Trabalho Interdisciplinar: Back-End

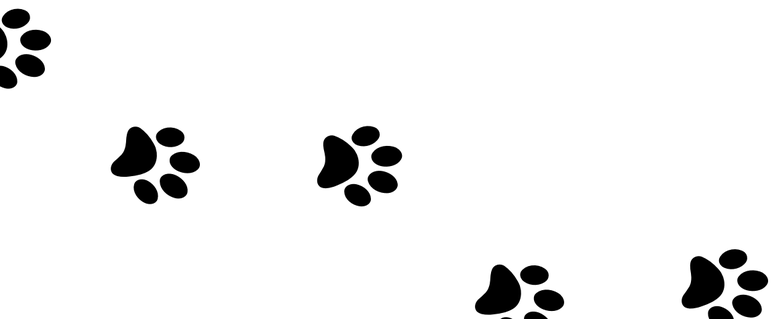
PUC Minas

Membros da equipe:

Davi Cândido de Almeida , Letícia da Silva Rocha, Matheus Eduardo Campos Soares, Rayssa Mell de Souza Silva

Professores:

Max, Sandro, Waldimir



PUC Minas

SOS Bichinhos

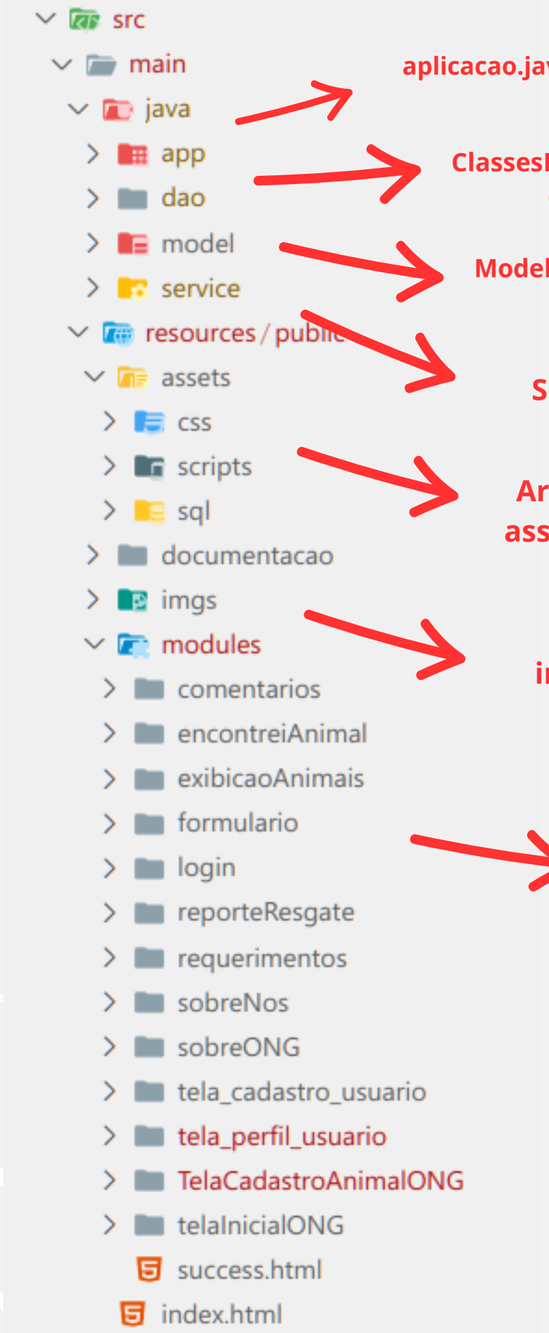
PUC Minas

Tópicos

Sprint-3

- Organização de pastas do projeto
  - Aplicação, Model, Service, Resouce...
- Fluxo dos CRUDS
  - Cadastro de usuario, Formulario, Comentarios, Animais
- Definição do Sistema Inteligente

Organização de pastas



aplicacao.java - Responsavel por iniciar a aplicação

ClassesDAO.java: Classe responsável pelas operações CRUD no BD relacionadas aos objetos

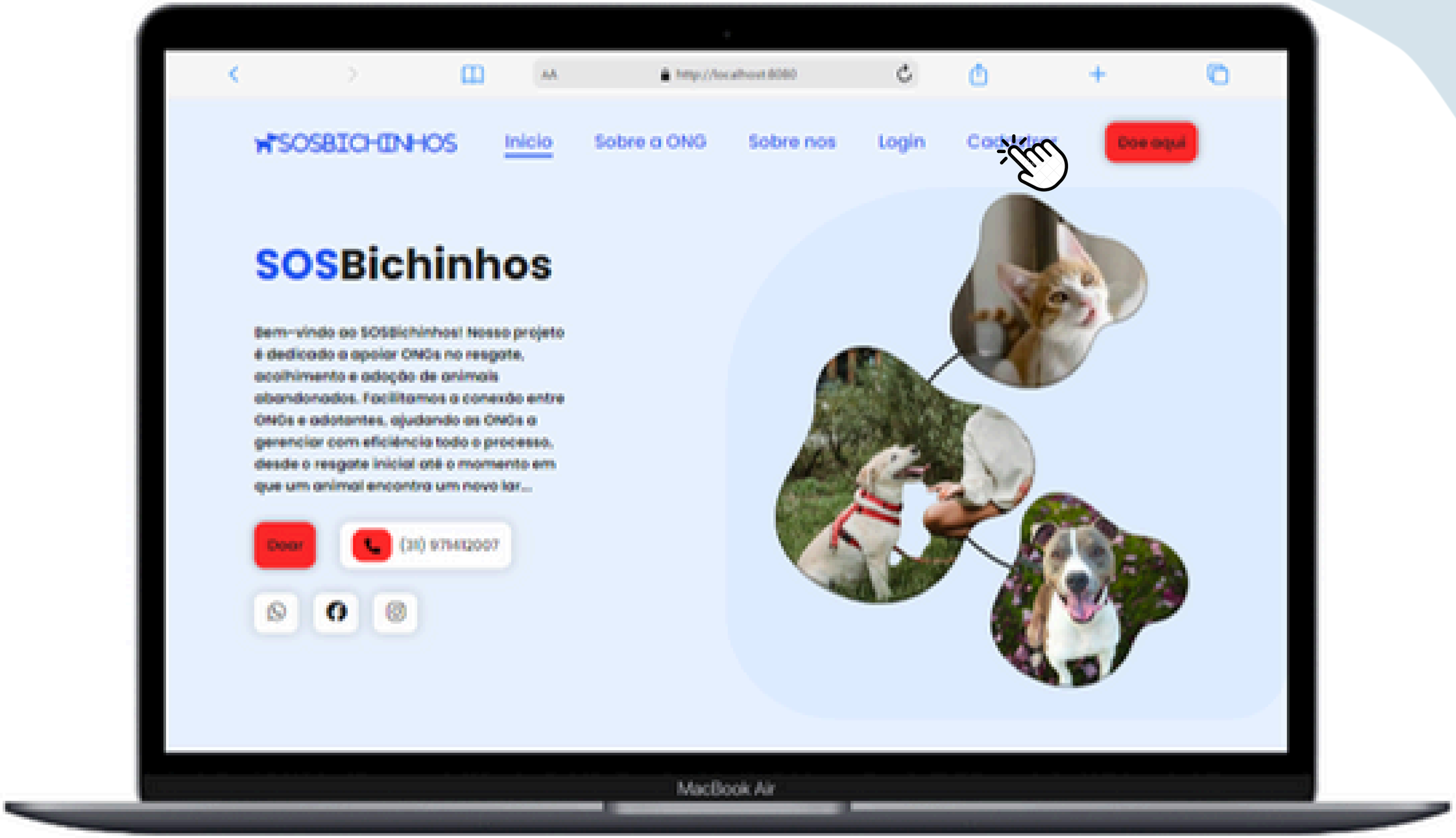
Model: Contém as classes que definem os modelos de dados (entidades).

