

**Documentação de um**

**Produto de Software**

**Ache seu pet**

**Victor Arruda dos Santos: 82012896**

**Gustavo Matheus Pedro: 820110579**

**Carlos Eduardo de Melo: 820129752**

**Vinicius Lourenco da Silva: 820139864**

**Lucas de Castro Nolli: 820146333**

**2021**

**ÍNDICE DETALHADO**

**1.** **Introdução 3**

1.1. Tema 3

1.2. Objetivos a serem alcançados 3

1.3. Escopo principal 3

**2.** **Definição do Modelo de Processo 4**

**3.** **Requisitos do Sistema de Software 5**

3.1. Requisitos Funcionais 5

3.2. Requisitos Não-Funcionais 5

3.3 Casos de Uso 6

**4.** **Projeto 7**

4.1. Arquitetura Lógica 7

4.2. Arquitetura Física 9

**5.** **Protótipo de Interface 10**

**6.** **Critérios de Qualidade de Software 11**

**7. Testes 12**

7.1. Plano de Testes 12

7.2. Roteiro de Testes 16

**Anexo I 18**

1. **Introdução**
   1. **Tema**

Aplicativo para procurar uma possível última localização do seu pet, a partir da ajuda do compartilhamento de pets encontrados na rua por outros usuários.

* 1. **Objetivos a serem alcançados**

O aplicativo será desenvolvido para mobile(desenvolvido no android studio) exclusivamente para usuários de android.

Conseguir desenvolver uma inteligência artificial boa o suficiente para encontrar semelhanças entre as fotos do seu pet e o pet encontrado na rua que foi compartilhado por outro usuário.

Esse app deve ser fluido, contendo um mapa mundial, uma área para fazer um novo chamado de pet encontrado e uma parte para o usuário adicionar fotos do seu pet e procurar pets perdidos semelhantes ao dele.

O público alvo são os donos de pets.

Queremos receber muitos chamados de pet desaparecido para conseguir devolver o máximo de pets para o seu lar, conseguindo ajudar muitas pessoas, pois não encontramos nenhum aplicativo que consegue ajudar nessa tarefa de procura de pets, só conseguimos compartilhar em redes sociais até o momento. Então queremos ter um efeito bem positivo em relação a quantidade de pets encontrados após a estreia do aplicativo.

Ajudar na denúncia de animais abandonados.

* 1. **Escopo principal**

Encontrar pets por meio das fotos do pet cadastrado pelo dono e as fotos de pets encontrados na rua, utilizando

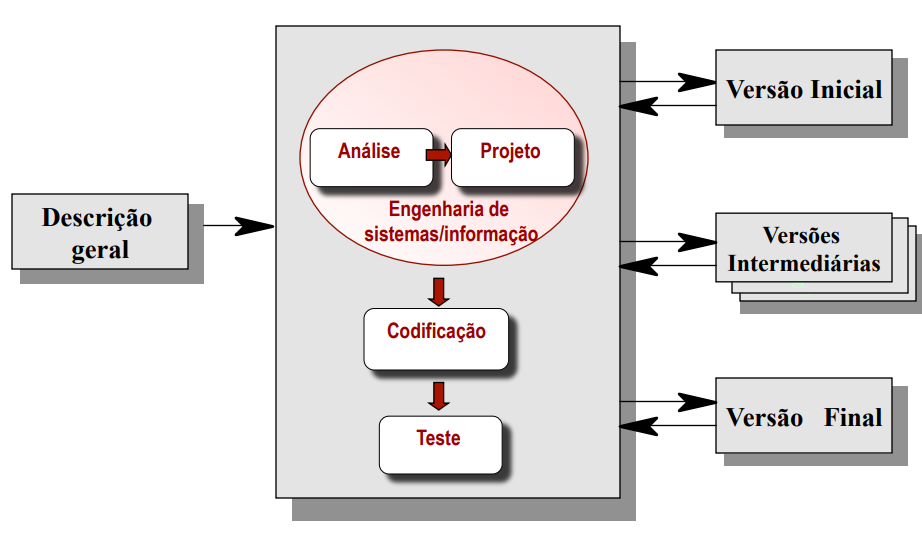
inteligência artificial para ver semelhanças nos pets reportados anteriormente, com as fotos do pet do

usuário com base na cor,

raça, tamanho, etc.

Ajudando na diminuição de pets abandonados nas ruas, e no aumento de adoções de animais carentes.

