema. |
| Regras do Negócio | Confirmações devem ser enviadas aos envolvidos. |

| | |
|--------------------------|---|
| ID | RF09 |
| Nome do Requisito | Receber/Retirar Recursos |
| Descrição | Registrar entrada ou saída de doações físicas ou financeiras. |
| Categoria | Oculto |
| Prioridade | Essencial |
| Informações | Detalhamento do recurso, data e responsável pelo registro. |
| Regra do Negócio | Todos os registros devem ter um usuário autenticado como responsável. |

| | |
|--------------------------|--|
| ID | RF010 |
| Nome do Requisito | Agradecer Doação |
| Descrição | Enviar automaticamente uma mensagem de agradecimento após doação registrada. |
| Categoria | Oculto |
| Prioridade | Importante |

| | |
|-------------------------|---|
| Informações | Mensagem com nome do doador e dados da doação (se permitido). |
| Regra do Negócio | O envio só ocorre com a confirmação da doação no sistema. |

| | |
|---------------------------|--|
| ID: | RF011 |
| Nome do Requisito: | Acessar Sistema |
| Descrição: | Acessar o sistema para cadastrar e logar o usuário |
| Categoria: | Evidente |
| Prioridade: | Essencial |
| Informações: | Início dos processos de cadastro e login |
| Regra do Negócio: | Não possui |

| | |
|---------------------------|--|
| ID | RF012 |
| Nome do Requisito: | Cadastrar Voluntário |
| Descrição: | Cadastrar o login/cadastro do voluntário |
| Categoria: | Oculto |
| Prioridade: | Essencial |
| Informações: | Dados de cadastro, usuário e senha |
| Regra do Negócio: | Não Possui |

| | |
|---------------------------|-----------------------------|
| ID: | RF013 |
| Nome de Requisito: | Cadastrar Animal |
| Descrição: | Cadastrar animal no sistema |
| Categoria: | Evidente |
| Prioridade: | Essencial |
| Informações: | Situação do animal. |

| | |
|---------------------------|--|
| Regras do negócio: | Cadastra animal para ter um registro sobre cirurgias e situação atual. |
|---------------------------|--|

| | |
|---------------------------|--|
| ID: | RF014 |
| Nome de Requisito: | Cadastrar Adotante |
| Descrição: | Realizar o Cadastro do adotante |
| Categoria: | Oculto |
| Prioridade: | Essencial |
| Informações: | Informações do adotante, como endereço, número e data de adoção. |
| Regras do negócio: | Ter informações de contato para entrar em contato com adotante para saber a situação em que ele se encontra. |

| | |
|---------------------------|---|
| ID: | RF015 |
| Nome de Requisito: | Cadastrar Doador. |
| Descrição: | Realizar cadastro do doador |
| Categoria: | Evidente |
| Prioridade: | Essencial |
| Informações: | Informações do doador, como quanto doou e data de adoção. |
| Regras do negócio: | Ter as informações sobre o doador como quanto doou. |

| | |
|---------------------------|------------------------------|
| ID: | RF016 |
| Nome de Requisito: | Cadastrar funcionário da ONG |

| | |
|---------------------------|---|
| Descrição: | Realizar cadastro do diretor. |
| Categoria: | Evidente |
| Prioridade: | Essencial |
| Informações: | Informações sobre a ONG, como doações, cadastro de animais e ter acesso ao sistema inteiro. |
| Regras do negócio: | O diretor da ONG deverá ter o cadastro para acessar o sistema. |

| | |
|---------------------------|--|
| ID: | RF017 |
| Nome de Requisito: | Inserir dados dos animais. |
| Descrição: | Inserir dados do Animal |
| Categoria: | Essencial |
| Prioridade: | Informações do Animal, contendo como ele se encontra no momento, onde ele se encontra. |
| Informações: | No sistema deverá haver todos os Dados do Animal. |
| Regras do negócio: | Evidente |

| | |
|---------------------------|---|
| ID: | RF018 |
| Nome de Requisito: | Inserir os dados do usuário. |
| Descrição: | Inserir os dados do usuário. |
| Categoria: | Evidente. |
| Prioridade: | Essencial. |
| Informações: | Inserir as informações, contendo suas informações |
| Regras do negócio: | Conter todos os dados do usuário. |

| | |
|------------|--------------|
| ID: | RF019 |
|------------|--------------|

| | |
|---------------------------|--|
| Nome de Requisito: | Salvar informações de Cadastro |
| Descrição: | Salvar informações do usuário |
| Categoria: | Evidente |
| Prioridade: | Essencial |
| Informações: | Conter todas as informações de cadastro |
| Regras do negócio: | Conter a informações de todos que acessam ao sistema |

| | |
|---------------------------|---|
| ID: | RF020 |
| Nome de Requisito: | Acessar área do adotante |
| Descrição: | Acessar área de adoção do site ou app com o intuito de adotar um animal |
| Categoria: | Evidente |
| Prioridade: | Essencial |
| Informações: | Detalhes do cachorro desejado |
| Regras do negócio: | Ter responsabilidade e lugar bem arrojado para poder adotar o animal |

| | |
|---------------------------|------------------------------------|
| ID: | RF021 |
| Nome de Requisito: | Escolher o Animal |
| Descrição: | Escolher qual animal deseja adotar |
| Categoria: | Evidente |
| Prioridade: | Essencial |
| Informações: | Foto do animal |
| Regras do negócio: | Não possui |

| | |
|------------|--------------|
| ID: | RF022 |
|------------|--------------|

| | |
|---------------------------|--|
| Nome de Requisito: | Preencher Formulário |
| Descrição: | Preencher formulário com informações pessoais e locais |
| Categoria: | Essencial |
| Prioridade: | Essencial |
| Informações: | CPF, RG e Endereço |
| Regras do negócio: | Possuir documentos |

| | |
|--------------------|--|
| ID: | RF023 |
| Nome de Requisito: | Enviar solicitação de adoção |
| Descrição: | Enviar formulário e solicitação de adoção para ONG |
| Categoria: | Evidente |
| Prioridade: | Essencial |
| Informações: | CPF, RG, endereço e contato com a ONG |
| Regras do negócio: | Não possui |

| | |
|---------------------------|--|
| ID: | RF024 |
| Nome de Requisito: | Buscar ou receber animal |
| Descrição: | Ir até a ONG buscar o animal ou receber o animal em casa |
| Categoria: | Oculto |
| Prioridade: | Essencial |
| Informações: | Endereço da ONG ou da casa onde será recebido o animal |
| Regras do negócio: | Não Possui |

| | |
|---------------------------|---|
| ID: | RF025 |
| Nome do Requisito: | Assinar os termos de responsabilidade |
| Descrição: | Assinar os documentos de responsabilidade fornecidos pela ONG |
| Prioridade: | Essencial |
| Informações: | Documentação de Responsabilidade |
| Regra do Negócio: | Não Possui |
| Categoria: | <u>Evidente</u> |

| | |
|---------------------------|---|
| ID: | RF026 |
| Nome do Requisito: | Acessar área de voluntários |
| Descrição: | O ator voluntário deverá acessar a área de voluntários afim de realizar o formulário e conseguir se inscrever |
| Prioridade: | Essencial |
| Informações: | CPF, RG, Endereço e Foto 3x4 |
| Regra do Negócio: | Não Possui |
| Categoria: | <u>Evidente</u> |

| | |
|---------------------------|--|
| ID: | RF027 |
| Nome do Requisito: | Preencher o formulário |
| Descrição: | Preencher o formulário com informações do voluntário para se inscrever para a vaga |

| | |
|--------------------------|------------------------------|
| Prioridade: | Essencial |
| Informações: | CPF, RG, Endereço e Foto 3x4 |
| Regra do Negócio: | Não Possui |
| Categoria: | Evidente |

| | |
|---------------------------|--|
| ID: | RF028 |
| Nome do Requisito: | Enviar Solicitação de voluntário |
| Descrição: | Enviar o formulário solicitando fazer parte da ONG |
| Prioridade: | Essencial |
| Informações: | Essencial |
| Regra do Negócio: | Não Possui |
| Categoria: | Evidente |

| | |
|---------------------------|---|
| ID: | RF029 |
| Nome do Requisito: | Realizar Tarefas Solicitadas |
| Descrição: | Realizar as tarefas que são solicitadas e comandadas pela ONG |
| Prioridade: | Essencial |
| Informações: | Cumprir tarefas delegadas |
| Regra do Negócio: | Não Possui |
| Categoria: | Evidente |

| | |
|---------------------------|-------------------------|
| ID: | RF030 |
| Nome do Requisito: | Acessar área de doações |

| | |
|--------------------------|---|
| Descrição: | Acessar a área de doação dentro do site ou app para realizar a doação |
| Prioridade: | Essencial |
| Informações: | Acessar site para realizar doação |
| Regra do Negócio: | Não Possui |
| Categoria: | Evidente |

| | |
|---------------------------|--|
| ID: | RF031 |
| Nome do Requisito: | Inserir Valor Desejado |
| Descrição: | Inserir o valor (quantia) que será doado |
| Prioridade: | Essencial |
| Informações: | Valor a ser doado |
| Regra do Negócio: | Não Possui |
| Categoria: | Evidente |

| | |
|---------------------------|---|
| ID: | RF032 |
| Nome do Requisito: | Informar recurso a ser doado |
| Descrição: | Dar informação de qual recurso será doado para a ONG (saco de ração, recurso de higiene, remédios etc.) |
| Prioridade: | Essencial |
| Informações: | Recursos que serão doados |
| Regra do Negócio: | Não possui |
| Categoria: | Evidente |

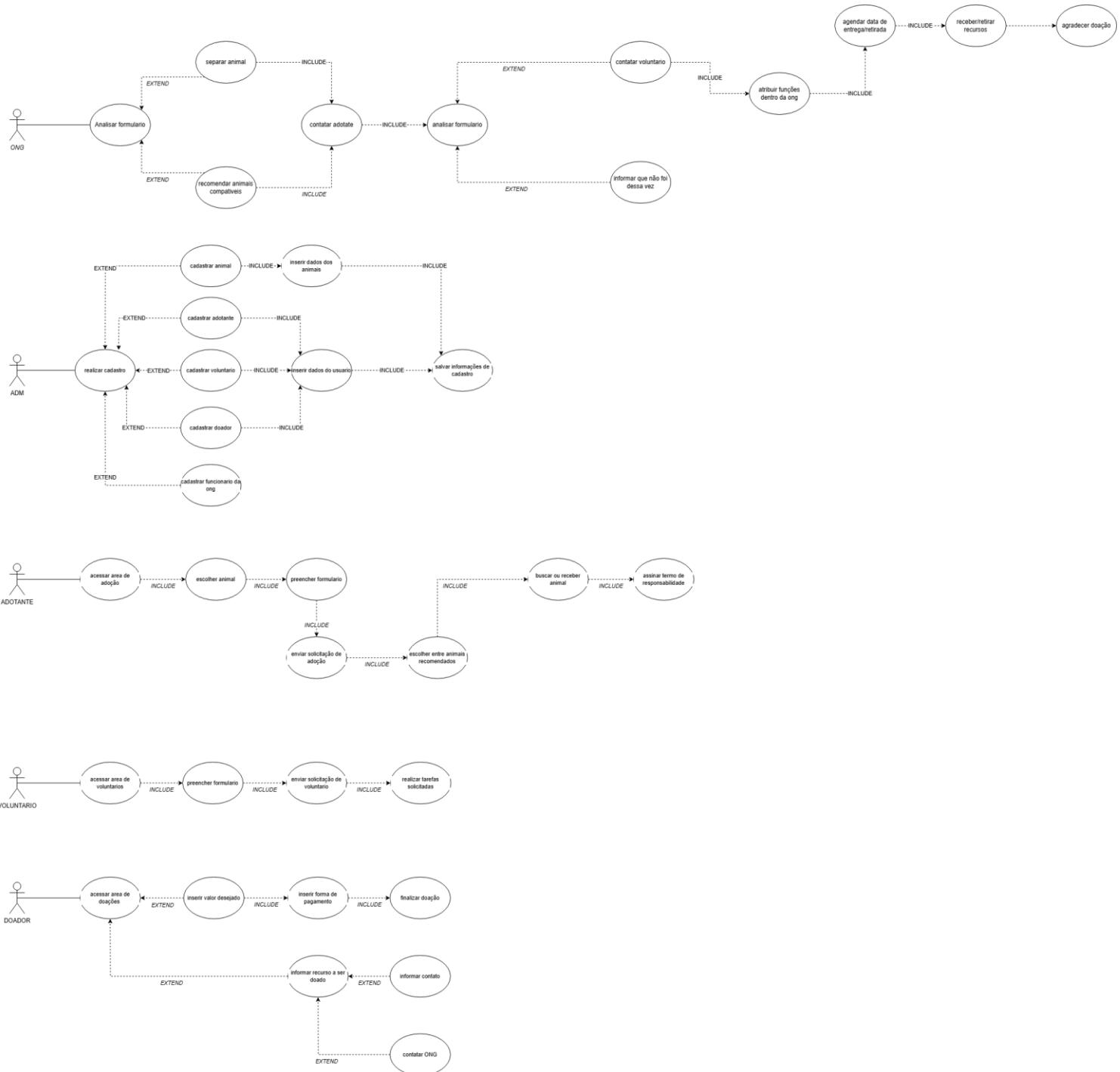
| | |
|---------------------------|--|
| ID: | RF033 |
| Nome do Requisito: | Contatar ONG |
| Descrição: | Mandar mensagem para ONG informando que quer doar recursos |
| Prioridade: | Essencial |
| Informações: | Utilizar contato da ONG para enviar mensagem |
| Regra do Negócio: | Não Possui |
| Categoria: | Evidente |