Service: encapsula as operações de negócio

Armazena os arquivos utilizados pela aplicação
assets: Arquivos de estilos (CSS), JavaScript e SQL

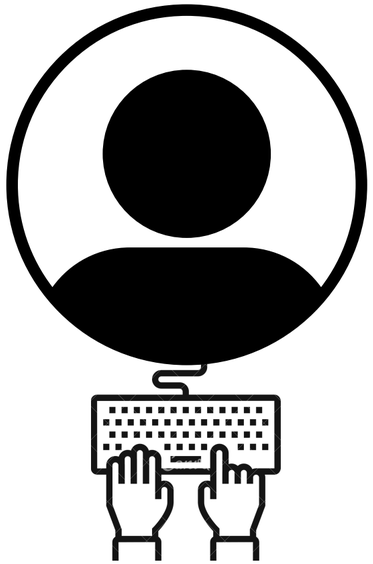
imgs: Imagens utilizadas no frontend.

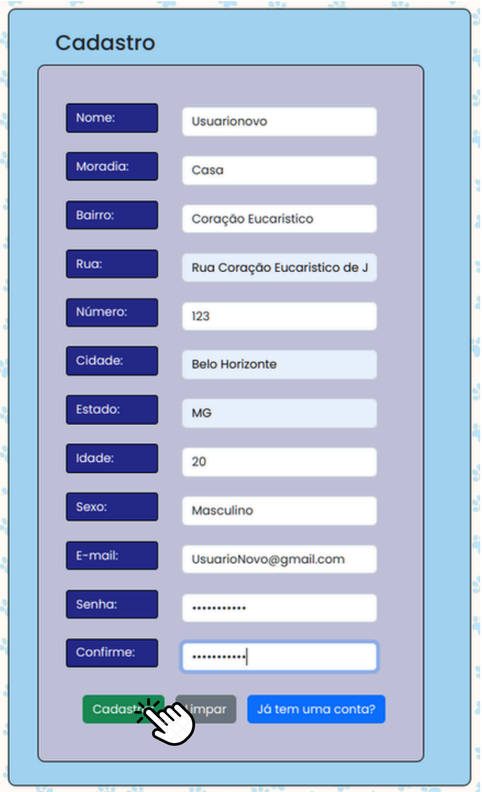
Modules: Aquivos Html de cada tela.


SOSBICHINHOS
Início
Sobre a ONG
Sobre nos
Login
Doe aqui
SOSBichinhos
Bem-vindo ao SOSBichinhos! Nosso projeto é dedicado a apoiar ONGs no resgate, acolhimento e adoção de animais abandonados. Facilitamos a conexão entre ONGs e adotantes, ajudando as ONGs a gerenciar com eficiência todo o processo, desde o resgate inicial até o momento em que um animal encontra um novo lar...
Doar
(31) 971482007
MacBook Air

Levando dados para o Back-End

Cadastro de pessoa





HTML

```
<form id="form-contato">
  <!-- Campos existentes -->
  <div class="form-group row align-items-center mb-3">...</div> flex
  <!-- Moradia -->
  <div class="form-group row align-items-center mb-3">...</div> flex
  <!-- Campos de endereço -->
  <div class="form-group row align-items-center mb-3">...</div> flex
  <div class="form-group row align-items-center mb-3">...</div> flex
  <div class="form-group row align-items-center mb-3">...</div> flex
  <div class="form-group row align-items-center mb-3">...</div> flex
  <div class="form-group row align-items-center mb-3">...</div> flex
  <!-- Idade -->
  <div class="form-group row align-items-center mb-3">...</div> flex
  <!-- Sexo -->
  <div class="form-group row align-items-center mb-3">...</div> flex
  <!-- Campo E-mail -->
  <div class="form-group row align-items-center mb-3">...</div> flex
  <!-- Senha, confirmar senha, e botões -->
  <div class="form-group row align-items-center mb-3">...</div> flex
  <div class="form-group row align-items-center mb-3">...</div> flex
  <div class="form-group row align-items-center">...</div> flex
</form>
```

```
async function HG_createUserAndEndereco(user, endereco) {
    try {
        // Criação do usuário
        console.log(user);
        const userResponse = await fetch('/usuario', {
            method: 'POST',
            headers: {
                'Content-Type': 'application/json',
            },
            body: JSON.stringify(user)
        });

        if (!userResponse.ok) {
            throw new Error('Erro ao criar o usuário');
        }

