



PUC Minas

Blog Dog
Documentação do Website

André Fellipe Carvalho Silveira
Samuel Spineli Rodrigues
Sarah Luiza de Souza Magalhães

1- Front-end, Back-end e Banco de dados	3
1.1 Ferramentas utilizadas	3
1.2 Funcionamento	3
1.3 Links	3
2- Sistemas Inteligentes	10
2.1 Ferramentas utilizadas	10
2.2 Justificativa	10
2.3 Funcionamento	10
2.4 IS Canvas	10
2.5 Jornada de usuário	12
2.6 Fluxo de diálogo	12
2.7 Implementação do Chatbot	13

1- Front-end, Back-end e Banco de dados

1.1 Ferramentas utilizadas

No desenvolvimento do site foram usadas as seguintes ferramentas e linguagens:

1. Github para armazenamento do código
2. IDE Eclipse (java developers) para desenvolvimento do software
3. PostgreSQL como gerenciador de banco de dados
4. PgAdmin 4 administração e desenvolvimento para PostgreSQL
5. Framework Spark para para processamento de dados

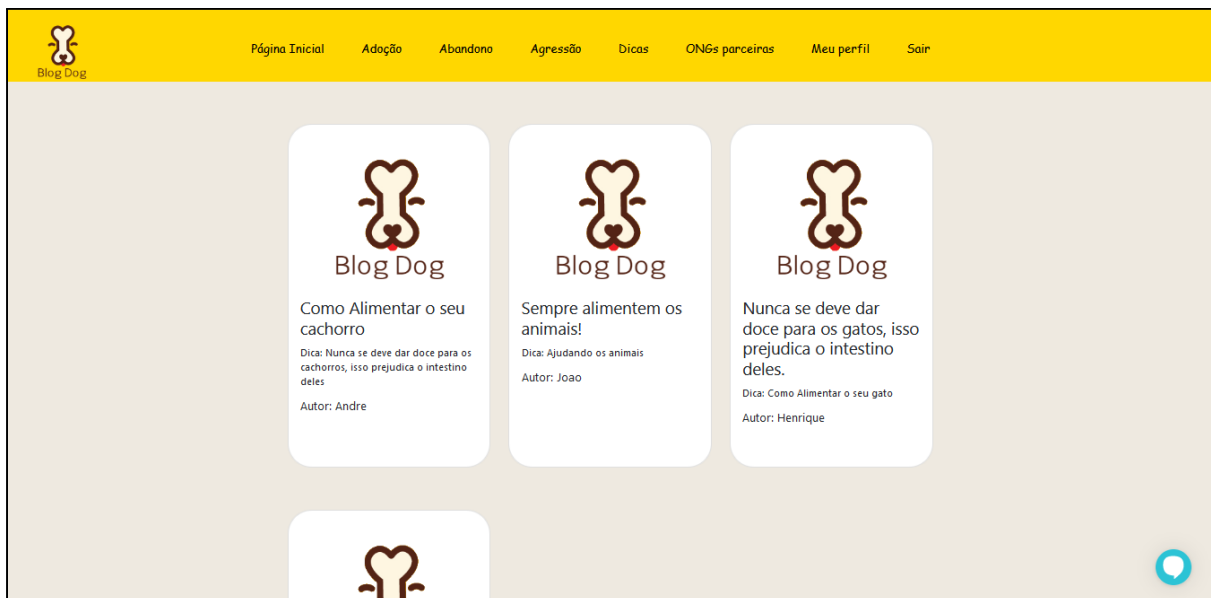
1.2 Funcionamento

Através do nosso Back-end utilizando a linguagem Java, nós fazemos a conexão com o nosso Banco de Dados que nós utilizamos PostgreSQL, através do framework Spark o nosso site consegue fazer requisições como POST, GET, UPDATE, DELETE pro nosso BD, e o nosso front-end está conectado com o back-end e com o banco de dados e qualquer aplicação feita pelo usuário no front-end será atualizado na hora tanto no Banco de Dados quanto na própria interface.