1. **Definição do Modelo de Processo**

Pensando em relação a dinâmica escolhida do nosso Software, o modelo que mais se encaixaria seria o modelo Incremental, onde é mais apropriado para sistemas ainda pequenos, onde as novas versões podem ser planejadas de modo que os riscos técnicos possam ser administrados.  
O objetivo desse modelo é trabalhar junto do usuário para descobrir seus requisitos, de maneira incremental, até que o produto final seja obtido.  


Nós da Ache seu Pet, escolhemos esse modelo justamente pelo fato de seguirmos uma ordem de criação mais apropriada. Começando pela Especificação do Software, seguindo da formulação do Software, a implementação do teste de Unidade, para validar os requisitos e fases do Projeto, a integração do Teste de Sistema, para testar o Software por completo, e por fim, Operar e passar o feedback do Software.

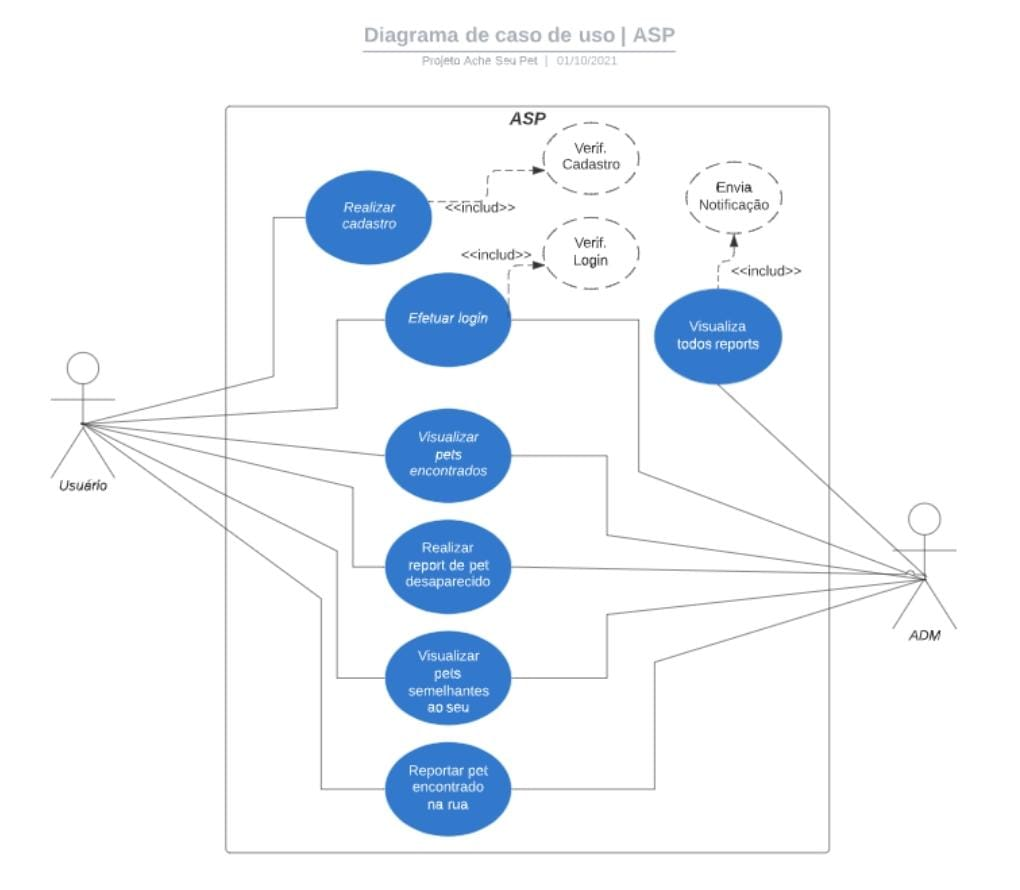
1. **Requisitos do Sistema de Software**
   1. **Requisitos Funcionais**

Levando em consideração os requisitos Funcionais de nosso Software, conseguimos aplicar tais funções:

