

Perdigogs: Plataforma de Compartilhamento de Animais Perdidos

Ewelly Fabiane Cunha Costa Josemar Rocha da Silva Joshua Kook Ho Pereira Luigi Muller Sousa Linhares

Markus Kaul Monteiro Gerrits

Miller Raycell Monteiro Correia

Tarlison Sander Lima Brito

Abstract—No Brasil, existem aproximadamente 30 milhões de animais abandonados¹, dentro desse número são contabilizados animais que se perderam de seus donos, ou fugiram da sua residência. Para realizar a busca pelos animais, os donos utilizam redes sociais como forma de divulgar a busca pelo animal. A proposta do projeto apresentado é uma aplicativo multiplataforma para o compartilhamento dos animais.

I. INTRODUÇÃO

O projeto é um sistema multiplataforma de divulgação de animais, o sistema funciona com 3 partes principais, primeiramente um *bot* no *Telegram*, no qual é a forma de realizar a inserção dos animais perdidos, e também o responsável pela verificação da presença dos animais, em segundo existe um *bot* no *Twitter*, a principal plataforma de divulgação dos animais, e por fim um *site* com um mapa apresentando a presença dos animais.

II. DESCRIÇÃO DO PROJETO

A. Divisão da equipe

A equipe de desenvolvimento é composta por 7 pessoas, divididas em 3 cargos:

- Ewelly Fabiane Cunha Costa: *Maintainer*
- Josemar Rocha da Silva: *Reporter*
- Joshua Kook Ho Pereira: *Maintainer*
- Luigi Muller Sousa Linhares: *Developer*
- Markus Kaul Monteiro Gerrits: *Reporter*
- Miller Raycell Monteiro Correia: *Maintainer, developer*
- Tarlison Sander Lima Brito: *Developer*

B. Elicitação de requisitos

Com o levantamento de requisitos pode-se definir a qual a real necessidade do trabalho e quais as demandas a serem atendidas.

1) Requisitos Funcionais:

- O sistema deve ser capaz de fazer o cadastro da atividade de animais, tendo como informações a localização atual, a foto do animal.
- O sistema deve fazer a atualização do registro de animais perdidos na plataforma Twitter e no site próprio.
- O bot do Telegram deve ser capaz de fazer o cadastro dos animais desaparecidos.

- O mapa desenvolvido para o sistema deve ser sempre o mais atualizado de acordo com todas as novas inserções de animais.
- O perfil do Twitter deve sempre estar o mais atualizado possível de acordo com as inserções do banco de dados.
- Todos os animais cadastrados no sistema devem ser atualizados após 10 dias, realizando a exclusão do banco, ou serem atualizados após 10 votos apresentando a não presença do animal.

2) Requisitos Não Funcionais:

- Todos os containers da aplicação devem estar sempre funcionando.
- O bot do Telegram deve ter um tempo de resposta de no máximo 5 segundos.

C. Descrição Desenvolvimento

Para a construção do Perdigogs, foram utilizadas as linguagens de programação: *Python*, *TypeScript*, também foi utilizado no projeto, *Docker*, *Docker Compose*. A estrutura da implementação é dividida em 4 partes.

1) *Controle Bots*: Para a implementação dos *bots* foi utilizada a linguagem *Python*, para a criação do *bot* do *Telegram* foi utilizada a biblioteca *python-telegram-bot*, o *bot* realiza:

- Cadastro dos animais perdidos no banco.
- Função de análise da presença de animais próximos.

Para o *bot* do *Twitter*, a sua função se resume em pegar a informação processada pelo *bot* do *Telegram*, e realizar um post na conta do Perdigogs no *Twitter*, contendo no máximo 4 fotos e a localização na qual o animal foi encontrado.

Toda a estrutura é contida dentro de um container *Docker*.

2) *Backend*: Para a alimentação do *site* do Perdigogs, foi criado um *backend* na linguagem *Python*, utilizando *Flask*, no qual a função era realizar a requisição para o banco, tendo como parâmetro a latitude, longitude e raio de busca, e retornar esses dados para o *Frontend*. Toda a estrutura está contida dentro de um container *Docker*.

3) *Frontend*: A exibição dos dados de animais perdidos próximos ao local de busca, em um mapa, foi realizado utilizando a linguagem de programação *TypeScript* e o *Framework React*, a biblioteca do mapa de exibição é oferecida pela biblioteca *Mapbox*. Toda a estrutura é contida dentro de um container *Docker*.

¹<https://anda.jusbrasil.com.br/noticias/100681698/brasil-tem-30-milhoes-de-animais-abandonados>

4) *Banco de Dados*: Para a solução foi escolhido o banco *NoSQL MongoDB*, na plataforma *Atlas*, que é a solução em nuvem do *MongoDB*, essa escolha foi baseada no conceito de manutenção de containers, utilizar o banco na nuvem, simplifica alguns passos da manutenção do projeto, como preocupação com persistência de dados, possibilidade de quedas.