| | |
|---------------------------|---|
| ID: | RF034 |
| Nome do Requisito: | Inserir forma de pagamento |
| Descrição: | Informar qual será a forma de pagamento a ser utilizada (pix, cartão, dinheiro) |
| Prioridade: | Essencial |
| Informações: | Forma de pagamento utilizada pelo doador |
| Regra do Negócio: | Não Possui |
| Categoria: | Evidente |

| | |
|---------------------------|--|
| ID: | RF035 |
| Nome do Requisito: | Finalizar doação |
| Descrição: | Realizar o pagamento do valor a ser doado para a ONG |

| | |
|--------------------------|-------------------------------|
| Prioridade: | Essencial |
| Informações: | Processo de doação finalizado |
| Regra do Negócio: | Não Possui |
| Categoria: | Evidente |

1.7-DIAGRAMA DE CASO DE USO



1.8-DOCUMENTAÇÃO DE CASO DE USO

Analisar formulário

| | |
|---------------------------|---|
| ID | UC001 |
| Descrição | Permitir que um voluntário autorizado analise um formulário de adoção enviado por um adotante, atendendo ao RF01. |
| Autor Primário | ONG |
| Pré-condição | Estar logado como ONG com permissão para análise |
| Cenário Principal: | <ol style="list-style-type: none">1. O voluntário acessa a área de formulários pendentes.2. O sistema exibe a lista de formulários aguardando análise.3. O voluntário seleciona um formulário.4. O sistema exibe os dados preenchidos pelo adotante.5. O voluntário escolhe aprovar ou rejeitar.6. O sistema registra a decisão e atualiza o status.7. O sistema notifica o adotante. |
| Pós-condição | Formulário fica com status travado. |

| | |
|----------------------------|--|
| Cenário Alternativo | *a – O voluntário sai sem concluir a análise. 5a – Nenhuma opção selecionada gera alerta. |
| Inclusão | UC04 - Contatar Adotante |
| Extensão | UC07 - Informar que Não Foi Dessa Vez |

Separar Animal

| | |
|----------------------------|--|
| ID | UC002 |
| Descrição | Reservar automaticamente um animal para o adotante aprovado, atendendo ao RF02. |
| Autor Primário | Sistema |
| Pré-condição | Formulário aprovado pelo voluntário |
| Cenário Principal | <p>1. O sistema identifica a aprovação do formulário.</p> <p>2. O animal indicado é reservado automaticamente.</p> <p>3. O sistema retira o animal da lista pública de adoção.</p> |
| Pós-condição | Animal marcado como “reservado” |
| Cenário Alternativo | 2a – Caso o animal já esteja reservado, sistema exibe erro |

| | |
|-----------------|------------|
| Inclusão | Não possui |
| Extensão | Não possui |

Recomendar Animais Compatíveis

| | |
|----------------------------|---|
| ID | UC003 |
| Descrição | <i>Sugerir automaticamente animais compatíveis com o perfil do adotante, atendendo ao RF03.</i> |
| Autor Primário | Sistema |
| Pré-condição | Formulário enviado pelo adotante |
| Cenário Principal | <ol style="list-style-type: none"> 1. O sistema analisa os dados preenchidos pelo adotante. 2. Executa o algoritmo de compatibilidade. 3. Exibe sugestões de animais na tela do adotante |
| Pós-condição | Sugestões visíveis ao adotante. |
| Cenário Alternativo | 2a – Se nenhum animal compatível for encontrado, exibe mensagem padrão. |
| Exclusão | Não possui |
| Inclusão | Não possui |

Contatar Adotante

| | |
|----------------------------|--|
| ID | UC004 |
| Descrição | Permitir que voluntários enviem atualizações ao adotante, atendendo ao RF04. |
| Autor Primário | ONG |
| Pré-condição | Estar autenticado e com permissão de envio |
| Cenário Principal | <ol style="list-style-type: none">1. O voluntário acessa o perfil do adotante.2. Escolhe a opção de contato.3. Escreve ou seleciona uma mensagem.4. O sistema envia via e-mail ou WhatsApp. |
| Pós-condição | Mensagem registrada no histórico do adotante. |
| Cenário Alternativo | 3a – Se a mensagem estiver em branco, sistema emite alerta. |
| Inclusão | Não possui |
| Extensão | Não possui |

Analisar formulário

| | |
|---------------------------|---|
| ID | UC005 |
| Descrição | Permitir que um funcionário autorizado analise um formulário de adoção enviado por um voluntario |
| Autor Primário | ONG |
| Pré-condição | Estar logado como ONG com permissão para análise |
| Cenário Principal: | <p>1. O funcionário acessa a área de formulários pendentes.</p> <p>2. O sistema exibe a lista de formulários aguardando análise.</p> <p>3. O funcionário seleciona um formulário.</p> <p>4. O sistema exibe os dados preenchidos pelo voluntariado.</p> <p>5. O funcionário escolhe aprovar ou rejeitar.</p> <p>6. O sistema registra a decisão e atualiza o status.</p> <p>7. O sistema notifica o voluntariado.</p> |
| Pós-condição | Formulário fica com status travado. |

| | |
|----------------------------|--|
| Cenário Alternativo | *a – O voluntário sai sem concluir a análise. 5a – Nenhuma opção selecionada gera alerta. |
| Inclusão | UC06 – Contratar Voluntário |
| Extensão | UC07 - Informar que Não Foi Dessa Vez |

Contratar Voluntário

| | |
|----------------------------|--|
| ID | UC006 |
| Descrição | Notificar voluntários sobre suas tarefas ou decisões internas, atendendo ao RF05. |
| Autor Primário | Sistema |
| Pré-Condição | Voluntário cadastrado com canal de contato válido |
| Cenário Principal | <p>1. O sistema detecta uma tarefa atribuída.</p> <p>2. Verifica dados de contato do voluntário.</p> <p>3. Envia mensagens com instruções.</p> |
| Pós-Condição | Notificação registrada. |
| Cenário Alternativo | 2a – Se não houver contato válido, sistema registra falha. |
| Inclusão | UC029 - Atribuir Funções |
| Extensão | Não possui |

Informar que Não Foi Dessa Vez

| | |
|----------------------------|--|
| ID | UC007 |
| Descrição | <i>Enviar uma mensagem automática e gentil ao adotante não aprovado, atendendo ao RF06.</i> |
| Autor Primário | ONG |
| Pré-Condição | <i>Formulário reprovado</i> |
| Cenário Principal | <ol style="list-style-type: none">1. O sistema detecta reprovação no formulário.2. Seleciona mensagem personalizada.3. Envia mensagem ao adotante. |
| Pós-Condição | <i>Mensagem registrada.</i> |
| Cenário Alternativo | <i>3a – Se o envio falhar, sistema tenta novamente em 1 hora.</i> |
| Inclusão | UC01 - Analisar Formulário |
| Exclusão | <i>Não possui</i> |

Agendar data de entrega/retirada

| | |
|-----------------------------|---|
| ID: | UC 008 |
| Descrição: | Agendar uma data e hora para entrega ou retirada do animal adotado, atendendo ao RF09. |
| Autor Primário: | Voluntário ou Administrado |
| Pré-Condição: | Adoção aprovada e animal reservado |
| Cenário Principal: | 1. O ator acessa o perfil do adotante ou animal. 2. Clica na opção “Agendar entrega/retirada”. 3. Escolhe a data e o horário disponíveis. 4. O sistema registra o agendamento. 5. O adotante recebe a confirmação do agendamento. Pós-condição: Data registrada e visível para as partes envolvidas |
| Cenário Alternativo: | Animal retirado do sistema após adoção. |
| Inclusão: | Não possui |
| Extensão: | Não possui |
| Cenário Alternativo: | 3a – Se o horário já estiver ocupado, o sistema bloqueia a seleção. |

Receber/Retirar Recursos

| | |
|-------------------|---|
| ID: | UC 009 |
| Descrição: | Registrar a retirada de doações físicas (como |

| | |
|-----------------------------|---|
| | ração, produtos de limpeza, medicamentos), atendendo ao RF10. |
| Autor Primário: | Voluntário ou Administrador |
| Pré-Condição: | Registro da doação no sistema |
| Cenário Principal: | 1. O voluntário acessa a área de doações registradas. 2. Seleciona o item a ser retirado. 3. Informa a data e o destino da retirada. 4. O sistema grava o registro da movimentação. |
| Pós-Condição: | Pós-condição: Registro de retirada disponível no sistema |
| Inclusão: | Não Possui |
| Extensão: | Não Possui |
| Cenário Alternativo: | 2a – Se não houver saldo no item, sistema impede a ação. |

Agradecer Doação

| | |
|---------------------------|---|
| ID: | UC 0010 |
| Descrição: | Enviar uma mensagem automática de agradecimento ao doador, atendendo ao RF11. |
| Autor Primário: | Sistema |
| Pré-Condição: | Doação registrada no sistema |
| Cenário Principal: | 1. O sistema detecta nova doação registrada. 2. Recupera dados de contato do doador. 3. Envia |

| | |
|-----------------------------|---|
| | mensagem personalizada de agradecimento. 4. Registra o envio da mensagem no histórico do doador |
| Pós-Condição: | Doador é notificado e agradecimento é registrado. |
| Inclusão: | Não possui |
| Extensão: | Não Possui |
| Cenário Alternativo: | 3a – Se o doador não tiver contato registrado, o sistema gera alerta para preenchimento manual. |

Acessar sistema

| | |
|---------------------------|--|
| ID: | UC 0011 |
| Descrição: | Este caso de uso tem por objetivo iniciar o processo de cadastro no sistema. |
| Ator Primário: | Administrador |
| Pré-Condição: | Estar autenticado no sistema com permissão de administrador |
| Cenário Principal: | 1. O caso de uso inicia quando o administrador acessa o sistema e seleciona a opção "Realizar Cadastro". 2. O sistema apresenta opções de cadastro: animal, adotante, voluntário, |

| | |
|-----------------------------|--|
| | doador, diretor da ONG. 3. O administrador seleciona a opção desejada. 4. O sistema direciona para o formulário de cadastro correspondente |
| Pós-Condição: | Fluxo de cadastro iniciado corretamente. |
| Inclusão: | UC 0013, UC 0014, UC 0015, UC 0016 |
| Extensão: | UC 0017, UC 0018, UC 0019 |
| Cenário Alternativo: | a – O administrador pode sair do sistema a qualquer momento. |

