

DOCUMENTAÇÃO DE REQUISITOS

TrackPet: Animais perdidos

IDENTIFICAÇÃO DOS COMPONENTES DO GRUPO

Eduarda Vieira Netto, Ana Luísa Burjack, André Lopes, João Victor Rocha.

PROBLEMA

Atualmente há uma grande dificuldade em divulgar animais desaparecidos, para quem perdeu ou para quem encontrou. Os mais usados são redes sociais e grupos em apps de conversação, que acabam não sendo muito eficazes devido os seguintes fatores, por exemplo:

- Mecanismos de pesquisa deficientes;
- Informações descentralizadas (são diversos grupos em diversas redes sociais)

OBJETIVO

O intuito do sistema é criar um meio para conectar pessoas que perderam seus animais, com pessoas que encontraram animais perdidos.

IDENTIFICAÇÃO DOS COMPONENTES DO SISTEMA

Não há legislação brasileira, goiana ou goianiense que possa interferir na construção desse software. Entretanto, há aplicativos que já realizam serviços parecidos, como o “Procura-se Cachorro”, disponível em iOS.

Serão consumidos do smartphone os recursos: câmera e galeria.

Para esse sistema funcionar corretamente, é necessário o uso de um smartphone com sistema operacional Android numa versão igual ou superior ao Android 23.

IDENTIFICAÇÃO DOS USUÁRIOS FINAIS

Os usuários finais serão os habitantes da área metropolitana de Goiânia. Não há especificação acerca do sexo, raça ou religião dos usuários finais, entretanto, nosso público alvo são pessoas acima de 18 anos.

FONTES DE REQUISITOS

Utilizamos da técnica do brainstorm e adaptação das ferramentas providas pelo Design Thinking. Decidimos por essas ferramentas baseado na familiaridade do grupo com o problema apresentado.

REQUISITOS FUNCIONAIS

<Lembrar que o requisito funcional deve ser sempre capaz de ser testado.>

RF01

Como **<USUÁRIO>**, desejo realizar **<CADASTRO DO MEU TELEFONE>** para acessar o aplicativo e informações de contato de outros usuários.

Os dados obrigatórios para cadastro são: Nome completo, data de nascimento, número de telefone, foto e email.

RF02

Como **<DONO DE UM ANIMAL>**, desejo **<CADASTRAR OS DADOS DO MEU ANIMAL PERDIDO>**, para que **<ALGUÉM POSSA RECONHECÊ-LO>**.

Os dados obrigatórios são: foto, espécie, nome, sexo do animal, dia que desapareceu, onde desapareceu e informações de contato.

RF03

Como **<USUÁRIO>**, desejo **<CADASTRAR OS DADOS DO ANIMAL QUE EU ENCONTREI NO SISTEMA>**, para que **<ALGUÉM POSSA RECONHECÊ-LO>**.

Os dados obrigatórios são: foto, sexo do animal, dia que foi encontrado, onde foi encontrado e informações de contato.

RF04

Como **<DONO DE UM ANIMAL>**, desejo **<FILTRAR POR ESPÉCIE, SEXO E REGIÃO>**, para tornar **<A BUSCA POR ANIMAIS MAIS FÁCIL>**.

RF05

Como **<DONO DE ANIMAL PERDIDO>**, desejo que **<TENHA UMA ABA PARA ANIMAIS ENCONTRADOS>**, para tornar a busca filtrada apenas para animais que foram encontrados por outros usuários.

RF06

Como **<PESSOA QUE ENCONTROU O ANIMAL>**, desejo que **<TENHA UMA ABA PARA ANIMAIS PERDIDOS>**, para tornar a busca filtrada apenas para animais que foram encontrados por outros usuários.

RF07

Como <USUÁRIO>, desejo <TER ACESSO AO MEU PERFIL> com meus dados pessoais e foto

RF08

Como <USUÁRIO> desejo <EDITAR MEUS DADOS PESSOAIS>

RF09

Como <USUÁRIO>, desejo <PODER VISUALIZAR TODAS AS MINHAS POSTAGENS ORGANIZADAS POR ORDEM DECRESCENTE DE DATA DE PUBLICAÇÃO>

RF10

Como <USUÁRIO>, desejo <INFORMAR QUE O ANIMAL JÁ ESTÁ NOVAMENTE COM A FAMÍLIA>

O usuário que postou o anúncio (seja um anúncio de animal perdido ou encontrado) deve utilizar o botão “Família Reunida” para informar que o animal já retornou ao seu lar.

RF11

O sistema deverá <VINCULAR CADASTROS DE ANIMAIS ÀS CONTAS DE USUÁRIO> pois um usuário pode possuir vários cadastros de animais perdidos/encontrados.