```

JAVASCRIPT

```
const createdUser = await userResponse.json();
  console.log(createdUser);
  LCdisplayMessage("Sucesso ao criar usuário");

  // Atribuir o ID do usuário recém-criado ao endereço
  endereco.id_pessoa = createdUser.id;

  // Criação do endereço
  console.log(endereco);
  const enderecoResponse = await fetch('/endereco', {
      method: 'POST',
      headers: {
          'Content-Type': 'application/json',
      },
      body: JSON.stringify(endereco)
  });
```

Recebendo dados no Back-End

Aplicação

```
// ------------------------Aplicações CRUD Pessoa  ------------------------------------- //

        post("/usuario", (request, response) -> aplicacaousuario.add(request, response));

        get("/usuario/:id", (request, response) -> aplicacaousuario.get(request, response));

        post("/usuario/update/:id", (request, response) -> aplicacaousuario.update(request, response));

        post("/usuario/delete/:id", (request, response) -> aplicacaousuario.remove(request, response));

        get("/usuario", (request, response) -> aplicacaousuario.getAll(request, response));

        // ------------------------Aplicações CRUD Endereço  ------------------------------------- //

        post("/endereco", (request, response) -> aplicacaoEndereco.add(request, response));

        get("/endereco/:id", (request, response) -> aplicacaoEndereco.get(request, response));

        post("/endereco/update/:id", (request, response) -> aplicacaoEndereco.update(request, response));

        post("/endereco/delete/:id", (request, response) -> aplicacaoEndereco.remove(request, response));

        get("/endereco", (request, response) -> aplicacaoEndereco.getAll(request, response));
```

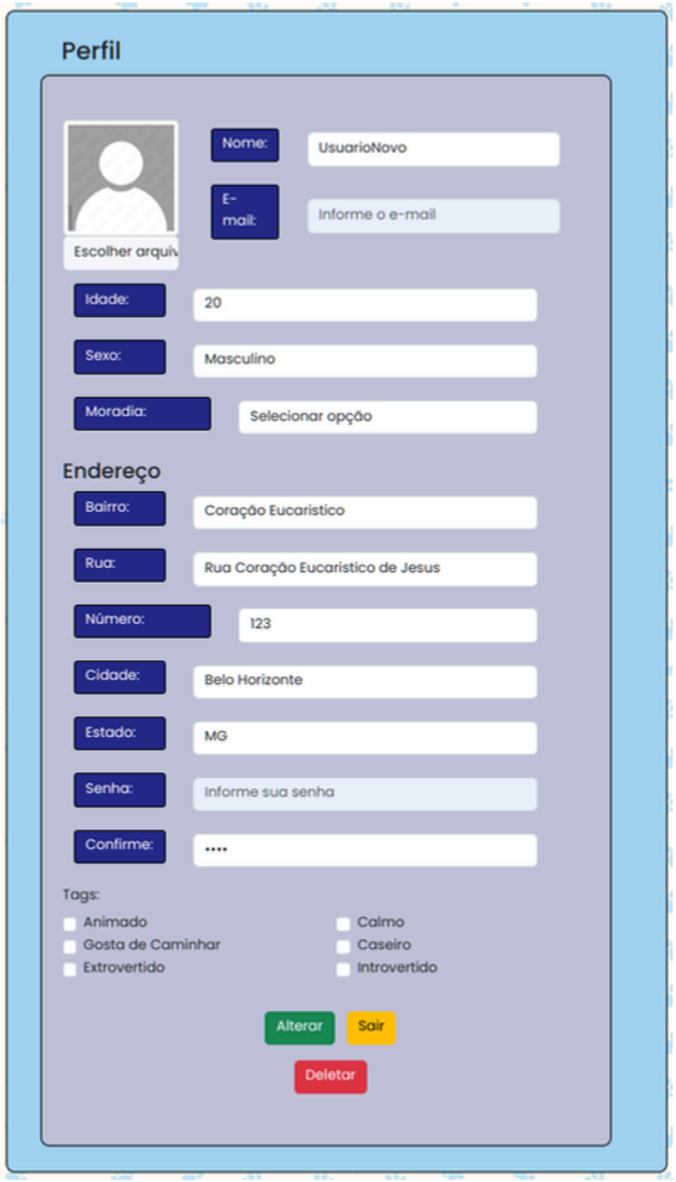
PessoaService

```
public Object add(Request request, Response response) {
    Gson gson = new Gson();
    Pessoa registro = gson.fromJson(request.body(), Pessoa.class);

    int id = this.pessoaDAO.getMaxId_pessoa() + 1; // Corrigido o getMaxId
    registro.setId(id);
    System.out.println(registro);

    pessoaDAO.inserirpessoa(registro);

    response.status(201); // 201 Created
    return id;
}
```