1.3 Links

Repositório de códigos: <https://github.com/andrefellipe3/BlogDog>

Interface de Dicas:



Coloque uma Dica!

Nome:

Ex: Jose Ferreira

Título da Dica:

Ex: Ajudando os animais

Conteúdo:

Ex: É recomendado o animal...

Inserir Dica

BlogDog - Todos os direitos reservados - 2020

Aplicação de Dica:

```
//Dica
post("/cadastrardica", (request, response) -> serv.addDicas(request, response));
get("/dicas", (request, response) -> serv.getDicas(request, response));
```

POST e GET do Dicas:

```
blogdog  ▶  src/main/java  ▶  dog  ▶  Servicos  ▶
109
110 //-----
111 //Dica
112
113 //Pegar a informação no formulário e enviar pro inserirDica
114 public Object addDicas(Request request, Response response)
115 {
116     response.header("Access-Control-Allow-Origin", "");
117     response.header("Content-Type", "application/json");
118     request.attribute("org.eclipse.jetty.multipartConfig", new MultipartConfigElement("/temp"));
119     String nome_dica = request.queryParams("nome_dica");
120     String titulo_dica = request.queryParams("titulo_dica");
121     String conteudo_dica = (request.queryParams("conteudo_dica"));
122     Dica dica = new Dica(nome_dica, titulo_dica, conteudo_dica);
123
124     BD.inserirDica(dica);
125
126     response.status(201); // 201 Created
127     response.redirect("http://localhost:5500/src/main/resource/dicas.html");
128     return 0;
129 }
130
131
132 //Solicitar as informações para no nosso banco de dados sobre Dicas
133 public Object getDicas (Request request, Response response)
134 {
135     response.header("Access-Control-Allow-Origin", "");
136     response.header("Content-Type", "application/json");
137
138     JSONArray resp = new JSONArray();
139     Dica[] D = BD.getDicas();
140
141     for(int i = 0; i < D.length ; i++)
142         resp.put(D[i].toJson());
143
144     return resp;
145 }//end getDicas
146
```

Código da Tabela de Dicas no Banco de Dados:

```
-- Table: public.dica

-- DROP TABLE public.dica;

CREATE TABLE public.dica
(
    id integer NOT NULL DEFAULT nextval('dica_id_seq'::regclass),
    nome text COLLATE pg_catalog."default",
    titulo text COLLATE pg_catalog."default",
    conteudo text COLLATE pg_catalog."default",
    CONSTRAINT dica_pkey PRIMARY KEY (id)
)


TABLESPACE pg_default;

ALTER TABLE public.dica
    OWNER to postgres;
```

Tabela de Dicas no Banco de Dados:

	id [PK] integer	nome text	titulo text	conteudo text
1	1	Andre	Como Alimentar o seu cachorro	Nunca se deve dar doce para os cachorros, isso prejudica o intesti...
2	26	Joao	Sempre alimentem os animais!	Ajudando os animais
3	27	Henriq...	Nunca se deve dar doce para os gat...	Como Alimentar o seu gato
4	28	Ricardo	Nunca se deve dar doce para os gat...	Ajudando os animais

Interface ONGs parceiros:

Página InicialAdoçãoAbandonoAgressãoDicasONGs parceiras

Seja um parceiro agora!