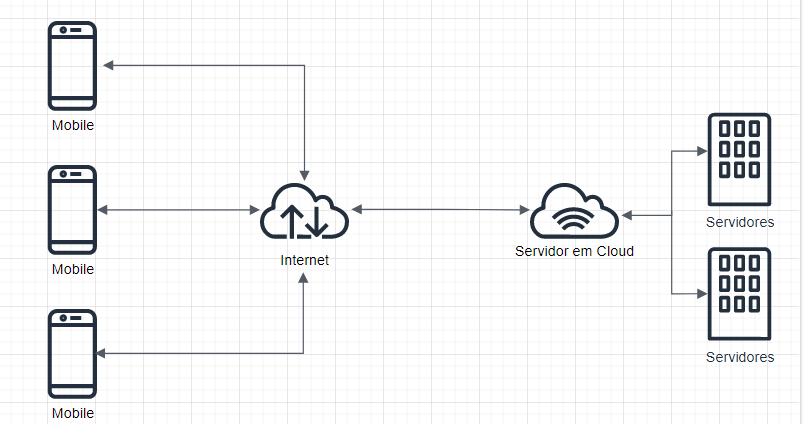
* F-1: Novos usuários terão de realizar a inserção de dados no formulário de cadastro, tanto do seu perfil quanto do seu pet;
* F-2: Usuários terão de realizar o login para utilizar o aplicativo;
* F-3: Usuários podem editar as informações do seu perfil ou do seu pet;
* F-4: Usuários podem cadastrar pets encontrados na rua;
* F-5: Usuários podem realizar a varredura da área, para comparar o pet que perdeu com outros pets achados e cadastrados por outros usuários na região atual;
* F-6: Usuários podem consultar a lista de pets cadastrados que estão atualmente perdidos;
* F-7: Usuários podem abrir o chat para conversar com outros usuários que cadastraram um pet perdido;
* F-8: Usuários podem editar as informações dos pets cadastros por ele, seja adicionando ou removendo informações;
* F-9: Usuários podem realizar o Logout do aplicativo;
  1. **Requisitos Não-Funcionais**

Agora, voltando para os requisitos Não Funcionais, podemos apresentar o Desempenho (Fazendo com que a Infraestrutura funcione sem problemas de lentidão e nada que possa comprometer a qualidade do uso do Sistema, ou seja, tendo uma infraestrutura limpa e leve); Podemos levar também em consideração a Segurança (Onde possuímos um sistema de armazenamento de senhas e usuários criptografados, impedindo o roubo de dados e garantindo a confiabilidade do Sistema, por exemplo, através do Banco de Dados); Como dito antes, temos a Interoperabilidade (Garantindo a integração com outras aplicações, comunicando de forma transparente com outro sistema.); Usabilidade Pode ser considerado um forte, visto que seria voltado a fácil compreensão do Software para os diferentes públicos envolvidos e possibilitando a fácil utilização de nosso Software; E principalmente voltado ao software ser Legal(Seguindo os padrões LGPD — Lei Geral de Proteção de Dados).

* 1. **Casos de Uso**



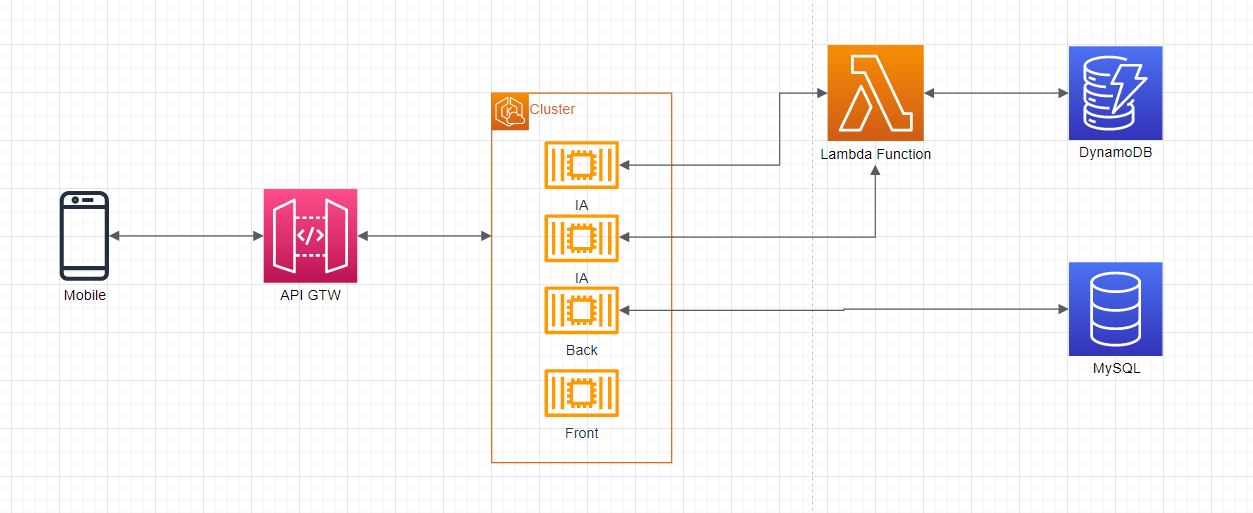
1. **Projeto**
   1. **Arquitetura Lógica**