EnderecoService

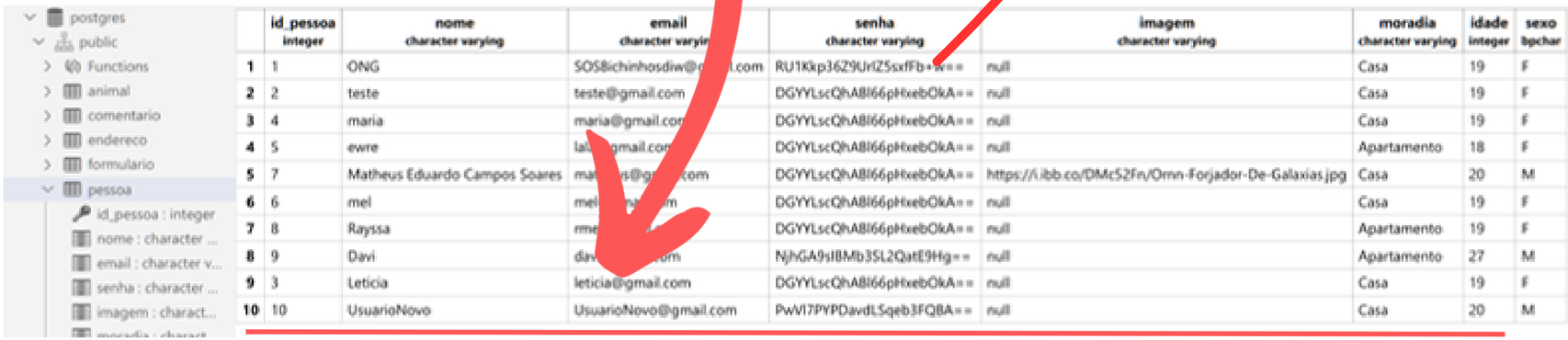
```
public Object add(Request request, Response response) {
        Gson gson = new Gson();
        Endereco registro = gson.fromJson(request.body(), Endereco.class); // Altere para Endereco

        // Gera o próximo id para o Endereco
        int id = this.enderecoDAO.getMaxId_endereco() + 1; // Altere para getMaxIdEndereco()
        int id_pessoa = this.pessoaDAO.getMaxId_pessoa();

        registro.setId_pessoa(id_pessoa);
        registro.setId_endereco(id); // Altere para setId_endereco

        System.out.println(registro);

        enderecoDAO.inserirEndereco(registro); // Altere para inserirEndereco

        response.status(201); // 201 Created
        return id;
    }
```

Back-End -> PessoaDao




```java
public boolean inserirpessoa(Pessoa pessoa) {
    boolean status = false;
    try {
        this.maxId_pessoa = (pessoa.getId() > this.maxId_pessoa) ? pessoa.getId() : this.maxId_pessoa;
        String senhaCriptografada = CriptografiaAES.criptografar(pessoa.getSenha());

        Statement st = conexao.createStatement();
        String sql = "INSERT INTO pessoa (id_pessoa, nome, email, senha, moradia, imagem, idade, sexo) " +
                     "VALUES (" +
                     pessoa.getId() + ", '" +
                     pessoa.getNome() + "', '" +
                     pessoa.getEmail() + "', '" +
                     senhaCriptografada + "', '" +
                     pessoa.getMoradia() + "', '" +
                     pessoa.getImagem() + "', '" +
                     pessoa.getIdade() + "', '" +
                     pessoa.getSexo() + "')";

        st.executeUpdate(sql);
        st.close();
        status = true;
    } catch (Exception e) {
        e.printStackTrace();
    }
    return status;
}
```

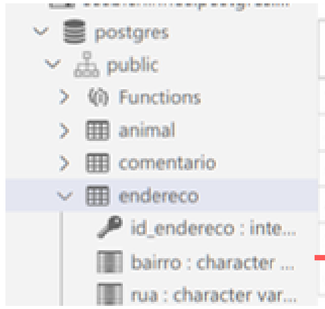
CriptografiaAES



	id_pessoa integer	nome character varying	email character varyin	senha character varying	imagem character varying	moradia character varying	idade integer	sexo bpchar
1	1	ONG	SOS8ichinhosdiw@	.com	RU1Kkp36Z9UrlZ5sxffb+	null	Casa	19	F
2	2	teste	teste@gmail.com		DGYYLscQhA8l66pHxebOkA==	null	Casa	19	F
3	4	maria	maria@gmail.co		DGYYLscQhA8l66pHxebOkA==	null	Casa	19	F
4	5	ewre	la	gmail.co	DGYYLscQhA8l66pHxebOkA==	null	Apartamento	18	F
5	7	Matheus Eduardo Campos Soares	mat	com	DGYYLscQhA8l66pHxebOkA==	https://i.ibb.co/DMc52Fn/Omn-Forjador-De-Galaxias.jpg	Casa	20	M
6	6	mel	mel	m	DGYYLscQhA8l66pHxebOkA==	null	Casa	19	F
7	8	Rayssa	rme		DGYYLscQhA8l66pHxebOkA==	null	Apartamento	19	F
8	9	Davi	dav	om	NjhGA9slBMb3SL2QatE9Hg==	null	Apartamento	27	M
9	3	Leticia	leticia@gmail.com		DGYYLscQhA8l66pHxebOkA==	null	Casa	19	F
10	10	UsuarioNovo	UsuarioNovo@gmail.com		PwVI7PYPDavdLSqeb3FQBA==	null	Casa	20	M

Back-End -> EnderecoDao

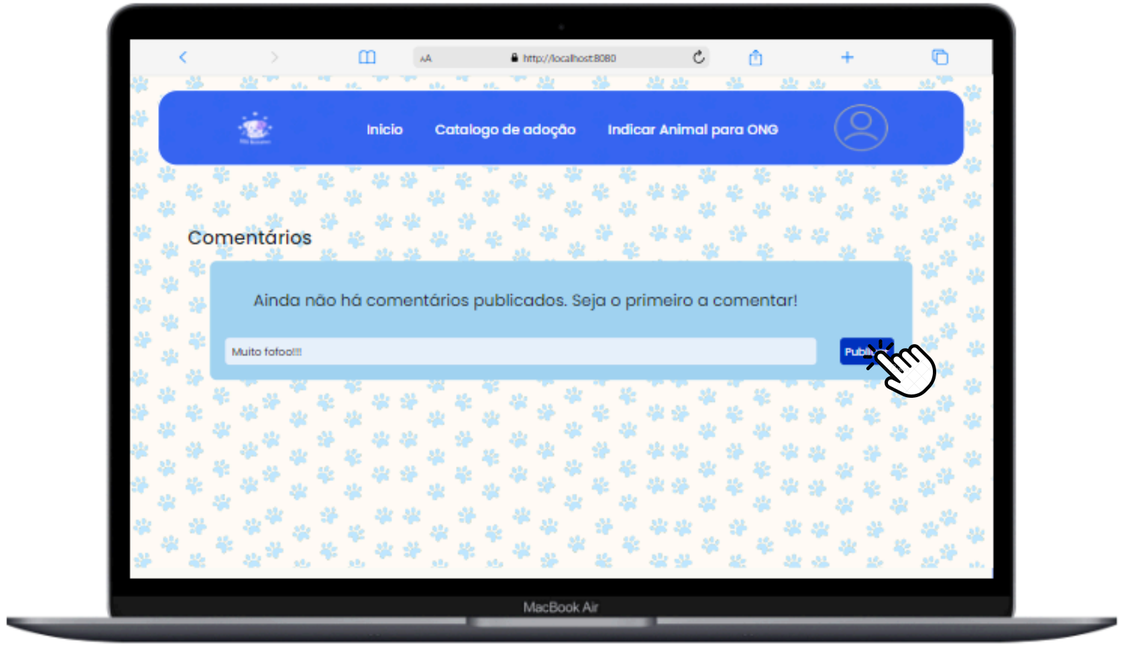

```java
public boolean inserirEndereco(Endereco endereco) {
    boolean status = false;
    String sql = "INSERT INTO endereco (id_endereco, bairro, rua, numero, cidade, estado, id_pessoa) VALUES (?, ?, ?, ?, ?, ?, ?)";

    try (PreparedStatement pstmt = conexao.prepareStatement(sql)) {
        pstmt.setInt(1, endereco.getId_endereco());
        pstmt.setString(2, endereco.getBairro());
        pstmt.setString(3, endereco.getRua());
        pstmt.setString(4, endereco.getNumero());
        pstmt.setString(5, endereco.getCidade());
        pstmt.setString(6, endereco.getEstado());
        pstmt.setInt(7, endereco.getId_pessoa());

        pstmt.executeUpdate();
        status = true;
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return status;
}
```