RF12

Como <PESSOA QUE ABRIU O ANÚNCIO DE ANIMAL PERDIDO/ENCONTRADO>, desejo <EDITAR AS INFORMAÇÕES DO ANÚNCIO>, para que possa <MANTER OS DADOS ATUALIZADOS> caso tenha sido encontrado um erro ou houve alguma mudança.

REQUISITOS NÃO-FUNCIONAIS

<Não é uma funcionalidade, mas uma restrição sobre uma funcionalidade do sistema proposto, mas que também precisa ser capaz de ser testado.>

RNF<número único do requisito não-funcional>

Como <ator/(sub)sistema>, desejo <ação/objetivo/meta>, para que <razão/benefício>.

OU

O sistema deverá <tipo de restrição sobre algum requisito funcional>.

<Requisitos não-funcionais podem ser de vários tipos, então pesquise por requisitos de éticos, legais, de segurança, de implementação, de privacidade, dentre outros.>

RNF01

O sistema deverá **<SER MOBILE>**, para que **<SEJA FÁCIL O ACESSO AO SISTEMA EM QUALQUER LOCAL>** devido ao fato de que muitos dos animais são encontrados na rua.

RNF02

O sistema deverá ter **<DESIGN RESPONSIVO DAS INTERFACES GRÁFICAS>** para que **<NÃO IMPORTANDO O TAMANHO DA TELA A INTERFACE PERMANEÇA COESA>**

RNF03

O sistema deverá ser **<COMPATÍVEL COM ANDROID>** para que **<UM MAIOR NÚMERO DE USUÁRIOS POSSA SER ALCANÇADO>** por ser **<ESSE É O SISTEMA OPERACIONAL MAIS UTILIZADO EM GOIÂNIA-GO>**

RNF04

O sistema deverá estar sempre **<CONECTADO A INTERNET>** para que funcione.