A arquitetura Cliente-Servidor:  


* Um dos principais objetivos de escolhermos essa arquitetura é por conta da segurança, pois com a Cliente-Servidor corremos menor perigo de roubo de dados via internet, apenas nossas aplicações dentro da VPC em Cloud tem acesso aos dados nos bancos e distribui apenas o necessário para os usuários, mantendo assim um fluxo controlado sobre o que entra e sai.
* Utilizamos React Native (JavaScript) como linguagem de programação para a interface no android. Ficamos sabendo que essa biblioteca do JavaScript criada pelo Facebook tem uma grande presença em aplicativos famosos como Uber, Instagram, a Shopee e o próprio Facebook, por isso, escolhemos essa biblioteca como referência.
* Utilizamos Python para as inteligências artificiais e para as funções lambdas que fazem o reconhecimento dos pets nas fotos. O Python é bastante utilizado em funções lambdas na AWS, fora que também é uma linguagem de programação de alto nível, podendo realizar tarefas complexas como análise e comparação de dados, por isso é bastante utilizado em Machine Learning.
* Para banco de dados não relacional, utilizamos o DynamoDB da AWS, para guardar fotos e dados que irão abastecer as funções lambdas e transportar para os pods de IAs.  
  Escolhemos o DynamoDB por ser um banco não relacional altamente escalável e por sua capacidade e potência que são gerenciadas pela AWS, fazendo com que o banco nunca fique em seu limite.
* Para banco de dados relacionais, utilizamos o Aurora MySQL também na AWS, para armazenar os cadastros e informações pessoais dos usuários.  
  O Aurora é um banco oferecido pela AWS que tem alta disponibilidade, alta elasticidade e um auto escalonamento, por isso nossos técnicos não precisam se preocupar em aumentar sua potência ou seu armazenamento.
  1. **Arquitetura Física**

A nossa Arquitetura física está hospedada na Nuvem AWS, sendo assim temos um API gateway realizando as chamadas para o ambientes em um Cluster de EKS (Elastic Kubernetes Service). Dentro do Cluster realizamos a parte da chamada do Front feito em React Native, do Backend em JavaScript servido pelo MySQL e de outros serviços com IAs sendo abastecidas por Funções Lambda (Ambientes Serverless que a AWS disponibiliza) desenvolvidas em Python, sendo abastecidos por informações no DynamoDB.  
  
Um ponto de escolhermos a Cloud é porque podemos utilizar vários recursos como o Lambda e o EKS, que são recursos altamente escaláveis, e com isso o aplicativo ficará altamente disponível. Outro ponto é por conta do sistema tolerante a falhas, com isso o aplicativo nunca ficará fora do ar, caso aconteça algo com os servidores. E por fim, centralizar os dados em um único local e de fácil acesso e manutenção, com isso nossas equipes podem localizar e cuidar rapidamente do erro.

A ferramenta de microsserviços, que é o EKS, nos permite escalonar horizontalmente e verticalmente a quantidade de pods necessárias para suportar a aplicação, com isso o hardware de cada Cluster pode ser de 64 vcpu para 32 de RAM, caso necessário, podemos aumentar quando quiser o poder computacional dos nossos clusters assim dobrando sua capacidade.  
  
Na parte de Redes do ambiente, fizemos com que apenas o frontend tenha acesso externo através das configurações de firewall da AWS, com isso privamos nosso backend, IAs e Bancos apenas para quem tem acesso via VPN, todos os ips cadastrados para alteração interna no Cluster e Bancos são gerenciados pela equipe de Segurança, onde apenas os mesmos podem alterar a regra de ips de entrada ao ambiente.  
  
A estrutura seria feita com os procedimentos baseados no Well-Architect Framework, documento disponibilizado pela Amazon para realizar as melhores práticas de arquitetura e segurança para Nuvem, como a alta disponibilidade para clientes no Brasil e tolerância a falhas com o ambiente implementado em duas ou mais Zonas de disponibilidades.  
  
Estamos pensando futuramente em deixar o ambiente disponível mundialmente, implementando o seguinte ambiente em várias regiões do Mundo.