	id_endereco integer	bairro character varying	rua character varying	numero character varying	cidade character varying	estado character varying	id_pessoa integer
1	1	Dom Cabral	Av. Trinta e Um de	1020	Belo Horizonte	Minas Gerais	1
2	3	Barreiro	Rua Wilson Tavar	587	Belo Horizonte	Minas Gerais	7
3	14	Nossa senhora desaparecida	Morrin dos cria	777	Tijuco	Pará	9
4	13	Dom Cabral	Rua L	4	BH	Minas Gerais	8
5	15	Coração Eucaristico	Rua Coração Eucaristico de Jesus	123	Belo Horizonte	MG	10
6	2	Coração Eucarístico	Avenida 31 de março	999	vespasiano	minas gerais	3

Levando dados para o Back-End

Comentários



HTML

```html
<form id="LCform">
    <input type="text" placeholder="Digite o seu comentário..."
    name="comment" id="LCcampoComment">
    <button type="submit" id="LCbtnPublicar">Publicar</button>
</form>
```

Javascript

```javascript
async function LCcreateComment(comment) {
    //  tentar fazer a chamada
    try {
        //  definir a chamada HTTP do JSON Server
        const response = await fetch(`${LCapiUrl}`, {
            method: "POST",
            headers: {
                "Content-Type": "application/json",
            },
            body: JSON.stringify(comment),
        });
        //  mostrar resultado
        LCdisplayMessage("Sucesso ao criar comentario!");
        //  atualizar a pagina
        LCmostrarComentarios();
    }
    //  chamada de erro
    catch (error) {
        console.error("Error:", error);
        LCdisplayMessage("Erro ao criar comentario (JSON Server indisponível).");
    }
}
```

Recebendo dados no Back-End

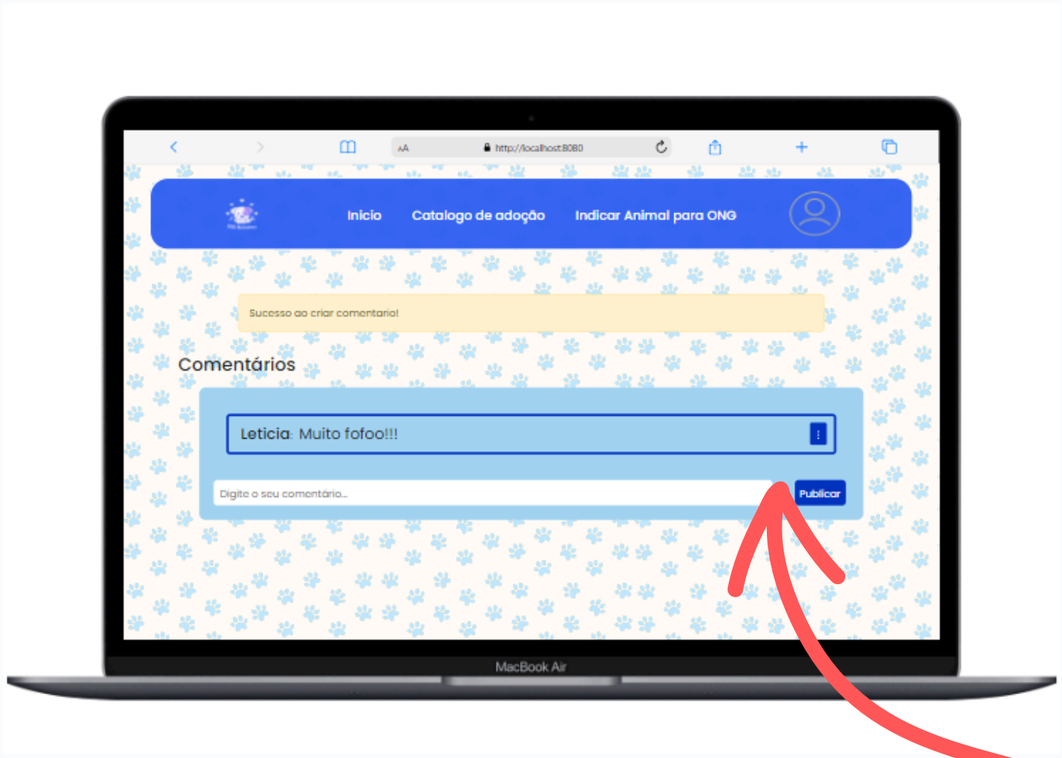
Aplicação

```
// ------------------------Aplicações CRUD Comentario  ---------------------------------------- //


 post("/comentario", (request, response) -> comentarioService.add(request, response));

 get("/comentario/:id", (request, response) -> comentarioService.get(request, response));

 post("/comentario/update/:id", (request, response) -> comentarioService.update(request, response));

 post("/comentario/delete/:id", (request, response) -> comentarioService.remove(request, response));

 get("/comentario", (request, response) -> comentarioService.getAll(request, response));

 get("/comentario/animal/:id", (request, response) -> comentarioService.getAnimal(request, response));