Nome da ONG:

Telefone:

Cidade:

[Cadastrar ONG](#)

Ongs Parceiras

#	Nome	Telefone	Cidade
10	Pets pela vida	(31)923842138	Belo Horizonte
11	Salve os Animais	(31)985342913	Sete Lagoas
13	Ajudando os Gatos	(31)934523412	Rio de Janeiro

BlogDog - Todos os direitos reservados - 2020

Aplicação de ONG:

```
//ONG
post("/cadastrar", (request, response) -> serv.addONG2(request, response));
get("/ong", (request, response) -> serv.getONG(request, response));
```

POST e GET de ONGs:

```
blogdog > src/main/java > dog > Servicos >
70
71 //-----
72 //ONG
73
74 //Inserindo no Banco de Dados as informações referente a ONG
75 public Object addONG2(Request request, Response response)
76 {
77     response.header("Access-Control-Allow-Origin", "*");
78     response.header("Content-Type", "application/json");
79     request.attribute("org.eclipse.jetty.multipartConfig", new MultipartConfigElement("/temp"));
80     String nome = request.queryParams("nome");
81     String cidade = request.queryParams("cidade");
82     String telefone = (request.queryParams("telefone"));
83     ONG ong = new ONG(nome, cidade, telefone);
84
85     BD.inserirOng(ong);
86
87     response.status(201); // 201 Created
88     response.redirect("http://localhost:5500/src/main/resource/ongs.html");
89     return 0;
90 }
91
92 //Buscar no Banco de Dados as informacoes sobre as ONGS
93 public Object getONG (Request request, Response response)
94 {
95     response.header("Access-Control-Allow-Origin", "*");
96     response.header("Content-Type", "application/json");
97
98     JSONArray resp = new JSONArray();
99     ONG[] D = BD.getOngs();
100
101     for(int i = 0; i < D.length ; i++)
102         resp.put(D[i].toJson());
103
104     return resp;
105 } //end getONG
106
```

Código da Tabela de ONGs no Banco de Dados:

```
-- Table: public.ongs

-- DROP TABLE public.ongs;

CREATE TABLE public.ongs
(
    id integer NOT NULL DEFAULT nextval('ongs_id_seq'::regclass),
    nome text COLLATE pg_catalog."default",
    cidade text COLLATE pg_catalog."default",
    telefone character(20) COLLATE pg_catalog."default",
    CONSTRAINT ongs_pkey PRIMARY KEY (id)
)

TABLESPACE pg_default;

ALTER TABLE public.ongs
    OWNER to postgres;
```

Tabela de ONGs no Banco de Dados:

	<div><div><div><div><div><div></div><div></div></div></div><div><div></div></div></div><div>id</div><div>[PK] integer</div></div></div>	<div><div><div><div><div><div></div><div></div></div></div><div><div></div></div></div><div>nome</div><div>text</div></div></div>	<div><div><div><div><div><div></div><div></div></div></div><div><div></div></div></div><div>cidade</div><div>text</div></div></div>	<div><div><div><div><div><div></div><div></div></div></div><div><div></div></div></div><div>telefone</div><div>character (20)</div></div></div>
1	10	Pets pela vida	Belo Horizonte	(31)923842138
2	11	Salve os Animais	Sete Lagoas	(31)985342913
3	13	Ajudando os Gatos	Rio de Janeiro	(31)934523412

Interface de Agressão:

Página Inicial Adoção Abandono Agressão Dicas ONGs parceiras Meu perfil Sair

Agressão de animais

Para minimizar o máximo possível o problema o site permite que voce denuncie uma situação de agressão à um animal ou relate que resgatou-o e necessita de algum tipo de auxilio de nossas organizações.

Denunciar

Nome:

Telefone:

Email:

Cidade:

Animal:

Informações Complementares:

[Denunciar Agressão](#)

#	Nome	Telefone	E-mail	Cidade	Espécie do animal	Informações complementares
1	Andre	(31)982321234	andre@hotmail.com	Belo Horizonte	Gato	O gato se encontra bastante magro
4	Jose	(31)933215523	jose@hotmail.com	Rio de Janeiro	Cachorro	Bastante ferido e precisando de um cuidado médico
5	Samuel	(31)933215523	samuel@hotmail.com	Belo Horizonte	Cachorro	Ele não esta comendo muito