Cadastrar Animal

| | |
|---------------------------|---|
| ID: | UC 012 |
| Descrição: | Este caso de uso tem por objetivo cadastrar um novo animal no sistema |
| Autor Primário: | Administrador |
| Pré-Condição: | Acessar o sistema e selecionar "Cadastrar Animal" |
| Cenário Principal: | 1. O administrador seleciona a opção "Cadastrar Animal". 2. O sistema apresenta o formulário de cadastro. 3. O administrador preenche os dados do animal. 4. O sistema valida os dados e exibe o botão para salvar. 5. As informações são registradas no banco de |

| | |
|----------------------------|---|
| | dados. 6. O sistema exibe mensagem de sucesso |
| Pós-Condição: | Animal cadastrado e disponível no sistema. |
| Inclusão: | UC 0018 – Inserir dados do animal |
| Extensão: | UC 00196 – Salvar informações |
| Cenário Alternativo | O administrador pode cancelar o cadastro. 3a – Dados obrigatórios ausentes ou inválidos |

Cadastrar Adotante

| | |
|---------------------------|---|
| ID: | UC 013 |
| Descrição: | Este caso de uso tem por objetivo cadastrar um novo adotante no sistema. |
| Autor Primário: | Administrador |
| Pré-Condição: | Acessar o sistema e selecionar "Cadastrar Adotante" |
| Cenário Principal: | 1. O administrador acessa a opção "Cadastrar Adotante". 2. O sistema apresenta o formulário de cadastro de usuário. 3. O administrador insere as informações do adotante. 4. O sistema valida os dados. 5. O sistema salva os dados no banco de dados. 6. O sistema confirma o cadastro |
| Pós-Condição: | Adotante cadastrado com sucesso. |

| | |
|-----------------------------|--|
| Inclusão: | UC 0018 – Inserir dados do usuário |
| Extensão: | UC 0019 – Salvar informações |
| Cenário Alternativo: | a – O administrador pode cancelar o cadastro. 4a – Dados inválidos ou CPF duplicado. |

Cadastrar Doador

| | |
|-----------------------------|--|
| ID: | UC 014 |
| Descrição: | Este caso de uso tem por objetivo cadastrar um novo doador no sistema |
| Autor Primário: | Administrador |
| Pré-Condição: | Acessar o sistema e selecionar "Cadastrar Doador" |
| Cenário Principal: | 1. O administrador seleciona "Cadastrar Doador". 2. O sistema apresenta o formulário de cadastro. 3. O administrador preenche os dados do doador. 4. O sistema valida as informações. 5. O sistema salvo os dados. 6. O sistema confirma o cadastro. |
| Pós-Condição: | Doador cadastrado com sucesso. |
| Inclusão: | UC 0018 – Inserir dados do usuário |
| Extensão: | UC 0019 – Salvar informações |
| Cenário Alternativo: | a – O administrador pode cancelar. 4a – Dados |

| | |
|--|------------------------------------|
| | obrigatórios ausentes ou inválidos |
|--|------------------------------------|

Cadastrar Voluntário

| ID | UC015 |
|-----------------------------|--|
| Descrição: | Este caso de uso tem por objetivo cadastrar um novo voluntário no sistema |
| Autor Primário: | Administrador |
| Pré-Condição: | Acessar o sistema e selecionar "Cadastrar Voluntário" |
| Cenário Principal: | 1. O administrador seleciona "Cadastrar Voluntário". 2. O sistema apresenta o formulário de cadastro. 3. O administrador preenche os dados do voluntário. 4. O sistema valida as informações. 5. O sistema salva os dados. 6. O sistema confirma o cadastro. |
| Pós-Condição: | Voluntário cadastrado com sucesso. |
| Inclusão: | UC 0018– Inserir dados do usuário |
| Extensão: | UC 0019 – Salvar informações |
| Cenário Alternativo: | a – O administrador pode cancelar. 4a – Dados obrigatórios ausentes ou inválidos |

Cadastrar Funcionário da ONG

| ID: | UC 016 |
|-------------------|---|
| Descrição: | Este caso de uso tem por objetivo cadastrar um novo |

| | |
|-----------------------------|---|
| | funcionário da ONG no sistema. |
| Autor Primário: | Administrador |
| Pré-condição: | Acessar o sistema e selecionar "Cadastrar Funcionário da ONG" |
| Cenário Principal: | <p>1. O administrador seleciona "Cadastrar Funcionário da ONG".</p> <p>2. O sistema apresenta o formulário de cadastro.</p> <p>3. O administrador insere as informações do diretor.</p> <p>4. O sistema valida os dados.</p> <p>5. O sistema salva os dados.</p> <p>6. O sistema exibe mensagem de confirmação.</p> |
| Pós-condição: | Diretor da ONG cadastrado e com acesso ao sistema. |
| Cenário Alternativo: | *a – O administrador pode cancelar o cadastro. 4a – Dados obrigatórios não preenchidos. |
| Inclusão: | UC 0018 – Inserir dados do usuário |
| Extensão: | UC 0019 – Salvar informações |

Inserir dados dos animais

| | |
|-------------------|--|
| ID: | UC 017 |
| Descrição: | Este caso de uso tem por objetivo, inserir e registrar |

| | |
|-----------------------------|---|
| | dados dos animais no banco de dados |
| Ator Primário: | Sistema |
| Pré-condição: | Existir banco de dados e cadastrar animais |
| Cenário Principal: | <ol style="list-style-type: none"> 1. O sistema recolhe dados dos animais resgatados 2. O sistema insere os dados recolhidos no banco de dados 3. O sistema salva os dados no banco de dados |
| Pós-condição: | Animal Cadastrado no Sistema |
| Cenário Alternativo: | *a – Cancelamento de cadastro 4a – Dados preenchidos incorretamente |
| Inclusão: | Não Possui |
| Extensão: | UC019 |

Inserir Dados do Usuário

| | |
|---------------------------|---|
| ID: | UC018 |
| Descrição: | Insere Dados do Usuário que irá logar no sistema |
| Ator Primário: | Sistema |
| Pré-condição: | O usuário efetua o login |
| Cenário Principal: | <ol style="list-style-type: none"> 1. O usuário loga no sistema 2. O usuário insere seus dados de senha e login 3. O sistema recebe os dados |
| Pós-condição: | O sistema recebe os dados do usuário |

| | |
|-----------------------------|---------------------------------|
| Cenário Alternativo: | *a – falha no login ou cadastro |
| Inclusão: | UC 0018 |
| Extensão: | Não Possui |

Salvar Informações de Cadastro

| | |
|-----------------------------|---|
| ID: | UC 019 |
| Descrição: | Salva os dados de login e cadastro no banco de dados |
| Autor Primário: | Sistema |
| Pré-condição: | Não Possui |
| Cenário Principal: | <p>1. O sistema busca os dados de login e cadastro de usuário</p> <p>2. O sistema guarda e salva os dados no banco de dados</p> |
| Pós-condição: | Dados dos usuários salvos e seguros |
| Cenário Alternativo: | Não Possui |
| Inclusão: | Não Possui |
| Extensão: | Não Possui |

Acessar Área do Adotante

| | |
|------------------------|---|
| ID: | UC 020 |
| Descrição: | Permite acessar a área com informações sobre animais disponíveis para adoção. |
| Autor Primário: | Adotante |
| Pré-condição: | Acesso a área de adoção (cadastrado). |

| | |
|-----------------------------|---|
| Cenário Principal: | 1. Usuário acessa o site da ONG. 2. Clica no menu ou botão "Quero Adotar". 3. Sistema exibe a área de adoção com informações dos animais disponíveis. |
| Pós-condição: | Área de adoção é exibida |
| Cenário Alternativo: | Sistema identifica localização do usuário e prioriza animais da mesma região. |
| Inclusão: | UC 0022– Escolher animal |
| Extensão: | Não possui. |

Preencher Formulário

| | |
|---------------------------|---|
| ID: | UC021 |
| Descrição: | O usuário preenche um formulário com seus dados, renda e informações sobre o ambiente onde o animal viverá. |
| Autor Primário: | Adotante. |
| Pré-condição: | Animal selecionado. |
| Cenário Principal: | 1. O sistema exibe o formulário de adoção. 2. O usuário preenche os campos obrigatórios (nome, endereço, tipo de moradia, renda, etc.). 3. Confirma os dados e envia. |
| Pós-condição: | Dados do formulário são validados e salvos. |

| | |
|-----------------------------|--|
| Cenário Alternativo: | Sistema sugere salvar rascunho para envio posterior. |
| Inclusão: | UC 0023 – Enviar solicitação de adoção |
| Extensão: | Não possui. |

Escolher animal

| | |
|-----------------------------|---|
| ID: | UC 022 |
| Autor Primário: | Adotante. |
| Descrição: | Permite acessar a área com informações sobre animais disponíveis para adoção. |
| Pré-Condição: | Sistema e site online |
| Cenário Principal: | <p>1. O sistema exibe a lista de animais disponíveis.</p> <p>2. O usuário navega pelas opções e clica em um animal de interesse.</p> <p>3. O sistema exibe informações detalhadas sobre o animal.</p> |
| Pós Condição: | Detalhes do animal são exibidos. |
| Cenário Alternativo: | Sistema recomenda animais similares ao selecionado. |
| Inclusão: | Sistema recomenda animais similares ao selecionado. |
| Extensão: | Não possui |

Caso De Uso – Enviar Solicitação de Adoção

| | |
|-----------------------------|---|
| ID: | UC023 |
| Autor Primário: | Adotante |
| Descrição: | Após preencher o formulário, o usuário envia a solicitação de adoção para análise da ONG. |
| Pré-Condição: | Formulário preenchido. |
| Cenário Principal: | <ol style="list-style-type: none">1. Usuário clica em "Enviar Solicitação".2. O sistema armazena os dados e envia uma notificação para a equipe da ONG.3. Sistema exibe mensagem de confirmação ao usuário. <p>Exceções: Erro no servidor impede o envio — sistema exibe mensagem de falha.</p> |
| Pós Condição: | Solicitação registrada para análise. |
| Cenário Alternativo: | Sistema sugere salvar rascunho para envio posterior. |
| Inclusão: | Analizar formulário |
| Extensão: | Não possui |

Caso De Uso – Buscar/ Receber Animal

| | |
|-----------------------------|---|
| ID: | UC 024 |
| Autor Primário: | Adotante e Administrador |
| Descrição: | Após aprovação da solicitação, o usuário busca o animal ou agenda entrega com a ONG. |
| Pré-Condição: | Solicitação de adoção aprovada. |
| Cenário Principal: | <ol style="list-style-type: none">1. Sistema notifica o usuário sobre a aprovação.2. Usuário escolhe entre retirar o animal ou receber em casa.3. ONG agenda data e horário. Entrega ou retirada ocorre conforme combinado. |
| Pós Condição: | Animal é retirado ou entregue |
| Cenário Alternativo: | ONG oferece visita para ajudar a adaptar animal ao novo ambiente. |
| Inclusão: | UC025 – Assinar termo de responsabilidade |
| Extensão: | Não possui. |

Assinar Termos de Responsabilidade

| | |
|------------------------|--|
| ID: | UC025 |
| Descrição: | O adotante assina um termo de responsabilidade pelo animal |
| Autor Primário: | Adotante |

| | |
|-----------------------------|--|
| Pré-Condição: | Estar com o animal em casa e ter o formulário de adotante aprovado |
| Cenário Principal: | <ol style="list-style-type: none"> 1. O adotante recebe ou busca o cachorro 2. O adotante assina os termos de responsabilidade |
| Pós-condição: | O adotante dá um lar e uma nova vida para o animal |
| Cenário Alternativo: | O adotante desiste da adoção |
| Inclusão | Nenhuma |
| Extensão: | Nenhuma |

Acessar área de voluntários

| | |
|-----------------------------|---|
| ID: | ID- UC026 |
| Descrição: | Permite ao voluntario acessar a área dedicada ao voluntario no sistema. |
| Autor Primário: | Voluntario |
| Pré-condição: | Estar logado no sistema como voluntário |
| Cenário Principal: | <ol style="list-style-type: none"> 1. O voluntário acessa a seção de voluntariado 2. O sistema exibe as opções para cadastro e atividades |
| Pós-condição: | Área de voluntario disponível para o usuário |
| Cenário Alternativo: | Sistema indisponível |
| Inclusão: | UC027 - Preencher formulário |
| Extensão: | Nenhuma |