```

ComentarioService

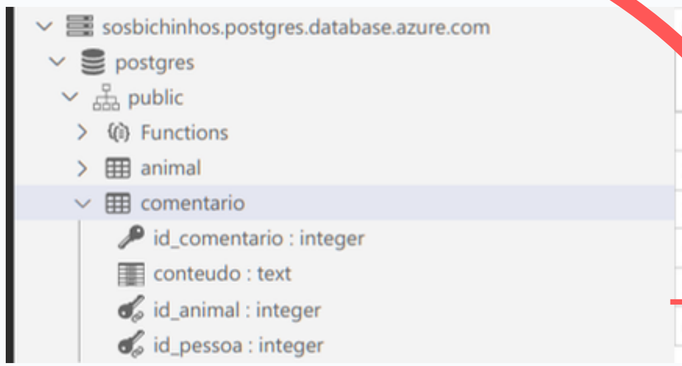
```

public Object add(Request request, Response response) {
    Gson gson = new Gson();

    Comentario registro = gson.fromJson(request.body(), Comentario.class);

    int id = this.ComentarioDAO.getMaxId() + 2;  // Corrigido o getMaxId

    registro.setId(id);
    System.out.println(registro);

    ComentarioDAO.inserirComentario(registro);

    response.status(201); // 201 Created
    return id;
}
```

Inserção no BD



ComentarioDAO

```java
public boolean inserirComentario(Comentario comentario)
 {
     boolean status = false;
     try
     {
         this.maxId = (comentario.getId() > this.maxId) ? comentario.getId() : this.maxId;
         // Cria um statement para executar a inserção no banco de dados
         Statement st = conexao.createStatement();

         // Monta a consulta SQL para inserir um novo comentário
         String sql = "INSERT INTO comentario (id_comentario, conteudo, id_animal, id_pessoa) " +
                      "VALUES (" +
                      comentario.getId() + ", '" +
                      comentario.getConteudo() + "', '" +
                      comentario.getIdAnimal() + "', '" +
                      comentario.getIdPessoa() + "')";

         // Executa a consulta de inserção
         st.executeUpdate(sql);

         // Fecha o Statement após a execução
         st.close();

         // Define o status como true para indicar sucesso
         status = true;
     }
     catch (SQLException u)
     {
         // Caso ocorra uma exceção SQL, imprime o erro
         u.printStackTrace();
     }

     // Retorna o status da operação (true se inseriu, false se ocorreu um erro)
     return status;
 }
```

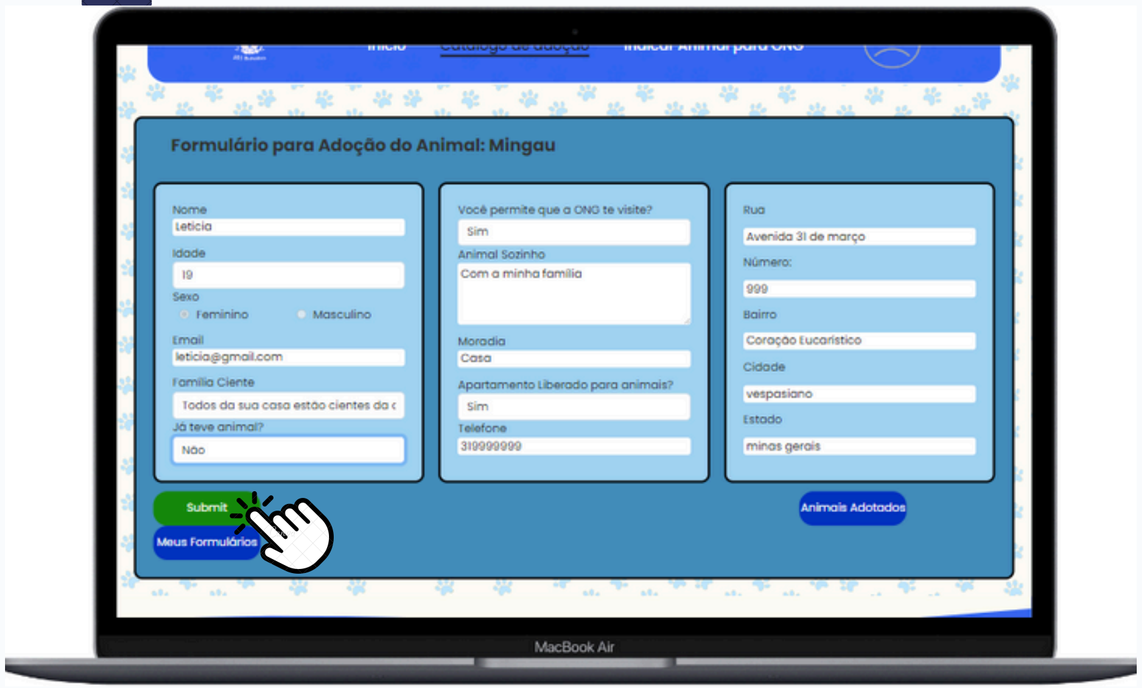


6 rows returned

	id_comentario integer	conteudo text	id_animal integer	id_pessoa integer
1	5	teste	2	3
2	4	ALTERA	2	3
3	7	novo	2	7
4	9	Legal!!!	2	9
5	11	Muito fofoo!!!	1	3
6	3	teste	2	3

Levando dados para o Back-End

Formulário



Javascript

```javascript
// Captura o botão de enviar
    const SubmeterFormBtn = document.querySelector(".botao_adocao");
    SubmeterFormBtn.addEventListener("click", async (event) => {
        event.preventDefault();
        // Lógica de envio do formulário
        ...

        let formulario = {
            ...
        };

        try {
            saveForm(formulario); //enviar formulario
            document.getElementById("formularioAdocao").reset(); // Limpa os campos após a submissão
        } catch (error) {
            console.error("Erro ao enviar o formulário:", error);
        }
    });

    // Função para salvar Formulário
    function saveForm(dado) {
        fetch(apiUrl_Formulario, {
            method: 'POST',
            headers: {
                'Content-Type': 'application/json',
            },
            body: JSON.stringify(dado)
        }).then(response => response.json())
          .then(dado => {
            alert("Formulário adicionado com sucesso");
          })
          .catch(error => {
              alert("Erro ao submeter formulário.");
          });
    }
```