Aplicação de Agressão:

```
//Agressao
post("/cadastrararessao", (request, response) -> serv.addAgressao(request, response));
get("/agressao", (request, response) -> serv.getAgressao(request, response));
```

POST e GET de Agressão:

```
//-----
//Agressao
//Adicionar no Banco de Dados as Dicas
public Object addAgressao(Request request, Response response)
{
    response.header("Access-Control-Allow-Origin", "*");
    response.header("Content-Type", "application/json");
    request.attribute("org.eclipse.jetty.multipartConfig", new MultipartConfigElement("/temp"));
    String nome_agressao = request.queryParams("nome_agressao");
    String telefone_agressao = request.queryParams("telefone_agressao");
    String email_agressao = request.queryParams("email_agressao");
    String cidade_agressao = request.queryParams("cidade_agressao");
    String animal_agressao = request.queryParams("animal_agressao");
    String info_agressao = (request.queryParams("info_agressao"));

    Agressao agressao = new Agressao (nome_agressao, telefone_agressao, email_agressao, cidade_agressao,
    animal_agressao,info_agressao);

    BD.inserirAgressao(agressao);

    response.status(201); // 201 Created
    response.redirect("http://localhost:5500/src/main/resource/relatos.html");
    return 0;
}
```



```

//Buscar no Banco de Dados as Dicas
public Object getAgressao (Request request, Response response)
{
    response.header("Access-Control-Allow-Origin", "*");
    response.header("Content-Type", "application/json");

    JSONArray resp = new JSONArray();
    Agressao[] D = BD.getAgressao();
    for(int i = 0; i < D.length ; i++)
    {
        resp.put(D[i].toJson());
    }

    return resp;
} //end getDicas
}

```

Código da Tabela de Agressão no Banco de Dados:

```

-- Table: public.agressao

-- DROP TABLE public.agressao;



CREATE TABLE public.agressao
(
    id integer NOT NULL DEFAULT nextval('agressao_id_seq'::regclass),
    nome text COLLATE pg_catalog."default",
    telefone text COLLATE pg_catalog."default",
    email text COLLATE pg_catalog."default",
    cidade text COLLATE pg_catalog."default",
    animal text COLLATE pg_catalog."default",
    info text COLLATE pg_catalog."default",
    CONSTRAINT agressao_pkey PRIMARY KEY (id)
)

TABLESPACE pg_default;

ALTER TABLE public.agressao
    OWNER to postgres;

```

Tabela Agressão no Banco de Dados:

	id [PK] 	nome text 	telefone text 	email text 	cidade text 	animal text 	info text 
1	1	Andre	(31)982321...	andre@hotmail.com	Belo Horiz...	Gato	O gato se encontra bastante magro
2	4	Jose	(31)933215...	jose@hotmail.com	Rio de Jan...	Cachorro	Bastante ferido e precisando de um cuidado mé...
3	5	Samuel	(31)933215...	samuel@hotmail.c...	Belo Horiz...	Cachorro	Ele não esta comendo muito

2- Sistemas Inteligentes

2.1 Ferramentas utilizadas

1. Chatbot Blip

2.2 Justificativa

Foi escolhida como Inteligência Artificial o chatbot pois ele tem intenção de ajudar e solucionar problemas que o usuário comum pode ter ao utilizar nosso site, além disso, ele possui funcionalidades interativas, no qual dependendo dos dados que o usuário nos fornece, a inteligência irá resolver de forma mais prática e intuitiva.

2.3 Funcionamento

É um assistente virtual que irá auxiliar tutores e possíveis tutores de animais de estimação.