Preencher Formulário

| | |
|-----------------------------|--|
| ID: | UC027 |
| Descrição: | Permite ao voluntário informar seus dados e interesses no formulário de voluntariado |
| Autor Primário: | Voluntário |
| Pré-condição: | Estar na área de voluntários. |
| Cenário Principal: | 1. O voluntário acessa o formulário. 2. Preenche as informações solicitadas. |
| Pós-condição: | Dados prontos para envio ao sistema. |
| Cenário Alternativo: | Falta de preenchimento obrigatório |
| Inclusão: | UC 028 – Enviar Solicitação de Voluntário |
| Extensão: | Nenhuma |

Enviar Solicitação de Voluntário

| | |
|---------------------------|--|
| ID: | UC028 |
| Descrição: | Permite ao voluntário enviar seus dados para cadastro e análise da solicitação. |
| Autor Primário: | Voluntário |
| Pré-condição: | Formulário preenchido corretamente. |
| Cenário Principal: | 1. O voluntário clica em "Enviar". 2. O sistema registra e processa a solicitação. |
| Pós-condição: | Solicitação armazenada para análise. |

| | |
|-----------------------------|--|
| Cenário Alternativo: | Problemas no envio ou dados inválidos. |
| Inclusão: | UC 029 – Realizar Tarefas Solicitadas |
| Extensão: | Nenhuma |

Realizar Tarefas Solicitadas

| | |
|-----------------------------|--|
| ID: | UC029 |
| Descrição: | : Permite que o voluntário execute as tarefas atribuídas conforme o aceite da solicitação. |
| Autor Primário: | Voluntário |
| Pré-condição: | Solicitação aceita e tarefa atribuída. |
| Cenário Principal: | 1. O voluntário acessa suas tarefas. 2. Executa as atividades conforme descrição. |
| Pós-condição: | Tarefa concluída e registrada. |
| Cenário Alternativo: | Voluntário não realiza as tarefas no prazo. |
| Inclusão: | Nenhuma |
| Extensão: | Nenhuma |

ACESSAR ÁREA DE DOAÇÕES

| | |
|------------|------------------|
| ID: | UC - 0030 |
|------------|------------------|

| | |
|-----------------------------|--|
| Descrição: | Permite que o usuário acesse a área de doações no site da ONG. |
| Autor Primário: | Doador |
| Pré-condição: | Site da ONG está online. |
| Cenário Principal: | <p>1. O usuário acessa o site da ONG.</p> <p>2. Clica na opção "Doar" ou equivalente.</p> <p>3. O sistema exibe a área de doações com as opções disponíveis.</p> <p>Exceções: Sistema fora do ar — mensagem de erro é exibida.</p> |
| Pós-condição: | A área de doação é exibida ao usuário. |
| Cenário Alternativo: | Sistema exibe mensagens personalizadas com campanhas atuais. |
| Inclusão: | UC 0031 – Inserir valor desejado |
| Extensão: | UC 005 – Informar recurso a ser doado |

CASO DE USO – INSERIR VALOR DESEJADO

| | |
|------------------------|---|
| ID: | UC 0031 |
| Descrição: | Permite ao usuário inserir o valor da doação. |
| Autor Primário: | Doador |

| | |
|-----------------------------|--|
| Pré-condição: | Usuário acessou a área de doações. |
| Cenário Principal: | <ol style="list-style-type: none"> O sistema exibe o campo de valor da doação. O usuário insere o valor desejado. Exceções: Valor não inserido ou inválido — sistema exibe alerta. |
| Pós-condição: | Valor registrado para a doação. |
| Cenário Alternativo: | Sistema sugere valores padrões. |
| Inclusão: | UC 0032 – Inserir forma de pagamento |
| Extensão: | Não possui |

CASO DE USO – INSERIR FORMA DE PAGAMENTO

| | |
|---------------------------|---|
| ID: | UC 0032 |
| Descrição: | Permite inserir informações sobre o método de pagamento. |
| Autor Primário: | Doador |
| Pré-condição: | Valor da doação informado |
| Cenário Principal: | <ol style="list-style-type: none"> O sistema exibe opções de pagamento (cartão, PIX, boleto). O usuário seleciona uma opção e insere os dados solicitados. Exceções: Dados incompletos ou |

| | |
|-----------------------------|--|
| | inválidos — alerta é exibido. |
| Pós-condição: | Forma de pagamento cadastrada para a doação. |
| Cenário Alternativo: | Sistema oferece salvar os dados para doações futuras |
| Inclusão: | UC 0033 – Finalizar doação |
| Extensão: | Não possui. |

CASO DE USO – FINALIZAR DOAÇÃO

| | |
|-----------------------------|--|
| ID: | UC 0033 |
| Descrição: | Permite concluir o processo de doação. |
| Autor Primário: | DO 0034 |
| Pré-condição: | Valor e forma de pagamento definidos |
| Cenário Principal: | <ol style="list-style-type: none"> O sistema exibe o resumo da doação. O usuário confirma os dados e clica em "Finalizar". O sistema processa e confirma a doação. <p>Exceções: Erro no processamento do pagamento.</p> |
| Pós-condição: | Doação realizada com sucesso |
| Cenário Alternativo: | Sistema envia comprovante por e-mail. |
| Inclusão: | Fim |
| Extensão: | Não possui |

CASO DE USO – INFORMAR RECURSO A SER DOADO

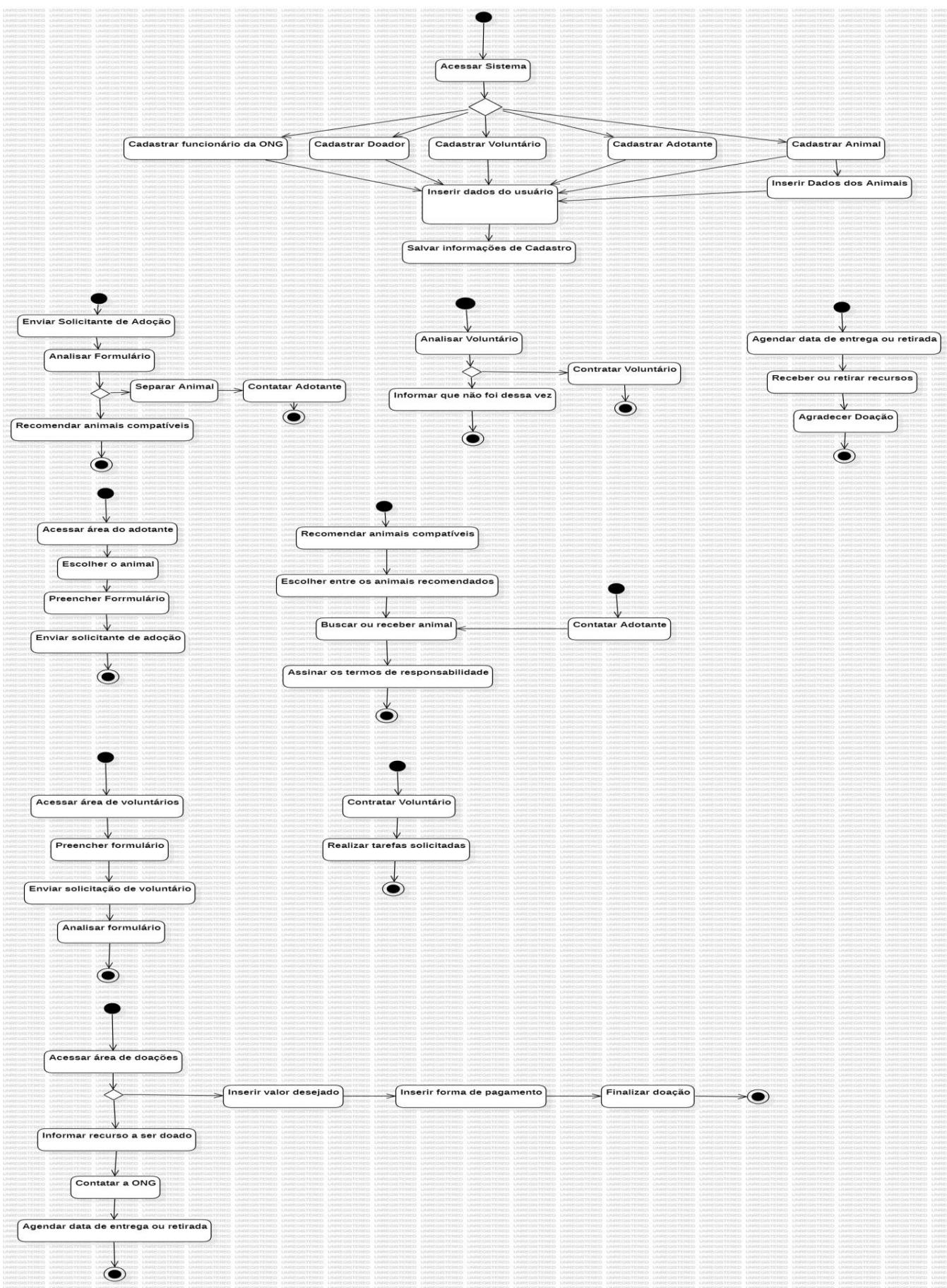
| | |
|-----------------------------|--|
| ID: | DO 0034 |
| Descrição: | Permite informar itens físicos (como ração, brinquedos) que serão doados |
| Autor Primário: | Doador |
| Pré-condição: | Acesso à área de doação. |
| Cenário Principal: | <ol style="list-style-type: none">1. O sistema exibe opção "Doar recursos físicos".2. Usuário insere informação do item. Exceções: Informação não preenchida. |
| Pós-condição: | Dados do recurso registrados |
| Cenário Alternativo: | Sistema exibe lista de recursos mais necessários. |
| Inclusão: | UC 0035 – Contatar ONG |
| Extensão: | UC 0035 – Contatar ONG |

CASO DE USO – CONTATAR ONG

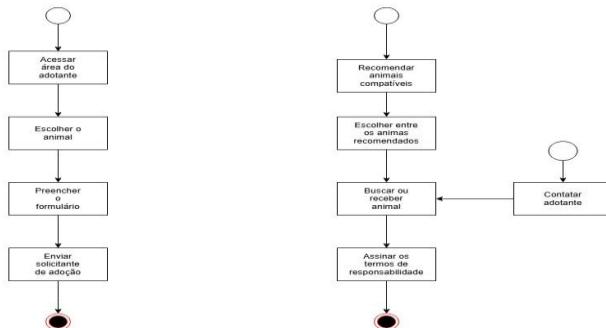
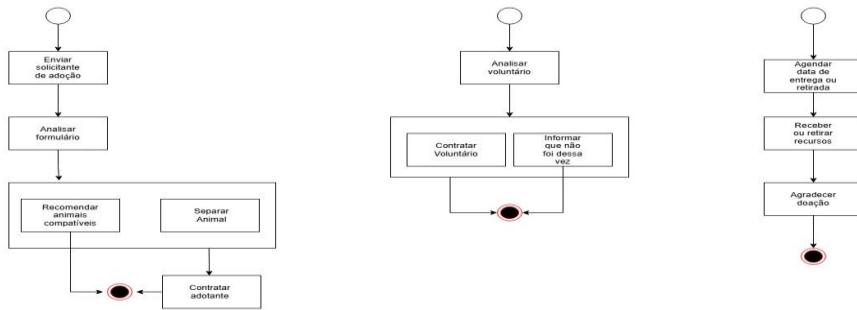
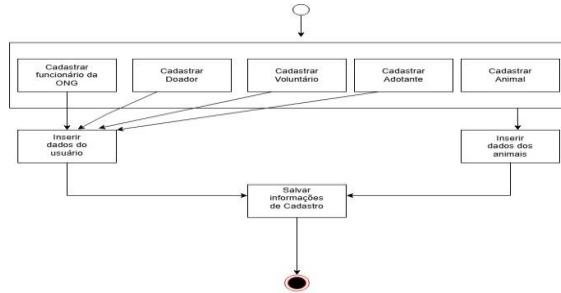
| | |
|---------------------------|--|
| ID: | UC 0035 |
| Descrição: | Usuário aciona a ONG para combinar a entrega do recurso. |
| Autor Primário: | Doador |
| Pré-condição: | Dados do contato preenchidos |
| Cenário Principal: | <ol style="list-style-type: none">1. O sistema envia os dados à ONG. |

| | |
|-----------------------------|---|
| | <p>2. ONG entra em contato com o doador.</p> <p>Exceções: Falha no envio dos dados.</p> |
| Pós-condição: | ONG ciente da intenção de doação. |
| Cenário Alternativo: | Sistema envia cópia do pedido ao e-mail do usuário. |
| Inclusão: | UC009 Agendar data de entrega / retirada |
| Extensão: | Não possui. |