HTML

```html
        <form id="formularioAdocao" class="formularios-container">
            <div class="formulario-box">
                ...
            </div>

            <div class="formulario-box">
                ...
            </div>

            <div class="formulario-box">
                ...
            </div>

        </form>
        <button type="submit" class="botao_adocao">Submit</button>
        <!-- Botão para abrir o modal de listagem -->
        <button type="button" class="botaoListarAnimais" id="openModalAnimais">Animais Adotados</button>
        <button type="button" class="botaoListar" id="openModalFormularios">Meus Formulários</button>
</div>
```

Recebendo dados no Back-End

Formulário

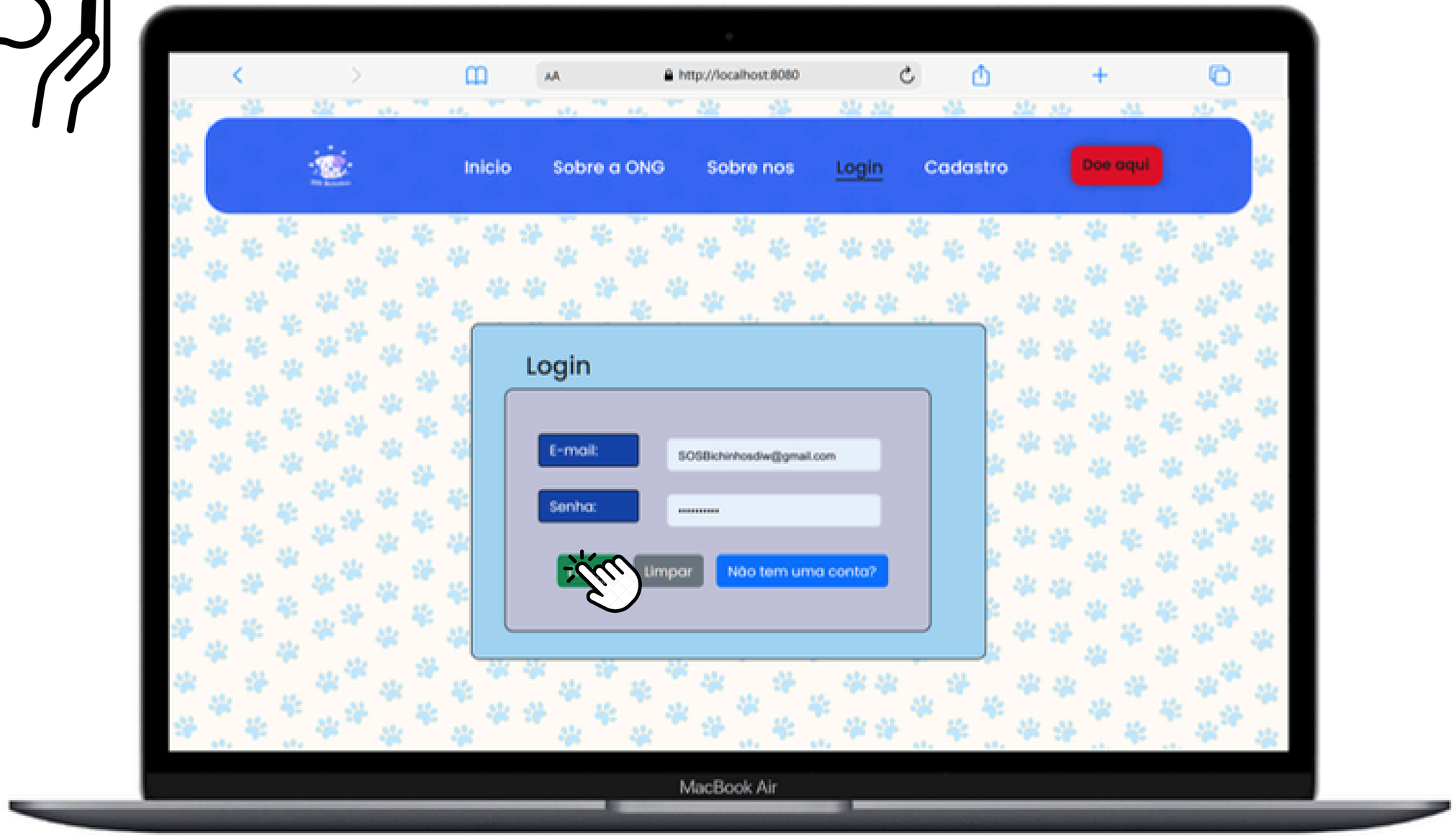
Aplicação

```
------------------------Aplicações CRUD Formulário  ---------------------------------------- //

    post("/formulario", (request, response) -> formularioService.insert(request, response));

    get("/formulario/:id", (request, response) -> formularioService.get(request, response));

    get("/formulario", (request, response) -> formularioService.getAll(request, response));

    post("/formulario/update/:id", (request, response) -> formularioService.update(request, response));

    post("/formulario/:id", (request, response) -> formularioService.delete(request, response));
```