2.4 IS Canvas

Ferramental de IA Um agente inteligente de texto	Entradas Informações do usuário	Proposição de valor Centralizar o contato de ONGs e instituições de apoio a animais domésticos Centralizar informações relevantes em relação a criação de animais domésticos Auxiliar pessoas mais leigas, a encontrar informações armazenadas no chatbot, sem a necessidade de ser realizada uma pesquisa Fácil uso por parte dos usuários Estar sempre disponível e ao alcance do usuário do site Ser atrativo ao usuário	Equipe Programadores	Clientes Tutores de animais domésticos Pessoas interessadas em se tornar tutores de animais domésticos Pessoas interessadas em se informar sobre animais domésticos Visitantes do site em geral
	Saídas Dicas de cuidados com animais domésticos Cardápio com receitas caseiras Contato de ONGs e instituições da causa dos animais domésticos Como identificar maus-tratos nos animais Como ajudar um possível animal abandonado		Stakeholders Chaves Desenvolvedores do site	
Custos Custo com a equipe de IA		Receitas Anúncios e doações		

1. Ferramenta de IA

A ferramenta escolhida para o projeto foi um agente de texto (ChatBot), a fim de auxiliar nossos usuários através da conversa com o robô. A plataforma escolhida para fornecer o chatbot foi a “Take Blip (Take.net)”, pois foi a que melhor atendeu nossos requisitos.

2. Entradas

As entradas do chat serão baseadas no que o chatbot perguntar, por exemplo o que o usuário deseja fazer em nosso site, daí em diante o chatbot guiará o usuário em toda a conversa.

3. Saídas

As saídas são mensagens para início de conversa -boas vindas- respostas e soluções para as dúvidas e desejos do usuário.

4. Proposição de valor

Tendo em vista que nosso projeto é um assistente virtual, seu objetivo é facilitar e/ou agilizar processos que envolvem o cliente.

5. Equipe

A equipe no momento se resume aos desenvolvedores, porém futuramente existe a possibilidade da inclusão de outros trabalhadores

6. Stakeholders chave

As pessoas que mais teriam influência na implementação do chatbot seriam os desenvolvedores do próprio site.

7. Clientes

O público alvo são quaisquer pessoas que acessarem nosso site, inclusive pessoas que querem adotar/colocar para adoção seu pet ou aquelas que apenas desejam conhecer nossos serviços.

8. Custos

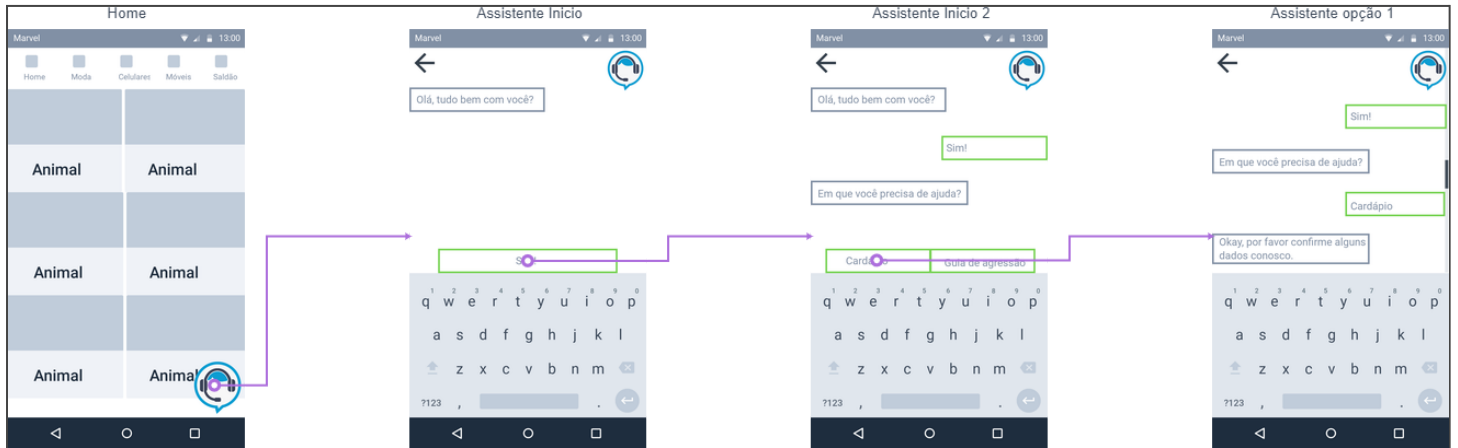
Os custos no momento seriam apenas o pagamento da equipe de programadores, ou seja, não existe nenhum gasto com plataformas externas.