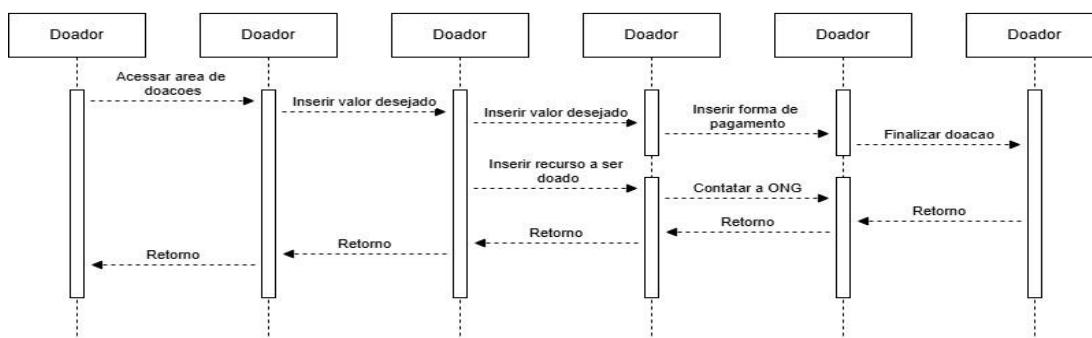
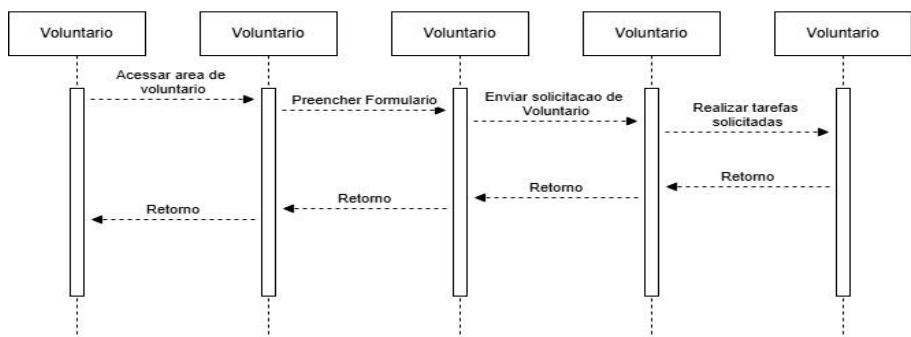
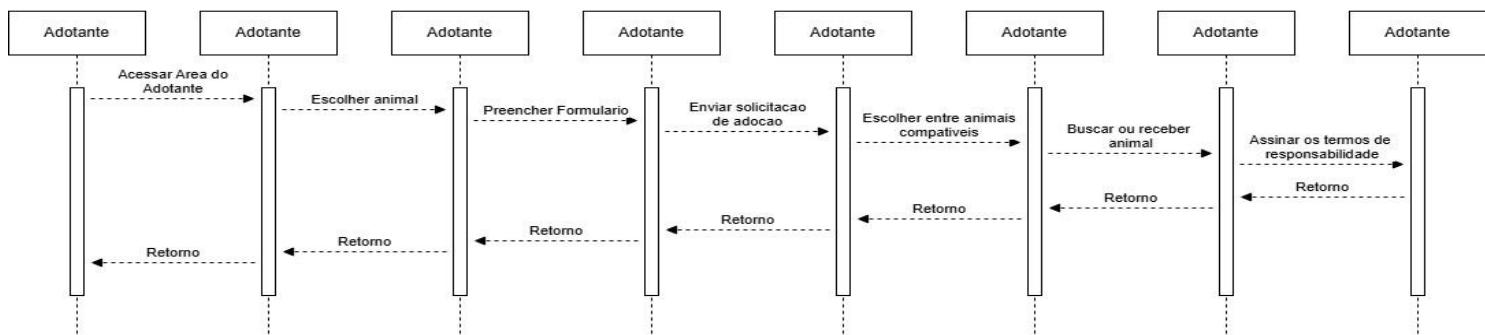
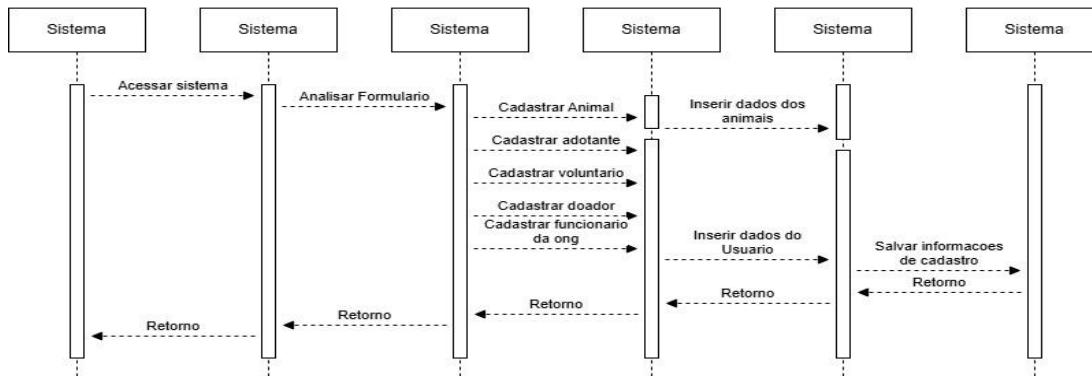
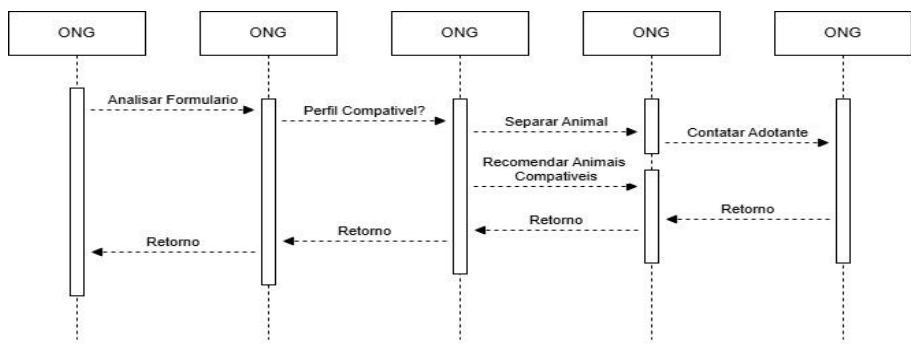
1.9-DIAGRAMA DE ATIVIDADE



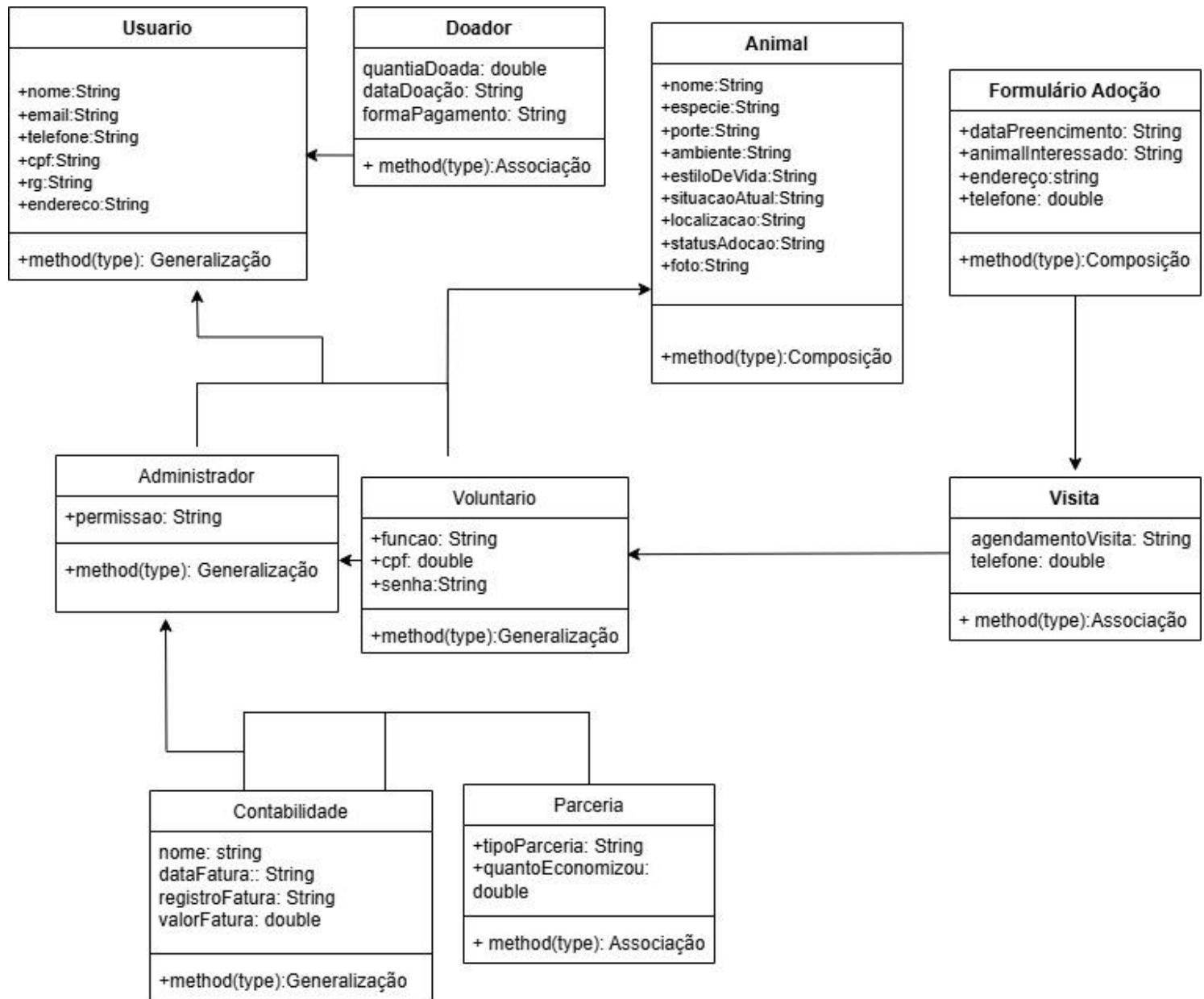
1.10-MÁQUINA DE ESTADO:



1.11-DIAGRAMA DE SEQUÊNCIA



1.12 DIAGRAMA DE CLASSE



1.13-REVISÃO DE PORTABILIDADE

1. Dados da empresa

Nome da empresa: Associação Só Patinhas de rua

Nome do contato: Lindsay Cardoso

Telefone: +55 16 99401-6763

2. Infraestrutura

Pontos de acesso: 1 ponto (Local: Área administrativa)

Meio de transmissão: WiFi padrão 2.4GHz, com alcance de até 50 metros em área aberta

Velocidade: 100mbps

Equipamento: Não possui

3. Infraestrutura

Hardware

-

Rede

elétrica:

| | |
|-----------------------|---------|
| 12 | Tomadas |
| 1 Disjuntor no quadro | |
| 16 Lâmpadas | |

Estabilizadores: Não possui

Nobreaks: Não possui

4.

Infraestrutura

Hardware: Notebooks pessoais: 2
Celulares Smartphone: 3

Arquitetura: processador: i3-1215U

Memória: 1 HD 1TB 1 - SSD 550GB – 256GB Memória Smartphones

Servidores: Não possui

5. Infraestrutura

Hardware: 2 mouses, 2 teclados, 2 fones de ouvido

Impressora: Possui 1 pessoal

Scanner: Não possui

6. Licenças

Photoshop

Windows

7. Recursos Humanos

Quantidade de usuários: 3

Nível de alfabetização digital: Uso Funcional da Tecnologia

8. Requisitos Mínimos

Navegador instalado

Conexão com a internet

Celular (Android 6+ ou iOS 12+)

Computador com Windows 7+, macOS 10.12+ ou distribuições Linux

mais recentes

1.14- FOTOS













DICIONÁRIO DE DADOS

2.1 PRD

1. Resumo

Este documento descreve os requisitos para o desenvolvimento de um sistema de software para a **Associação Só Patinhas de Rua**. A solução visa atender às necessidades críticas identificadas na ONG, como a **deficiência na organização interna**, notadamente a **ausência de um sistema de catalogação dos animais**, a **falta de voluntários** e a **falta de recursos financeiros**.