1. **Protótipo de Interface**

Para fazer a prototipagem deste projeto utilizamos o Adobe XD, segue o link do vídeo mostrando o protótipo por completo com uma breve explicação de cada área que o software possui e sua funções:

https://drive.google.com/file/d/1zeiypdQGdzSoIfYnktwifdxkpuYyPo9I/view?usp=drivesdk

1. **Critérios de Qualidade de Software**

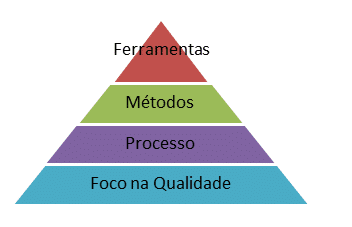
Em nosso projeto serão utilizados os os principais requisitos de qualidade de software que por usa vez são:

Usabilidade, onde implementamos essa característica de qualidade para manter o software bem organizado e de fácil manuseio e entendimento, tendo assim custos reduzidos e mantendo a aprovação do usuário.

Confiabilidade, é completamente necessário que o software por sua vez tenha o mínimo de chances de falhas no sistemas, por isso é necessário o foco principal na fase de desenvolvimento do software para que na fase operacional ocorra com o mínimo de erros possíveis.

Funcionalidade, outro fator que implementamos que por sua vez garantimos que o software atenda todas as especificações propostas pelo usuário, garantindo seu desempenho, dentro das condições estabelecidas.

Manutenibilidade, outro fator primordial que adicionaremos em nosso software é o que se refere a facilidade de modificação do software a fim de corrigir defeitos em curto prazo ou adicionar novos requisitos, atualizações frequentes e afins, aumentar a suportabilidade e adquirir novos ambientes em nosso software.

****

**7. Testes**

7.1. Plano de Testes

O público alvo do nosso app são donos(as) de pets, como cachorros e gatos.

O objetivo do projeto visa em criar um app para encontrar o seu o seu pet que fugiu.

Abreviações usadas: I.A = inteligência artificial.

Caso seguirmos para uma nova atualização ou feature do aplicativo iremos utilizar o seguinte plano de testes respectivamente (Realizando um loop até chegar ao resultado esperado):  
Os testes Estruturais -> os testes Não-Funcionais -> testes de Regressão -> os testes Estruturais…

Realizamos testes estruturais como o teste unitário e o teste de integração. Temos esses testes automatizados, também conhecidos como testes de caixa branca, onde realizamos ele diretamente no código. Segue os resultados:

* Teste unitário: O teste unitário é responsável por testar cada função e classe do nosso código, com isso podemos ver quais classes são independentes do resto do código, assim facilitando a vida do desenvolvedor para corrigir possíveis bugs existentes;
* Teste de integração: O teste de integração é responsável por testar a comunicação entre as partes dos sistemas e verificar se estão funcionando corretamente, com isso podemos realizar os testes de integração sempre que alguma parte do software não estiver realizando esta comunicação e verificar onde está localizado o problema;

Técnica:

* Execute os recursos, as integrações ou as funções de cada um dos cenários;
* Os resultados esperados ocorrerá quando forem usados dados válidos;
* As mensagens de erro ou de aviso apropriadas serão exibidas quando forem usados dados inválidos cada regra de negócio será aplicada de forma adequada;

Ferramenta:

* Ferramenta de Scripts de Teste Automatizada (Script desenvolvido internamente);
* Ferramentas de monitoramento (Zabbix + Prometheus + Grafana);
* Ferramenta de Backup do código atual (GitLab ou GitHub);

Realizamos também testes não funcionais como os de usabilidade, eficiência, confiabilidade e segurança foram realizados. Os testes são testes de caixa preta, onde não é aplicado diretamente no código. Segue os resultados abaixo:

* Usabilidade: o app é intuitivo, sua interface é de fácil interação, podendo assim ser de fácil entendimento para todas as pessoas.  
  Através de testes de usabilidade realizados, os resultados colhidos foram que:
* Sua interface é prática, com poucos passos para novos usuários realizarem as interações básicas do aplicativo, como cadastrar uma nova conta ou cadastrar um pet perdido;
* Os erros que podem ser cometidos por novos usuários são mínimos, pois a interface é explicativa por si própria, mas caso o usuário cometa um erro de digitação, por exemplo, a interface irá o alertar imediatamente;

Com isso nosso aplicativo é indicado para todos os tipos de pessoas com pets, pois desejamos a satisfação dos usuários.