9. Receita

São anúncios dentro do site e doações. Não existe pretensão de acrescentar outro método.

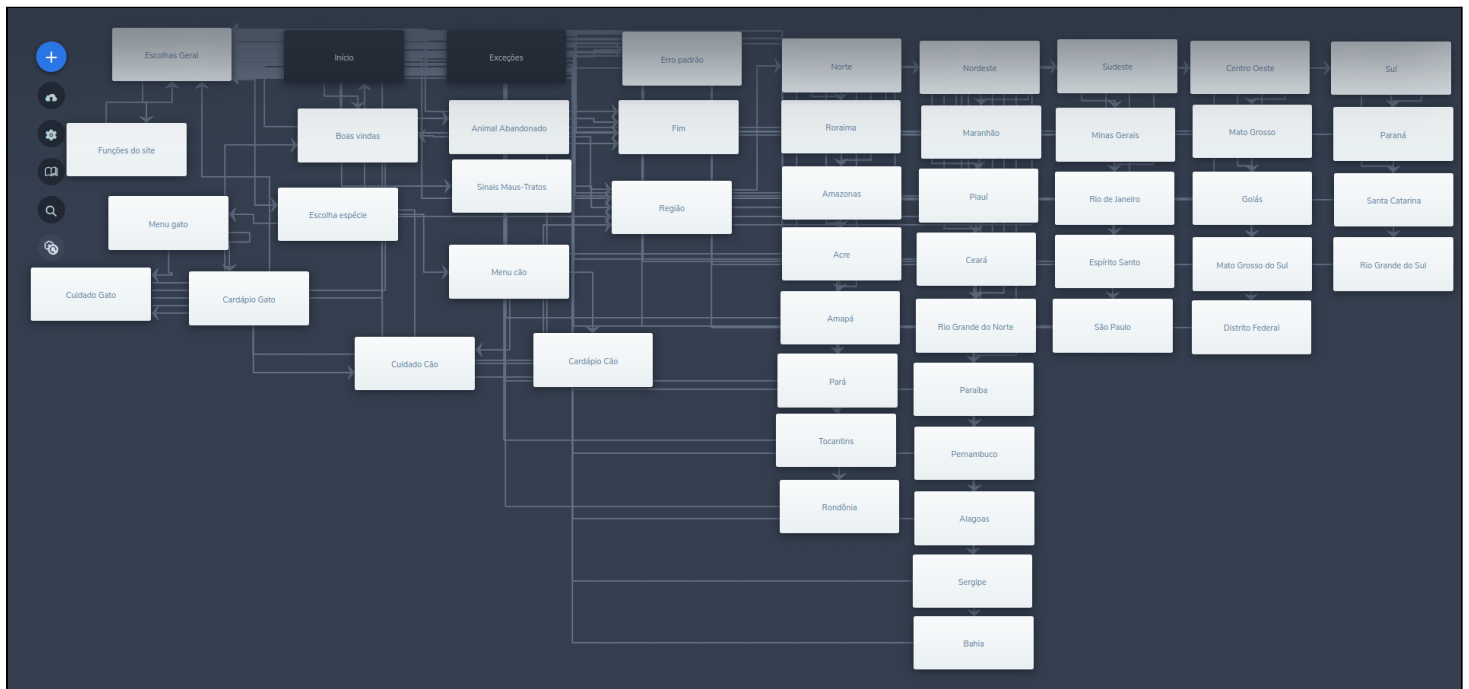
2.5 Jornada de usuário

O usuário conseguirá acessar e utilizar nosso chatbot em qualquer momento e lugar do site. Confira logo abaixo o protótipo da jornada de usuário feito na plataforma MarvelApp:



2.6 Fluxo de diálogo

O fluxo de diálogo do chatbot é que o cliente possa, de maneira rápida e simples, resolver ou alcançar seu objetivo final. (imagem pode ser encontrada no nosso git: <https://github.com/andrefellipe3/BlogDog>).