A proposta consiste em um sistema duplo: uma plataforma de gestão interna para a instituição (catalogação de animais e gerenciamento de voluntários) e um sistema externo (site) para dar visibilidade ao projeto, facilitar campanhas de adoção e arrecadação de recursos. O objetivo geral é otimizar a gestão da ONG, proporcionando maior controle sobre os animais abrigados e as atividades diárias, garantindo a sustentabilidade do projeto.

2. Histórias de Usuário (User Stories)

Como um gestor da ONG, eu quero um sistema para **catalogar os 300 animais**, registrando histórico de saúde, vacinas e castrações, para

facilitar a gestão, o controle de entradas e saídas e a tomada de decisões.

Como um gestor da ONG, eu quero uma plataforma para **gerenciar voluntários**, facilitando a organização de tarefas como limpeza, alimentação e acompanhamento veterinário, para reduzir a sobrecarga dos 3 colaboradores fixos.

Como um visitante ou potencial doador, eu quero acessar um site para conhecer a ONG, ver os animais disponíveis para adoção e entender claramente como doar (ração, medicamentos, recursos financeiros), para me sentir motivado a apoiar o projeto.

Como um voluntário, eu quero acessar uma área no sistema para registrar, ter a lista e fazer o controle e a gestão dos animais.

3. Requisitos Funcionais e Módulos

O sistema será dividido nos seguintes módulos:

Módulo de Gestão de Animais (Catalogação)

O sistema deverá permitir o cadastro completo do animal, incluindo nome, espécie (cão/gato), fotos, idade, gênero e data do resgate.

Deverá ser possível alterar o status do animal (ex: "Em tratamento", "Disponível para adoção").

O sistema deverá permitir a busca e filtragem de animais por diferentes critérios.

Deverá ser possível registrar o controle de entradas (resgates) e saídas (adoções).

Módulo de Gestão de Voluntários

O sistema deverá permitir o cadastro de novos voluntários.

Deverá permitir que gestores criem e atribuam tarefas (limpeza, alimentação, transporte, administrativo).

O sistema deverá ter um calendário de atividades em que voluntários possam se inscrever.

Portal de Visibilidade (Site Externo e Adoção)

A tela inicial deverá exibir informações sobre a ONG, sua missão e seus 15 anos de atuação.

Deverá exibir uma "vitrine" de animais disponíveis para adoção, integrada ao Módulo de Gestão de Animais.

O sistema deverá ter uma seção clara para arrecadação de recursos, explicando a necessidade de doações para ração, medicamentos e vacinas.

4. Critérios de Aceite

O sistema deve permitir o cadastro de um novo animal, coletando todos os campos obrigatórios (Nome, espécie, idade e etc.).

O acesso à funcionalidade de alterar dados ou excluir um animal deve ser exclusivo para usuários com perfil de "Gestor".

Ao registrar uma adoção, o sistema deve mover o animal do status "Disponível" para "Adotado" e atualizar o controle de saídas.

Usuários com perfil "Voluntário" devem conseguir visualizar as tarefas, mas não devem ter permissão para criar novas tarefas.

O portal externo deve exibir publicamente apenas os animais com status "Disponível para adoção".

5. Fora do Escopo para esta Etapa

Módulo Financeiro-Contábil: O sistema não incluirá funcionalidades para gestão de fluxo de caixa, pagamento de despesas ou contabilidade. O foco é na arrecadação, não na gestão financeira completa.

Controle de Estoque: Um módulo detalhado para controle de estoque de ração e medicamentos não será desenvolvido nesta fase.

Aplicativo Móvel Nativo: A solução será desenvolvida como um sistema web. Não haverá o desenvolvimento de um aplicativo para iOS ou Android.

JORNADAS DE USUÁRIOS

Jornada 1: O Visitante / Potencial Doador

Histórias de Usuário:

Como um potencial doador, eu quero acessar o site e entender rapidamente o propósito da "Só Patinhas de Rua" e por que ela precisa de ajuda financeira.

Como um membro da comunidade, eu quero ver facilmente quais animais estão disponíveis para adoção.

Como um interessado, eu quero encontrar informações claras sobre como doar (ração, medicamentos, dinheiro).

Requisitos Funcionais e Interações:

Tela Inicial: Apresentar a ONG e os desafios (falta de recursos , 300 animais).

Página "Adote": Galeria de animais (integrada ao Módulo de Gestão) com fotos e breve histórico.

Página "Como Ajudar": Detalhes sobre a necessidade de doações (ração, vacinas, etc.) e voluntariado, com links claros para doação.

Fora do Escopo:

Área de login para doadores.

Sistema de depoimentos de adotantes.

Jornada 2: O Voluntário

Histórias de Usuário:

Como um novo voluntário, eu quero me cadastrar na plataforma de forma simples para poder ajudar.

Como um voluntário cadastrado, eu quero acessar uma área pessoal para visualizar as tarefas disponíveis (limpeza, alimentação, etc.) e me inscrever nelas.

Como um voluntário, eu quero registrar quando concluí uma tarefa, para que o gestor saiba o que foi feito.

Requisitos Funcionais e Interações:

Fluxo de Cadastro: Formulário simples de cadastro de voluntário.

Painel do Voluntário (Área Logada):

Mural de Tarefas: Exibe tarefas postadas pelos gestores (ex: "Ajuda na limpeza sábado", "Transporte para veterinário").

Minhas Tarefas: Lista de tarefas nas quais o voluntário se inscreveu.

Fora do Escopo:

Chat direto com gestores.

Upload de documentos ou relatórios pelo voluntário.

Jornada 3: O Gestor da ONG (Colaborador Fixo)

Histórias de Usuário:

Como gestor, eu quero acessar o sistema para cadastrar um novo animal resgatado, inserindo seu histórico de saúde.

Como gestor, eu quero atualizar o status de um animal (ex: "Vacinado", "Castrado", "Adotado") para manter o sistema de catalogação funcional.

Como gestor, eu quero criar novas tarefas para voluntários (ex: "Campanha de conscientização") e ver quem se inscreveu.

Como gestor, eu quero monitorar o status de todos os 300 animais de forma rápida, para facilitar a tomada de decisões.

Requisitos Funcionais e Interações:

Painel Administrativo (Dashboard):

Gestão de Animais: Ferramentas para criar, editar e buscar animais no catálogo.

Gestão de Voluntários: Visualizar voluntários cadastrados, criar e gerenciar tarefas.

Controle de Adoções: Aprovar e registrar saídas de animais.

Fora do Escopo:

Módulo financeiro para controle de doações.

Relatórios de desempenho detalhados sobre voluntários.

FLUXOS DE OPERAÇÕES

Processo de Resgate (Entrada)

1.1 Resgate: O Gestor da ONG realiza o resgate de um novo animal.

1.2 Catalogação: O Gestor acessa o sistema e "Cadastra Animal", preenchendo o histórico de saúde inicial e definindo o status como "Em Acolhimento" ou "Em Tratamento".

Ciclo de Cuidados e Voluntariado

2.1 Definição de Tarefas: O Gestor identifica necessidades (limpeza, alimentação, vacinação).

2.2 Atuação do Voluntário: O Voluntário acessa o sistema e faz registro do animal.

Processo de Adoção (Saída)

3.1 Disponibilização: Após o tratamento, o Gestor "Altera Status" do animal para "Disponível para Adoção". O animal aparece no site externo.

3.2 Adoção: Um interessado adota o animal.

3.3 Efetivação: O Gestor acessa o sistema, localiza o animal e registra a "Saída por Adoção", atualizando o controle.

DEFINIÇÃO DO MODELO DE DADOS

Tabela 1: Voluntários

Propósito:

Armazenar os dados dos voluntários que auxiliam o abrigo.

Colunas:

- **id** — Chave Primária
- **nome** — Nome completo do voluntário
 - **telefone** — Contato telefônico
 - **senha** — Senha de acesso ao sistema
 - **usuário** — Identificador único de acesso (login)

Tabela 2: Adotante

Propósito:

Registrar pessoas interessadas ou já responsáveis pela adoção dos animais.

Colunas:

- **id** — Chave Primária
- **nome** — Nome completo
- **endereço** — Endereço residencial
 - **telefone** — Contato telefônico
 - **cep** — CEP do endereço
- **e-mail** — E-mail único para contato e identificação

Tabela 3: Animal

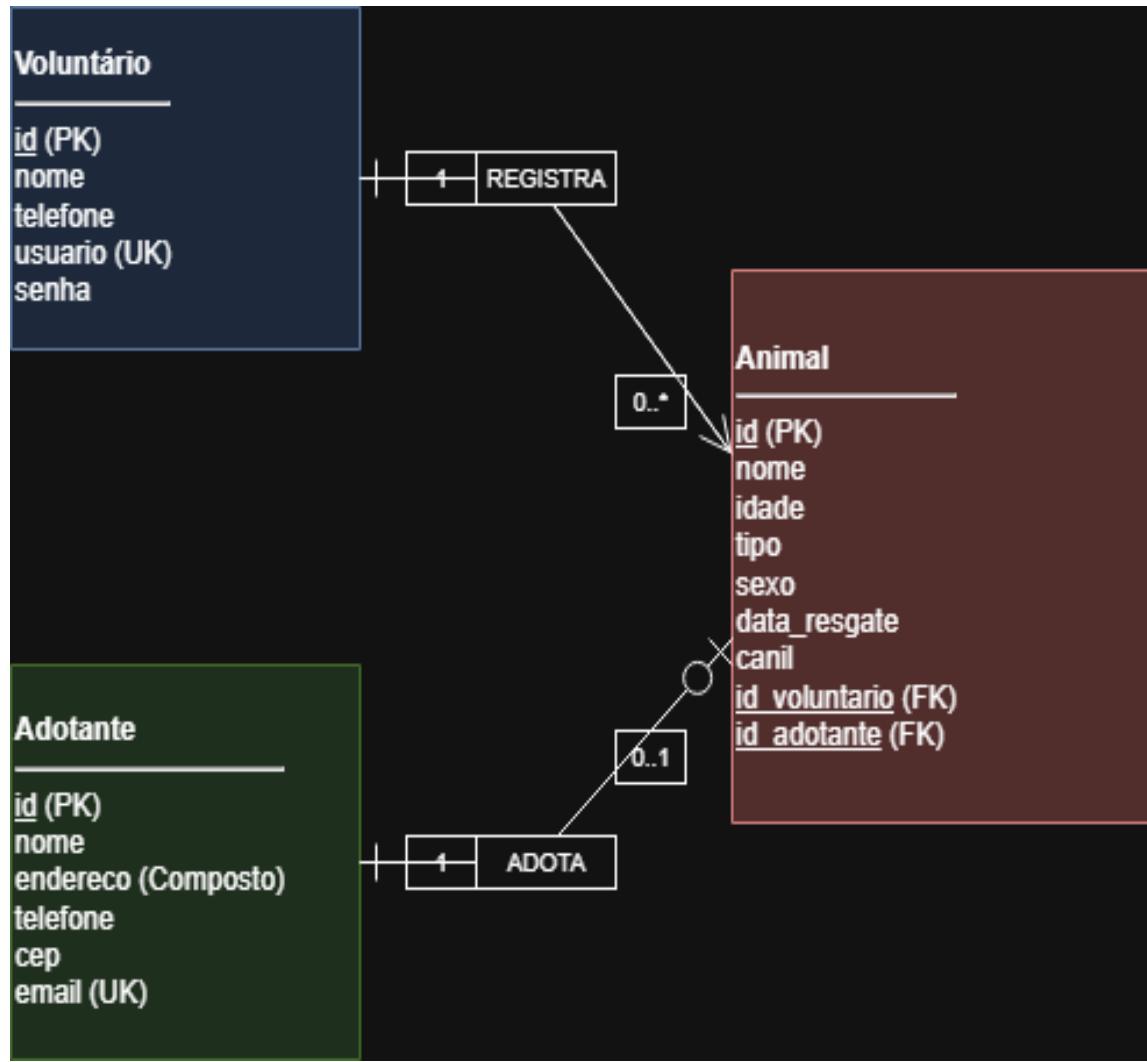
Propósito:

Armazenar todas as informações cadastrais dos animais atendidos pelo abrigo.