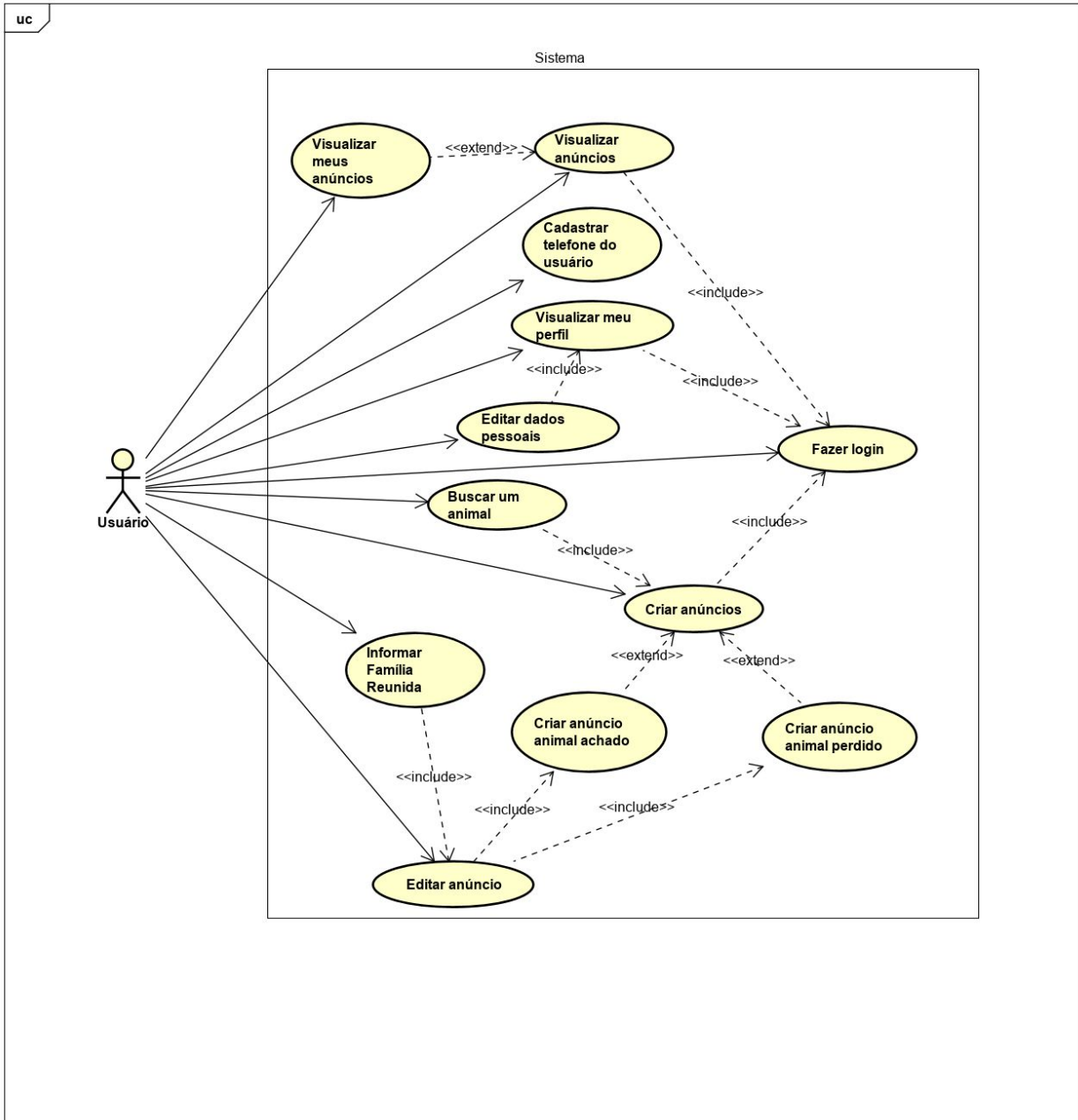
RNF05

O usuário deverá **<ESTAR LOGADO>** para acessar toda e qualquer função do sistema.

RNF06

O usuário deverá **<REALIZAR O LOGIN ATRAVÉS DA API DO GOOGLE>**, o cadastro da pessoa deve ser completado dentro do sistema, conforme RF01.

DIAGRAMA DE CASOS DE USO



Descrição de atores

Usuário do sistema: Temos um único tipo de usuário no sistema, mas podemos ter propósito de uso diferente, o mesmo usuário pode criar um anúncio de um animal perdido, ou criar um anúncio informando um animal achado.

CASOS DE USO DETALHADOS

*<Escolha os casos de uso que você considera mais relevantes de serem detalhados. Essa relevância pode ser determinada em função dos requisitos mais importantes **do ponto de vista dos usuários finais**. Mas, também pode representar requisitos que possuem grandes chances de mudarem ao longo do tempo de vida útil do sistema.>*

CASO DE USO UC01: CRIAR ANÚNCIO DE ANIMAL PERDIDO.

Ator primário: Usuário (Dono de Cão perdido)

Pré-Condições:

- O Usuário está cadastrado.
- O Usuário está logado.

Pós-Condições:

- O anúncio de animal perdido foi devidamente cadastrado no sistema.

Cenário Principal:

1. O usuário logado solicita a adição de anúncio ao sistema.
2. O usuário preenche os campos obrigatórios requeridos pelo sistema.
3. O sistema cadastra as informações e mostra o anúncio na tela de animais perdidos.

Fluxo Alternativo:

- 2a. O usuário deixa um dos campos obrigatório em branco.
 1. O sistema não deixa o cadastro do anúncio ser finalizado e transmite uma mensagem que indica a falta de um campo.

UC02: MARCAR UM ANÚNCIO COMO RESOLVIDO.

Ator primário: Usuário (Que criou o anúncio de animal perdido/encontrado)

Pré-Condições:

- O usuário tem um anúncio de animal perdido ou encontrado com o status “aberto”.

Pós-Condições:

- O status do anúncio de animal perdido/encontrado foi “resolvido” e o sistema não mostra mais esse anúncio nas abas principais.

Cenário Principal:

1. O usuário acessa o anúncio que deseja resolver.
2. O usuário usuário solicita a resolução do anúncio e confirma a mensagem de confirmação do sistema.
3. O sistema marca o anúncio como resolvido e não mostra mais o anúncio nas abas principais.

Fluxo Alternativo:

2a. Quando a mensagem de confirmação do sistema aparece, o usuário nega.

1. O sistema não atualiza o status do anúncio e aguarda o próximo comando do usuário.

UC03: EDITAR UM ANÚNCIO EM ABERTO.

Ator primário: Usuário (Que criou o anúncio de animal perdido/encontrado)

Pré-Condições:

- O usuário tem um anúncio de animal perdido ou encontrado com o status “aberto”.

Pós-Condições:

- As alterações do anúncio foram registradas no sistema e o anúncio foi exibido com as novas informações.

Cenário Principal:

1. O usuário acessa o anúncio que deseja alterar e solicita a alteração.
2. O usuário substitui as informações antigas pelas novas de acordo com a sua preferência e salva.
3. O sistema atualiza as novas informações no banco de dados e exibe o anúncio com as informações atuais.

Fluxo Alternativo:

2a. O usuário deixou um dos campos obrigatórios em branco.

1. O sistema não salva as alterações e transmite uma mensagem indicando o usuário a preencher os campos inválidos.
2. O usuário preenche corretamente e clica em salvar.

2b. O usuário não fez nenhuma alteração aos campos do anúncio.

1. O botão de salvar fica indisponível enquanto o usuário não fazer nenhuma alteração.
E o sistema aguarda alguma interação com o usuário.