Técnica:

* Testar as interações do aplicativo com o usuário através de erros cometido por falhas humanas, como erros de digitação ou erros de não preenchimentos de campos obrigatórios
* Testar a navegação do aplicativo com interpretação de toques à tela, tamanhos dos layouts dos menus, funcionalidade dos botões e etc…

Ferramenta:

* Ferramenta de Scripts de Teste Automatizada (Script desenvolvido internamente);
* Testes manuais com técnicos;

* Eficiência: O software tem respostas rápidas para usuários em território nacional por conta da alta disponibilidade, sendo assim o aplicativo foi preparado para receber altos números de requisições.  
  Os testes realizado são testes de carga de requisições:  
  + Os testes de estresse são feitos regularmente em ambientes de desenvolvimento e de homologação, caso aconteça do serviço cair, iremos realizar a monitoração e realizar ajustes necessários, seja na infraestrutura ou no software;
  + Testamos também a eficiência em celulares de hardwares mais fracos, queremos que nosso aplicativo seja utilizado por todos, por isso sua otimização é um ponto fundamental;

Técnica:

* + - Testar recursos limitados, os testes deverão ser executados nos pods dentro do cluster, e a memória RAM e o espaço de armazenamento persistente no servidor deverão ser reduzidos ou limitados.

Ferramentas:

* + - Ferramenta de Scripts de Teste Automatizada (Script desenvolvido internamente);
    - Ferramenta de controle e de Programação de Carga (JMeter);
    - Ferramentas de monitoramento (Zabbix + Prometheus + Grafana);
* Confiabilidade: A segurança dos dados dos nossos clientes é prioridade do nosso projeto, pois com isso ganhamos a confiança e credibilidade deles.  
  Scans de Vulnerabilidade e Testes de invasão são feitos periodicamente com isso obtemos este resultado:
* A Cloud oferece proteção contra ataques de negação de serviço e seu gerenciamento de firewall é bem completo;
* Já no aplicativo, realizamos scans periódicos para consertar bugs e possíveis brechas no sistema, também contamos com a presença de um programa de bug bounty, com isso recebemos relatos de usuários que exploram possíveis vulnerabilidades e reportam ao nosso time de desenvolvimento;
* Realizamos também a instalação de uma imagem de sistema operacional monitorada nos computadores e notebooks de nossas equipes, caso alguém esteja tentando a invasão a partir do mesmo, conseguimos realizar o bloqueio imediato à rede VPN da empresa.
* Sistemas de monitoramento estão sempre captando e monitorando a aplicação no ambiente de infraestrutura, com isso, conseguimos automatizar o reinício rápido dos serviços caso algum deles fique indisponível.
* Realizamos backups diariamente de nossos serviços, tanto em bancos de dados e aplicações, com isso, podemos facilmente recuperar dados perdidos em casos de falha. Realizamos também a implantação em outras zonas de disponibilidades, por conta disso, podemos recuperar dados vitais para o funcionamento do sistema.

Técnica (Proteção contra invasão):

* Criar testes para cada tipo de usuário e verifique cada permissão criando transações específicas para cada tipo de usuário;

Ferramentas:

* Ferramentas de investigação e contra a violação da segurança por “hackers”;
* Ferramenta desenvolvida internamente pela equipe de segurança para scan de vulnerabilidades;
* Ferramentas de monitoramento (Zabbix + Prometheus + Grafana);

Técnicas (Tolerância a Falhas)

* Interrupção da energia para o servidor;
* Vários campos, ponteiros e chaves de banco de dados deverão ser corrompidos manualmente e diretamente no banco de dados (através das ferramentas de banco de dados).

Ferramentas:

* Ferramentas de monitoramento e reinicialização (Zabbix + Prometheus + Grafana, Run Deck e Jenkins);
* Ferramentas de Auto Scaling (Service do EKS);
* Backups e recuperação de dados (Snapshots);

Também realizamos os testes de regressão, que podem ser tanto em caixa preta como caixa branca:

* Teste de regressão: Os testes de regressão são realizados após uma grande atualização no aplicativo, com isso, pode conter novos bugs e erros. Quando erros são acusados, após a manutenção do mesmo, será refeito todo o plano de testes, até chegar ao resultado desejado. Quando tudo pronto, é lançado a atualização para os usuários finais;

Técnica:

* Testar novas funcionalidades até receber erros ou perceber bugs;
* Caso receba erros, atualizar o código principal e realizar novamente todo o plano de teste;