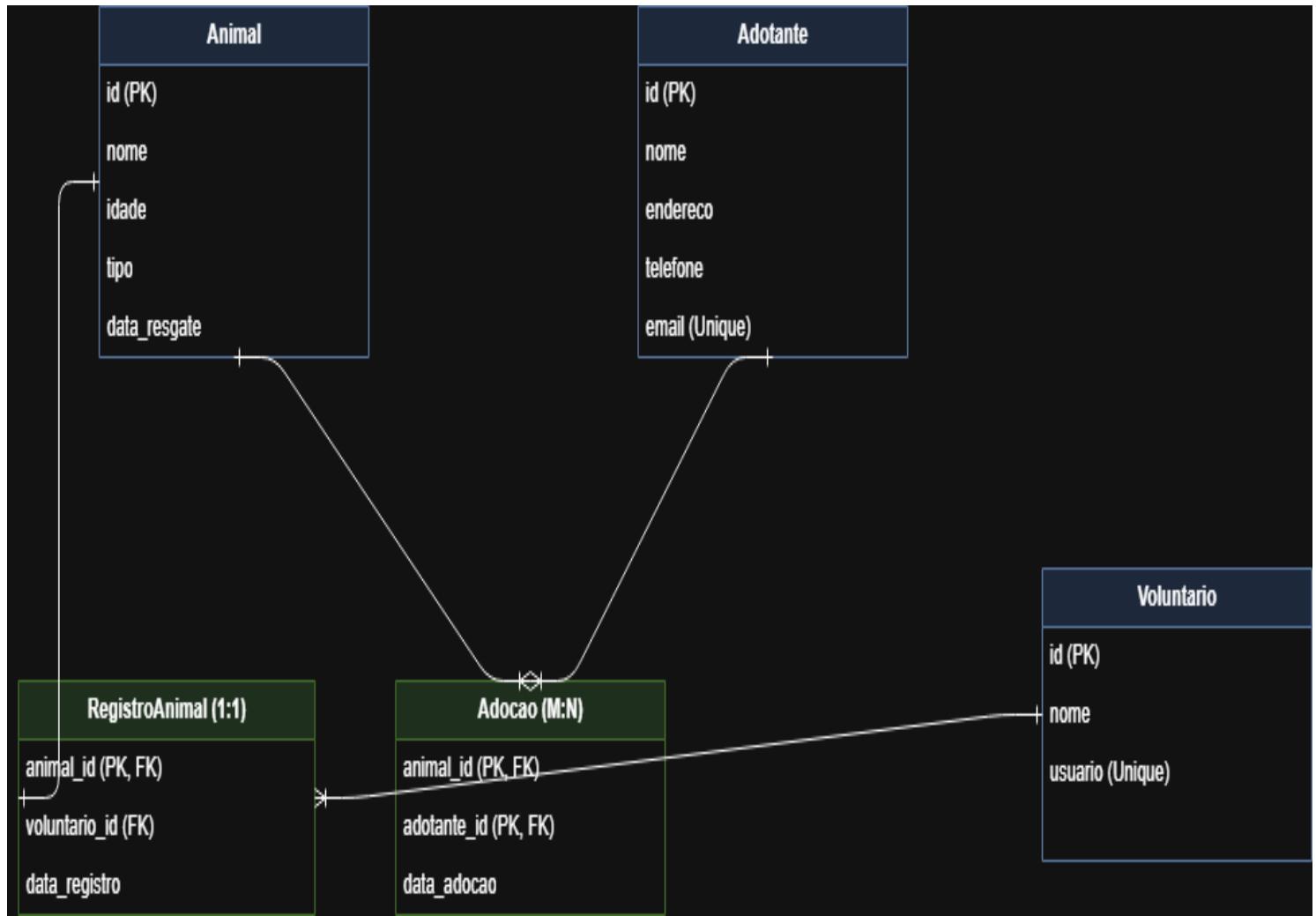
Colunas:

- **id** — Chave Primária
- **nome** — Nome do animal
- **idade** — Idade estimada
- **tipo** — Tipo/Espécie (ex.: cão, gato)
 - **sexo** — Sexo do animal
- **data_resgate** — Data em que o animal foi resgatado
- **canil** — Localização ou setor onde o animal está abrigado

2.2 - MER



2.3 - DER



3-LINGUAGEM DE PROGRAMAÇÃO

3.1 IMPLEMENTAÇÃO E TESTE DA SOLUÇÃO DESENVOLVIDA

Definição das tecnologias utilizadas

1. Linguagem Java Script

Utilizamos o Framework NEST JS para a construção da api. A estrutura desse framework funciona da seguinte forma:

Para cada entidade — Animal, Voluntário, Adotante — criamos um conjunto de arquivos seguindo o padrão do Nest:

Módulo: organiza o CRUD daquele recurso

Controlador: recebe requisições HTTP e define ROTAS

Serviço: contém a lógica de negócio

DTOs: definem o formato de entrada e saída e é o acesso direto ao banco de dados

Modelos/Schema (ou Prisma/ORM): definem estrutura no banco e faz a sua criação via migrations.

Isso mantém tudo separado e bem organizado.

2. Protocolo Prisma

Para gerenciar e interagir com o banco de dados, utilizamos o **Prisma**, um ORM (Object-Relational Mapping) moderno, rápido e muito produtivo. Ele substitui a necessidade de escrever SQL manualmente, oferecendo uma interface tipada e intuitiva para realizar operações de leitura e escrita.

No contexto da nossa API NestJS, o Prisma desempenha um papel fundamental no funcionamento do CRUD das entidades **Animal**, **Voluntário** e **Adotante**.

A seguir, explicamos como ele funciona e como se integra ao fluxo da aplicação.

No Prisma, todos os modelos do banco de dados são definidos dentro de um arquivo central chamado `schema.prisma`.

Para cada entidade da API — Animal, Voluntário e Adotante — criamos um modelo representando sua estrutura no banco:

```
model Animal {
    id      Int      @id @default(autoincrement())
    nome    String
    especie String
    idade   Int
    status  String
}

model Voluntario {
    id      Int      @id @default(autoincrement())
    nome    String
    telefone String
    email   String
}

model Adotante {
    id      Int      @id @default(autoincrement())
    nome    String
    endereco String
    telefone String
}
```

Esses modelos definem:

- tipos dos campos,
- chaves primárias,
- relacionamentos,
- padrões de criação (como `@default(now())`).

Eles funcionam como a base da estrutura da aplicação.

3. Banco de Dados PostgreSQL

Para armazenar e gerenciar todos os dados da nossa API — incluindo as entidades **Animal**, **Voluntário** e **Adotante** — utilizamos o **PostgreSQL**, um dos bancos de dados relacionais mais robustos, seguros e amplamente utilizados no mercado.

A hospedagem desse banco é feita por meio da plataforma **Railway**, que oferece um ambiente moderno, simples e altamente escalável para projetos de backend.

A seguir, detalhamos como essa infraestrutura funciona e por que ela foi escolhida.

O **Railway** é uma plataforma moderna de deploy e infraestrutura que facilita a criação e gerenciamento de bancos de dados com poucos cliques.

Ele oferece:

- Banco de dados PostgreSQL pronto para uso
- URL de conexão segura
- Painel web intuitivo
- Monitoramento em tempo real
- Backups automáticos
- Escalabilidade para ambientes de produção
- Integração simples com aplicações Node.js / NestJS

Isso nos permite focar no desenvolvimento sem nos preocupar com configurações complexas de infraestrutura.

ssa URL é configurada no arquivo `.env` da aplicação, por exemplo:

```
DATABASE_URL="postgresql://usuario:senha@host.b.internal:5432/mydb?schema=public"
```

O **Prisma** utiliza essa URL para se conectar ao banco, permitindo:

- criação de tabelas via migrations,
- inserção de dados,
- execução de consultas,
- atualização de registros,
- exclusões,
- e todo o restante do CRUD.

4. Padrão de Projeto DAO (Data Access Project)

O **DAO (Data Access Object)** é um padrão de projeto amplamente utilizado no desenvolvimento de aplicações que precisam interagir com bancos de dados. Seu principal objetivo é **isolar a lógica de acesso aos dados** do restante da aplicação, centralizando toda a comunicação com o banco em classes ou componentes específicos. Esse padrão torna o código mais organizado, reutilizável, fácil de manter e independente da tecnologia de armazenamento utilizada.

O ciclo funciona assim:

1. A **camada de negócios** (Service) solicita uma operação (ex.: buscar um Animal).
2. O **DAO** executa a operação diretamente no banco.
3. O DAO retorna os resultados para a camada de negócio.
4. A camada de negócio processa as regras e envia para o Controller.
5. O Controller responde ao cliente.

Esse fluxo mantém as responsabilidades bem definidas e separadas.

No contexto do nosso sistema, poderíamos ter:

- **AnimalDAO** — responsável pelo CRUD de animais
- **VoluntarioDAO** — responsável pelo CRUD de voluntários
- **AdotanteDAO** — responsável pela manipulação dos dados dos adotantes

Cada DAO conteria exclusivamente a lógica de acesso ao banco, enquanto os **Services** ficariam com as regras de negócio e validações.

Mesmo utilizando o **Prisma**, que já atua como uma camada intermediária, o DAO pode existir como uma camada adicional para manter o código ainda mais organizado em grandes projetos.

Explicação do código-fonte desenvolvido

1. Classe ConnectionFactory

```
import { Injectable, OnModuleInit, OnModuleDestroy, Logger } from '@nestjs/common';
import { PrismaClient } from '@prisma/client';

@Injectable()
export class PrismaService extends PrismaClient implements OnModuleInit, OnModuleDestroy {
  private readonly logger = new Logger(PrismaService.name);

  async onModuleInit() {
    try {
      await this.$connect();
      this.logger.log('✅ Conectado ao banco de dados com sucesso!');
    } catch (error) {
      this.logger.error('✖ Erro ao conectar ao banco de dados:', error.message);
      this.logger.warn('⚠ Verifique se:');
      this.logger.warn('  1. O banco de dados está rodando');
      this.logger.warn('  2. As credenciais no .env estão corretas');
      this.logger.warn('  3. A rede permite conexão ao servidor');
      throw error;
    }
  }

  async onModuleDestroy() {
    await this.$disconnect();
  }
}
```

Esse trecho de código representa a **classe responsável pela conexão com o banco de dados** na aplicação NestJS utilizando o **Prisma ORM**.