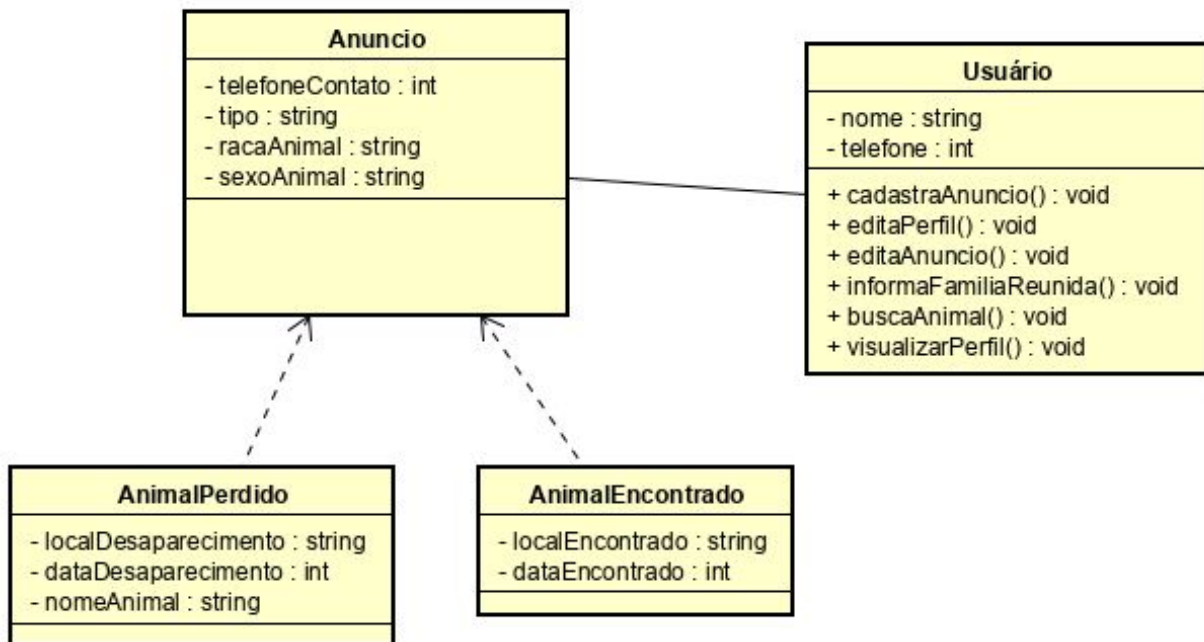
PROGRAMAÇÃO DAS ENTREGAS DOS REQUISITOS

Identificador	Assunto	Iteração	Fonte do Requisito	Última atualização
RF01	Integração da	1	<i>Não se aplica</i>	31/10/19

	API do Google para cadastro dos usuários			
RF11	Vincular usuário aos cadastros de animais	1	<i>Não se aplica</i>	31/10/19
RF02	Cadastrar dados de um animal perdido	1	<i>Não se aplica</i>	31/10/19
RF03	Cadastrar dados de um animal encontrado	1	<i>Não se aplica</i>	31/10/19
RF10	Botão 'Família Reunida' para animais perdidos entregue aos donos	2	<i>Não se aplica</i>	01/10/19
RF05	Aba para animais encontrados	2	<i>Não se aplica</i>	01/10/19
RF06	Aba para animais perdidos	2	<i>Não se aplica</i>	01/10/19
RF04	Filtro de animais por sexo, espécies e região	3	<i>Não se aplica</i>	01/10/19
RF07	Visualização do perfil do usuário	4	<i>Não se aplica</i>	01/10/19
RF08	Edição de dados do cadastro do usuário	4	<i>Não se aplica</i>	01/10/19
RF09	Usuário poder visualizar suas publicações	5	<i>Não se aplica</i>	01/10/19

Design

DIAGRAMA DE CLASSE



INTERFACES

UC01: CRIAR ANÚNCIO DE ANIMAL PERDIDO:

<https://www.figma.com/file/rHRJFO1jHop3TlqjBdmQy5/UC01-CRIAR-AN%C3%9ANCIO-DE-ANIMAL-PERDIDO?node-id=0%3A1>

UC02: MARCAR UM ANÚNCIO COMO RESOLVIDO:

<https://www.figma.com/file/iq4rUteCFzwTaxnWyA1yWQ/UC02-MARCAR-UM-AN%C3%9ANCIO-COMO-RESOLVIDO?node-id=0%3A1>

UC03: EDITAR UM ANÚNCIO EM ABERTO:

<https://www.figma.com/file/c8hefd0xRXibxWafH31mTx/UC03-EDITAR-UM-AN%C3%9ANCIO-EM-ABERTO>