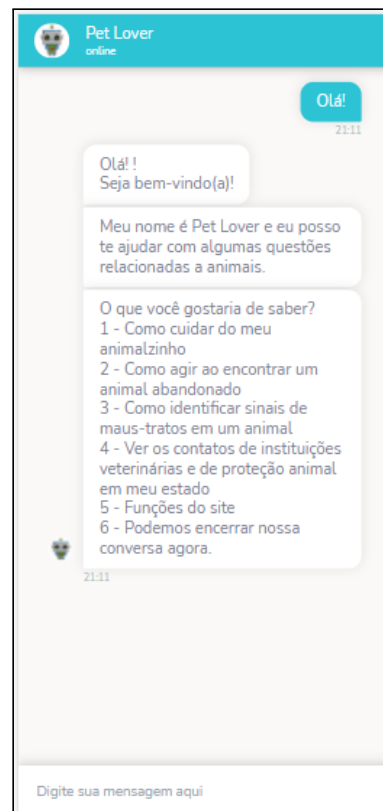
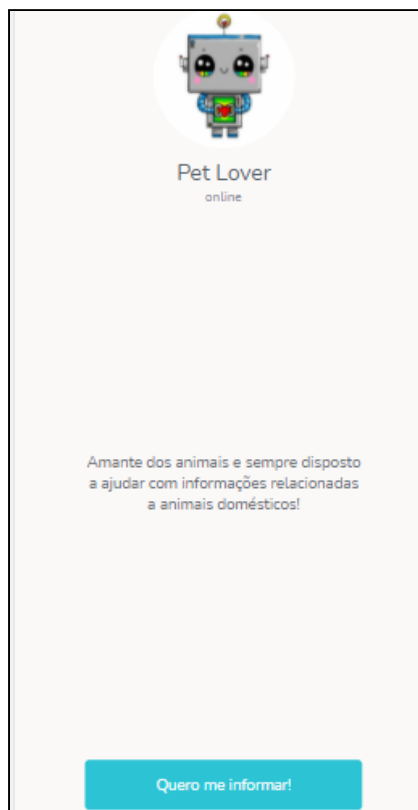
2.7 Implementação do Chatbot

A implementação do chatbot é muito simples, basta adicionar o script no html do site e é feita a conexão com o servidor do Blip.

```
<script src="https://unpkg.com/blip-chat-widget" type="text/javascript">
</script>
<script>
  (function () {
    window.onload = function () {
      new BlipChat()
        .withAppKey('cGVkcm8yODo5NjgxZjNkMS1hZWU3LTQxNTctOTk5OC0yMjcxNmFmYTgzNjY=')
        .withButton({"color":"#2CC3D5","icon":""})
        .withCustomCommonUrl('https://chat.blip.ai/')
        .build();
    }
  })();
</script>
```

O chatbot possui as seguintes funcionalidades:

1. Guiar novos usuários do site
2. Dicas de como cuidar do seu animal
3. Cardápio de receitas caseiras para animais de estimação
4. Auxiliar pessoas que encontrarem um animal abandonado
5. Auxiliar pessoas que encontrarem um animal em situação de maus-tratos
6. Mostrar contatos de instituições veterinárias e de proteção nas regiões brasileiras



O chatbot pode ser testado no link a seguir:

<https://chat.blip.ai/?appKey=cGVkcm8yODo5NjgxZjNkMS1hZWU3LTQxNTctOTk5OC0yMjcxNmFmYTgzNjY=>