O PrismaService herda o PrismaClient e gerencia todo o ciclo de vida da conexão:

- Ao iniciar a aplicação, o método onModuleInit() é executado, chamando this.\$connect(), que **abre a conexão com o banco**.
- Se a conexão for bem-sucedida, uma mensagem é exibida informando que o banco foi conectado com sucesso.
- Caso haja erro, ele é tratado, exibindo mensagens de alerta para ajudar na identificação do problema.
- Quando a aplicação é finalizada, o método onModuleDestroy() chama this.\$disconnect(), **fechando a conexão corretamente**.
- Em resumo:
Esse código é exatamente a camada que estabelece, monitora e encerra a conexão entre o Prisma e o banco de dados usado pela aplicação junto com a .env

2. Classes DAO com seus métodos de CRUD

Adotante:

```
import { ApiProperty } from '@nestjs/swagger';

export class AdotanteDTO {
  @ApiProperty({ required: false, description: 'ID do adotante (gerado automaticamente)' })
  id?: number;

  @ApiProperty({ description: 'Nome do adotante' })
  nome: string;

  @ApiProperty({ required: false, description: 'Endereço do adotante' })
  endereco?: string;

  @ApiProperty({ required: false, description: 'Telefone do adotante' })
  telefone?: string;

  @ApiProperty({ required: false, description: 'CEP do adotante' })
  cep?: string;

  @ApiProperty({ description: 'Email do adotante' })
  email: string;
}
```

Animal:

```
import { ApiProperty } from '@nestjs/swagger';

export class AnimalDto {
  @ApiProperty({ required: false, description: 'ID do animal (gerado automaticamente)' })
  id?: number;

  @ApiProperty({ description: 'Nome do animal' })
  nome: string;

  @ApiProperty({ required: false, description: 'Idade do animal' })
  idade?: number;

  @ApiProperty({ required: false, description: 'Tipo do animal (ex: cachorro, gato)' })
  tipo?: string;

  @ApiProperty({ required: false, description: 'Sexo do animal' })
  sexo?: string;

  @ApiProperty({ required: false, description: 'Data de resgate do animal' })
  data_resgate?: Date;

  @ApiProperty({ required: false, description: 'Canil onde o animal está' })
  canil?: string;
}
```

Voluntario:

```
import { ApiProperty } from '@nestjs/swagger';

export class VoluntarioDto {
  @ApiProperty({ required: false, description: 'ID do voluntário (gerado automaticamente)' })
  id?: number;

  @ApiProperty({ description: 'Nome do voluntário' })
  nome: string;

  @ApiProperty({ required: false, description: 'Telefone do voluntário' })
  telefone?: string;

  @ApiProperty({ description: 'Senha do voluntário' })
  senha: string;

  @ApiProperty({ description: 'Nome de usuário do voluntário' })
  usuario: string;
}
```

3. Classe Main.ts

```
import { NestFactory } from '@nestjs/core';
import { AppModule } from './app.module';
import { DocumentBuilder, SwaggerModule } from '@nestjs/swagger';

async function bootstrap() {
  const app = await NestFactory.create(AppModule);

  const corsOrigins = process.env.WEB_APP_URL
    ? process.env.WEB_APP_URL.split(',').map((origin) => origin.trim()).filter(Boolean)
    : ['http://localhost:5173'];

  app.enableCors({
    origin: corsOrigins,
    methods: [ 'GET', 'POST', 'PUT', 'PATCH', 'DELETE', 'OPTIONS' ],
    allowedHeaders: [ 'Content-Type', 'Authorization' ],
    credentials: true,
  });

  // Configuração do Swagger
  const config = new DocumentBuilder()
    .setTitle('APT Sô Patinhas de Rua')
    .setDescription(
      'API para gerenciar animais, adotantes e voluntários do projeto Sô Patinhas de Rua. ' +
      'Esta API permite realizar operações CRUD completas para cada entidade.'
    )
    .setVersion('1.0')
    .addTag('animal', 'Operações relacionadas aos animais')
    .addTag('adotante', 'Operações relacionadas aos adotantes')
    .addTag('voluntario', 'Operações relacionadas aos voluntários')
    .build();

  const document = SwaggerModule.createDocument(app, config);
  SwaggerModule.setup('api', app, document);

  await app.listen(process.env.PORT ?? 3000);
  console.log(`Aplicação rodando em: http://localhost:${process.env.PORT ?? 3000}`);
  console.log(`Documentação Swagger: http://localhost:${process.env.PORT ?? 3000}/api`);
}

bootstrap();
```

Esse código representa o arquivo principal da aplicação NestJS. Ele é responsável por iniciar o servidor, configurar permissões de acesso e ativar a documentação da API.

A função `bootstrap()` é executada ao iniciar o projeto. Ela cria a aplicação usando o `AppModule`, que é o módulo principal que reúne todos os outros módulos.

Em seguida, o código configura o CORS, que controla quais endereços externos podem acessar a API. Ele verifica se existe uma variável de ambiente com URLs permitidas; caso não exista, usa um endereço padrão. Também define quais métodos HTTP e cabeçalhos são permitidos.

Depois disso, o código configura o Swagger, que é a ferramenta responsável por gerar a documentação da API. Nessa configuração são definidos o título, a descrição, a versão e as tags, que agrupam os endpoints por categoria. Em seguida, o Swagger é habilitado na rota /api.

Por fim, o servidor é iniciado na porta definida no arquivo .env ou, caso essa porta não exista, na porta 3000. O console exibe mensagens informando onde a API está rodando e onde a documentação pode ser acessada.

Em resumo, esse arquivo é o ponto de entrada da aplicação NestJS. Ele inicia o servidor, configura regras de acesso e prepara a documentação da API.

Descrição de testes

1. Apresentação de um exemplo de CRUD de uma entidade Animal

O teste realizado corresponde ao endpoint de criação de um novo registro de Animal no sistema, conforme apresentado na interface gerada pelo Swagger.

POST /animal

Esse endpoint é responsável por cadastrar um novo animal no banco de dados. A documentação exibida no Swagger confirma

que não há parâmetros na URL, e a requisição exige um **Request Body** no formato JSON.

Estrutura do Corpo da Requisição (Request Body)

O Swagger apresenta o seguinte modelo de dados para criação de um animal:

```
{  
  "id": 0,  
  "nome": "string",  
  "idade": 0,  
  "tipo": "string",  
  "sexo": "string",  
  "data_resgate": "2025-11-18T12:57:26.352Z",  
  "canil": "string" }
```

Descrição dos campos:

- **id**: número inteiro (o backend ignora esse valor ao criar um novo animal)
- **nome**: nome do animal
- **idade**: idade aproximada
- **tipo**: espécie ou categoria (ex.: cachorro, gato)
- **sexo**: macho ou fêmea
- **data_resgate**: data em que o animal foi resgatado
- **canil**: canil de destino ou abrigo temporário

Resposta Esperada

Segundo o Swagger, a resposta do servidor em caso de sucesso é:

201 – Animal criado com sucesso

Essa resposta indica que o animal foi salvo no banco de dados e que o servidor reconhece a operação como bem-sucedida.

```
HTTP/1.1 201 Created
X-Powered-By: Express
Vary: Origin
Access-Control-Allow-Credentials: true
Content-Type: application/json; charset=utf-8
Content-Length: 124
ETag: W/"7c-Hqh+yxtgp1DiQFF0YfY73b4GA2I"
Date: Tue, 18 Nov 2025 12:55:54 GMT
Connection: close

{
  "id": 9,
  "nome": "Rex",
  "idade": 4,
  "tipo": "Cachorro",
  "sexo": "Macho",
  "data_resgate": "2025-11-03T00:00:00.000Z",
  "canil": "Canil 1"
}
```

Atualização (PUT) da Entidade Animal

Este teste corresponde ao endpoint responsável por **atualizar os dados de um animal já cadastrado**. A execução foi realizada por meio da interface Swagger, conforme a imagem fornecida.

1. Endpoint Utilizado

PUT /animal/{id}

Esse endpoint realiza a atualização de um animal existente, utilizando como parâmetro o **ID** do animal que será modificado.

2. Parâmetro Obrigatório (Path Parameter)

O Swagger exibe um único parâmetro:

id (string) – obrigatório

Esse valor representa o ID do animal que se deseja atualizar.

Ele deve ser informado diretamente na URL (exemplo: /animal/3).

3. Estrutura do Corpo da Requisição (Request Body)

O Swagger apresenta o seguinte modelo de dados que deve ser enviado para atualizar o animal:

```
{  
    "id": 0,  
    "nome": "string",  
    "idade": 0,  
    "tipo": "string",  
    "sexo": "string",  
    "data_resgate": "2025-11-18T13:09:10.073Z",  
    "canil": "string"  
}
```

Descrição dos campos:

- **id**: número inteiro (representa o ID do animal, mas na prática é ignorado e o ID da URL é o que prevalece)
- **nome**: novo nome do animal

- **idade**: idade atualizada
- **tipo**: espécie ou categoria
- **sexo**: macho ou fêmea
- **data_resgate**: data de resgate, caso precise ser alterada
- **canil**: local onde o animal está alojado

Todos os campos devem ser enviados no corpo da requisição.

4. Resposta Esperada

O Swagger indica que, ao atualizar o animal com sucesso, a API retorna:

200 – Animal atualizado com sucesso

Esse retorno confirma que a atualização foi realizada corretamente no banco de dados.

```
HTTP/1.1 200 OK
X-Powered-By: Express
Vary: Origin
Access-Control-Allow-Credentials: true
Content-Type: application/json; charset=utf-8
Content-Length: 127
ETag: W/"7f-L33CsvNnYEdnhZ4z4jeBVUnMiCg"
Date: Tue, 18 Nov 2025 13:14:23 GMT
Connection: close

{
  "id": 9,
  "nome": "Totó",
  "idade": 10,
  "tipo": "Cachorro",
  "sexo": "Macho",
  "data_resgate": "2025-11-06T00:00:00.000Z",
  "canil": "Canil 1"
}
```

Listagem (GET) da Entidade Animal

O teste refere-se ao endpoint responsável por **listar todos os animais cadastrados** no sistema. A execução do teste foi realizada por meio da interface do Swagger, conforme exibido na imagem fornecida.

1. Endpoint Utilizado

GET /animal

Este endpoint retorna uma lista contendo todos os registros da entidade Animal presentes no banco de dados.

2. Requisitos e Parâmetros

O endpoint **não exige parâmetros** na URL, tampouco corpo de requisição. O usuário apenas aciona a rota e aguarda o retorno.

3. Exemplo de Execução (Curl)

o Swagger apresenta o seguinte comando Curl correspondente à requisição:

```
curl -X 'GET' \
'http://localhost:3000/animal' \
-H 'accept: */*'
```

Esse comando confirma que a requisição consiste em uma chamada HTTP GET simples.

4. URL da Requisição

<http://localhost:3000/animal>

Essa é a rota acessada pelo cliente durante o teste.

5. Corpo da Resposta (Response Body)

O servidor retornou status **200** acompanhado da lista completa de animais cadastrados, no formato JSON.

O Swagger exibiu o seguinte conjunto de dados (baseado na imagem fornecida):

```
HTTP/1.1 200 OK
X-Powered-By: Express
Vary: Origin
Access-Control-Allow-Credentials: true
Content-Type: application/json; charset=utf-8
Content-Length: 511
ETag: W/"1ff-HYjJp+5exvukZpN/t3xvTEh+OUw"
Date: Tue, 18 Nov 2025 13:15:34 GMT
Connection: close

>[
  {
    "id": 6,
    "nome": "Rex",
    "idade": 4,
    "tipo": "Animal Silvestre",
    "sexo": "Macho",
    "data_resgate": "2025-11-03T00:00:00.000Z",
    "canil": "Selva Amazônica"
  },
  {
    "id": 7,
    "nome": "rex",
    "idade": 15,
    "tipo": "cachorro",
    "sexo": "masc",
    "data_resgate": "2025-11-10T13:47:21.240Z",
    "canil": "02"
  }
]
```

Este retorno mostra que o endpoint está funcionando corretamente, exibindo múltiplos registros já cadastrados.

Exclusão (DELETE) da Entidade Animal

O teste refere-se ao endpoint responsável por excluir permanentemente um animal existente do banco de dados. A execução foi realizada através da interface do Swagger.

```
HTTP/1.1 200 OK
X-Powered-By: Express
Vary: Origin
Access-Control-Allow-Credentials: true
Content-Type: application/json; charset=utf-8
Content-Length: 119
ETag: W/"77-/ETB3EVdnoyWFDVTnLHBNyuu1MUs"
Date: Tue, 18 Nov 2025 13:20:29 GMT
Connection: close

{
  "id": 7,
  "nome": "rex",
  "idade": 15,
  "tipo": "cachorro",
  "sexo": "masc",
  "data_resgate": "2025-11-10T13:47:21.240Z",
  "canil": "02"
}
```

1. Endpoint Utilizado

DELETE /animal/{id}

Esse endpoint recebe o **ID** do animal que deve ser removido e efetua a exclusão do registro correspondente no banco de dados.

2. Parâmetro Obrigatório (Path Parameter)

O endpoint exige apenas um parâmetro:

id (string) – obrigatório

Esse valor deve ser informado diretamente na URL da requisição.

Exemplo: /animal/8

Nesse caso, o animal com ID 8 será removido.

3. Corpo da Requisição

O método **DELETE** não necessita de corpo de requisição.

Basta informar o ID na URL e executar a operação.