Ferramentas:

* Ferramenta de Scripts de Teste Automatizada (Script desenvolvido internamente);

Para a métrica de teste, utilizamos uma ferramenta chamada SonarQube, com ela definimos as métricas necessárias e com elas medimos a qualidade do código atual. O código só é passado de desenvolvimento para homologação quando atinge a qualidade de 60% +, para a passagem para Pré-Produção a qualidade necessária é de 80%+;

**7.2. Roteiro de Testes**

O roteiro de testes começa com:

Cenário: Realizar Cadastro:  
Pré Condição: Aplicativo instalado;

* Tela Cadastro: usuário efetuando o cadastro, onde ele irá inserir suas informações pessoais e de seu pet, onde também irá adicionar 3 fotos de seu pet, após isso ele irá definir uma senha para sua conta.
* OK: Caso esteja tudo correto irá retornar algumas informações e a mensagem “Cadastro realizado” e o usuário terá acesso ao aplicativo;
* Not OK: Caso aconteça um erro por parte do cliente, como, um erro de digitação ou o não preenchimento de um campo obrigatório o sistema informará o usuário através de uma mensagem na sua tela informando o erro do usuário;

Cenário: Realizar Login  
Pré Condição: Ter Cadastro;

* Tela Login: Para fazer o login em uma conta já criada, o usuário irá abrir seu aplicativo e clicar no botão login, ele será enviado a uma tela que irá pedir para o usuário inserir seu e-mail e a sua senha.
* OK: Caso esteja correto, ele será conectado à sua conta, com a mensagem “Login Concluído” e levado para tela de início;
* Not OK: Caso o usuário não preencha algum campo necessário, uma mensagem aparecerá dizendo que é obrigatório o preenchimento de todos os campos. Se o usuário completar os campos e as informações não estiverem corretas, uma mensagem aparecerá “e-mail ou senha estão incorretas/e ou não tem cadastro”;

Cenário: Realizar Logout;  
Pré Condição: Estar Logado;

* Logout: O usuário poderá fazer um logout clicando no seu perfil localizado na parte superior à direita da tela principal do aplicativo simbolizada pela foto de perfil do seu pet.
* OK: Após clicar, um pop-up se abrirá com a opção de logout e editar perfil, ao clicar em desconectar, o usuário será desconectado da sua conta;

Cenário: Atualizar informações do seu perfil e do Pet;  
Pré Condição: Estar Logado;

* Tela Principal: O usuário poderá realizar atualizações do seu perfil e do seu pet clicando no ícone do seu perfil, localizado no canto superior direito e clicando em Editar perfil. Aparecerá uma nova tela com as opções para alteração, como, foto do perfil, nome, idade, endereço de email e senha;  
  + OK: Caso re-preencha as informações corretamente e clique em salvar, as informações serão alteradas e aparecerá uma mensagem “Perfil salvo com sucesso!”;
  + Not OK: Caso re-preencha as informações incorretamente e clique em salvar, o sistema alertará o usuário com a mensagem “Informações inválidas, por favor revisar”;

Cenário: Realizar varredura em busca do Pet;  
Pré Condição: Estar Logado;

- Tela Principal: Caso o pet do usuário fuja de sua casa, o seu dono irá clicar no botão procurar, localizado na parte central inferior, ao iniciar, a inteligência artificial fará uma busca através das informações cedidas pelo usuário das características do seu pet, e buscará por pets semelhantes ao seu.

* OK: Caso encontrado pets semelhantes, irá aparecer uma tela com as últimas localizações onde esses pets foram encontrados e cadastrados;
* Not OK: Caso nada fosse retornado, irá aparecer a mensagem “Nenhum pet semelhante foi encontrado”;

Cenário: Encontrar Pet no mapa virtual  
Pré Condição: Estar Logado;

O usuário poderá clicar nesse indicador no mapa e terá a opção de visualizar as informações, a possibilidade de registrar atualizações daquele pet específico, de se comunicar com a outra pessoa que abriu esse alerta através de um chat de mensagens e ir para o local indicado, onde o aplicativo irá se comunicar com o Google Maps e levá-lo até aquele ponto;

* OK: Caso o pet encontrado ainda esteja cadastrado, abrirá um pop-up informações do Pet encontrado, um botão para atualizar o informações do pet, a possibilidade de se comunicar via Chat com o usuário que cadastrou o mesmo e a opção de ir para o local indicado;
* Not OK: Caso o pet não esteja mais cadastrado, aparecerá uma mensagem “Pet removido ou já encontrado pelo dono”;

Cenário: Cadastro de Pet Encontrado  
Pré Condição: Estar Logado;

* Tela Principal: Em uma outra situação o usuário do software encontra um pet que possivelmente tenha se perdido ou fugiu de seu dono, o usuário poderá clicar no botão adicionar, essa ação fará abrir uma tela onde é possível adicionar uma foto do animal inserir a hora que ele foi visto, o endereço da rua e alguma observação.
* OK: Caso preencha todos os campos corretamente, o usuário clica em salvar, adicionando um indicador no mapa do app;
* Not OK: Se todos os campos obrigatórios não forem preenchidos uma mensagem aparecerá com o aviso que aquele campo é obrigatório;

**Anexo I**

**Victor Arruda do Santos (Entrevistador)  
Vinicius Lourenço da Silva (Entrevistador)  
Gustavo Matheus Pedro (Entrevistado)**

**Carlos Eduardo de Melo (Roteirista)  
Lucas de Castro Nolli (Roteirista)**

**Roteiro de Entrevista**

1. Fale sobre quais são as suas principais atividades na organização (o entrevistado falará sobre suas tarefas e responsabilidades) e/ou explique como é atualmente o processo do escopo principal.  
   R: Gustavo Matheus Pedro é o gestor da parte de desenvolvimento do Software ASP, ele está responsável por desenvolver o escopo geral do projeto e executar a criação do software.
2. Quais são as vantagens e desvantagens do processo atual na organização?

r: Vantagem: A segurança e o planejamento do projeto, com a liberdade de ideias dos desenvolvedores  
Desvantagem: O Tempo, pois o time tem muita demanda para pouco tempo, isso pode acabar pressionando os desenvolvedores e acabar limitando a criatividade por conta da pressão

1. Quem poderia usar este produto na organização? Justifique.  
   R: É de livre acesso ao público, porém a manutenção e o desenvolvimento do software o time de desenvolvimento.
2. Quais necessidades este produto atenderá? Justifique.  
   R: As necessidades estão ligadas aos donos de Pets, garantindo a segurança, eficiência e usabilidade do aplicativo para os usuários.
3. Quais os objetivos deste produto? Justifique  
   R: O objetivo do produto é ajudar os donos a encontrar seus pets perdidos nas redondezas, com base na localização encontrada por outro usuário do aplicativo, que registram animais perdidos nas ruas.
4. Como este produto adicionará valor para a empresa? Justifique.  
   R:A Empresa almeja a confiança dos clientes, de uma empresa que se importa com seus clientes e seus companheiros pets.
5. Quais atributos deste produto são críticos para atender às necessidades da empresa e portanto, para o sucesso do projeto? Justifique  
   R: O investimento na ferramenta de localização e na inteligência artificial do aplicativo, isso ajudará a encontrar cada vez mais Pets perdidos;  
   .
6. Como o produto proposto se compara aos produtos existentes dos concorrentes ou da mesma empresa? O que você sabe sobre produtos similares? Justifique.  
   R: É uma inovação no mercado, são pioneiros no mercado, através de pesquisas a empresa descobriu que não existia um software parecido.
7. Quais os critérios de qualidade para o produto (software) proposto? Justifique.  
   R: Estão voltados para a garantia e gestão do software, voltados para a usabilidade, segurança do usuário através da confiabilidade e eficiência junto com a durabilidade do software.
8. Quais sugestões você teria para melhorar o produto proposto? Justifique  
   .R: Poderiam expandir para WebApp e para sistemas IOS
9. Comentando sobre critérios de qualidade, como você classificaria a usabilidade do seu produto?

R: Está voltado para ser um software fácil de utilizar, estão voltados também para jovens e principalmente idosos que não tem costumes com a tecnologia

1. Com base nas respostas de nossas perguntas, o software de vocês é voltado para animais que já possuem dono, o que pensam sobre os animais de rua?

R: Num caso futuro, querem expandir para esse meio, não querem perder essa oportunidade. Nisto a empresa investiria em ONG para cuidar dos animais e colocariam no aplicativo para adoção

1. Como surgiu a ideia do software?  
   R: Através de reportagens de um aumento de perda de pets, o gerente de projetos teve uma ideia e a levou para o presidente, onde logo foi aprovado.
2. Você teria algo a acrescentar sobre o que conversamos?  
   Não.