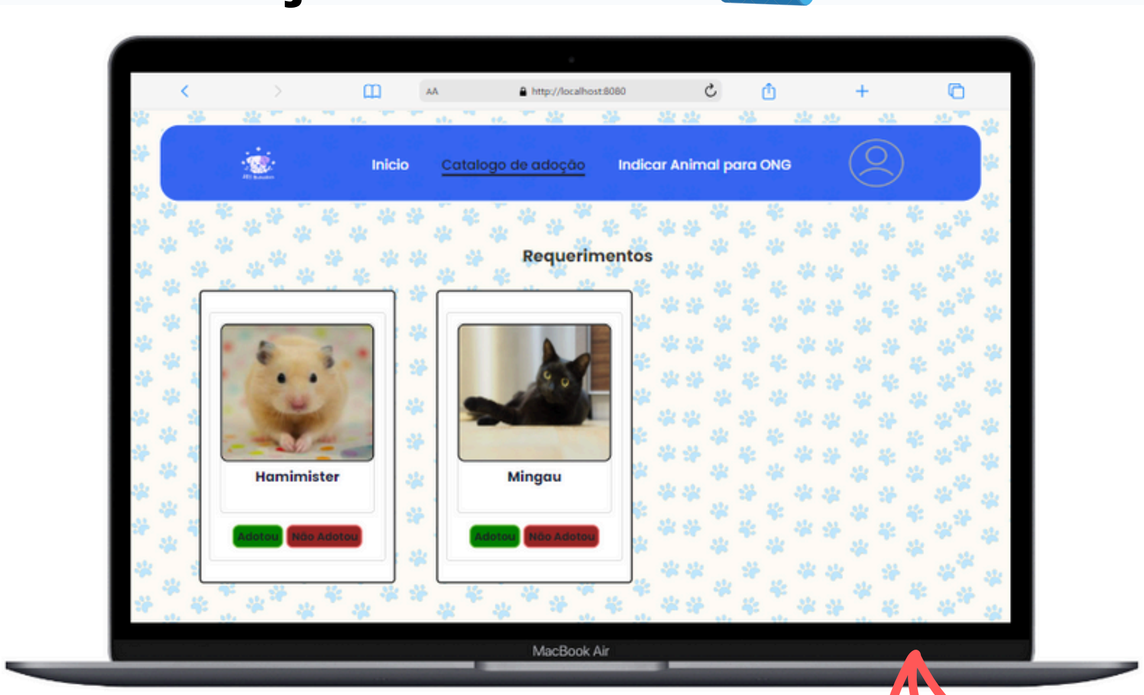
FormularioService

```
// Inserir Formulario
    public Object insert(Request request, Response response) {
        Gson gson = new Gson();

        Formulario registro = gson.fromJson(request.body(), Formulario.class);

        try {
            int id = this.formularioDAO.getMaxId() + 1;
            registro.setIdFormulario(id);
            System.out.println("Inserindo formulário: " + registro);
            formularioDAO.insert(registro);
            response.status(201); // 201 Created
            return id;
        } catch (Exception e) {
            System.out.println("Erro ao inserir no service: " + e.getMessage());
            return false;
        }
    }
```

Agora vamos para a ONG!



Inserção no BD



FormularioDAO

```java
// Inserir Formulario para adotar animal
    public boolean insert(Formulario formulario) {
        boolean status = false;
        try {
            this.maxId_form = (formulario.getIdFormulario() > this.maxId_form) ? formulario.getIdFormulario() : this.maxId_form;
            String sql = "INSERT INTO formulario (id_formulario, animal_sozinho, familia_ciente, permissao, teve_animal,
            id_animal, id_pessoa, telefone, ap_liberado) VALUES(?,?,?,?,?,?,?,?,?)";

            try (PreparedStatement stmt = conexao.prepareStatement(sql)) {
                stmt.setInt(1, formulario.getIdFormulario());
                stmt.setString(2, formulario.getAnimalSozinho());
                stmt.setBoolean(3, formulario.isFamiliaCiente());
                stmt.setBoolean(4, formulario.isPermissao());
                stmt.setBoolean(5, formulario.isTeveAnimal());
                stmt.setInt(6, formulario.getIdAnimal());
                stmt.setInt(7, formulario.getIdPessoa());
                stmt.setString(8, formulario.getTelefone());
                stmt.setBoolean(9, formulario.isApLiberado());

                stmt.executeUpdate();
            }
            status = true;
        } catch (SQLException u) {
            throw new RuntimeException(u);
        }
        return status;
    }
```

3 rows returned

	id_formulario integer	animal_sozinho character varying	familia_ciente boolean	permissao boolean	teve_animal boolean	id_animal integer	id_pessoa integer	telefone character varying	ap_liberado boolean
1	2	Com minha familia	true	true	true	4	8	77 99[illegible]4	true
2	3	Com amigos	true	true	true	1	8	7791548[illegible]2	true
3	4	Com minha familia	true	true	false	1	3	319999999	true

Levando dados para o Back-End

Cadastro de animal

HTML

```
<form id="animalForm" class="formularios-container">
  <div id="msg"></div>
  <!-- Adicione esta linha para exibir mensagens -->
  <div class="formulario-box">...</div>
  <div class="formulario-box">
    <!-- Historia -->
    <div class="form-group row align-items-center mb-3">
      <label for="historia-input-animal" class="HG_label_box col-sm-3
      Historia:</label>
      <div class="col-sm-9">
        <textarea type="text" class="form-control" id="historia-input-
        mal" name="historia" placeholder="Conte a historia desse anim
        required></textarea>
      </div>
    </div>
    <!-- Castrado -->
    <div class="form-group row align-items-center mb-3">...</div>
    <!-- Porte -->
    <div class="form-group row align-items-center mb-3">...</div>
    <!-- Tag 1 -->
    <div class="form-group row align-items-center mb-3">...</div>
    <!-- Tag 2 -->
    <div class="form-group row align-items-center mb-3">...</div>
    <!-- Tag 3 -->
    <div class="form-group row align-items-center mb-3">...</div>
  </div>
</form>
```

Javascript

```
/**
 * Manda para o BackEnd o objeto animal
 * @param {object} dado objeto a ser salvo no BackEnd
 */
async function saveDataAnimal(dado) {
  try {
      // Espera pelo resultado da requisição fetch
      const response = await fetch(apiUrlAnimal, {
          method: 'POST',
          headers: {
              'Content-Type': 'application/json',
          },
          body: JSON.stringify(dado)
      });

  const CadastrarAnimalBtn = document.querySelector(".botao-cadastra-animal");


  CadastrarAnimalBtn.addEventListener("click", async () => {

   //---------------------------------Animal---------------------------------//

   let id = await saveDataAnimal(Animal);
```

Recebendo dados no Back-End

Aplicação

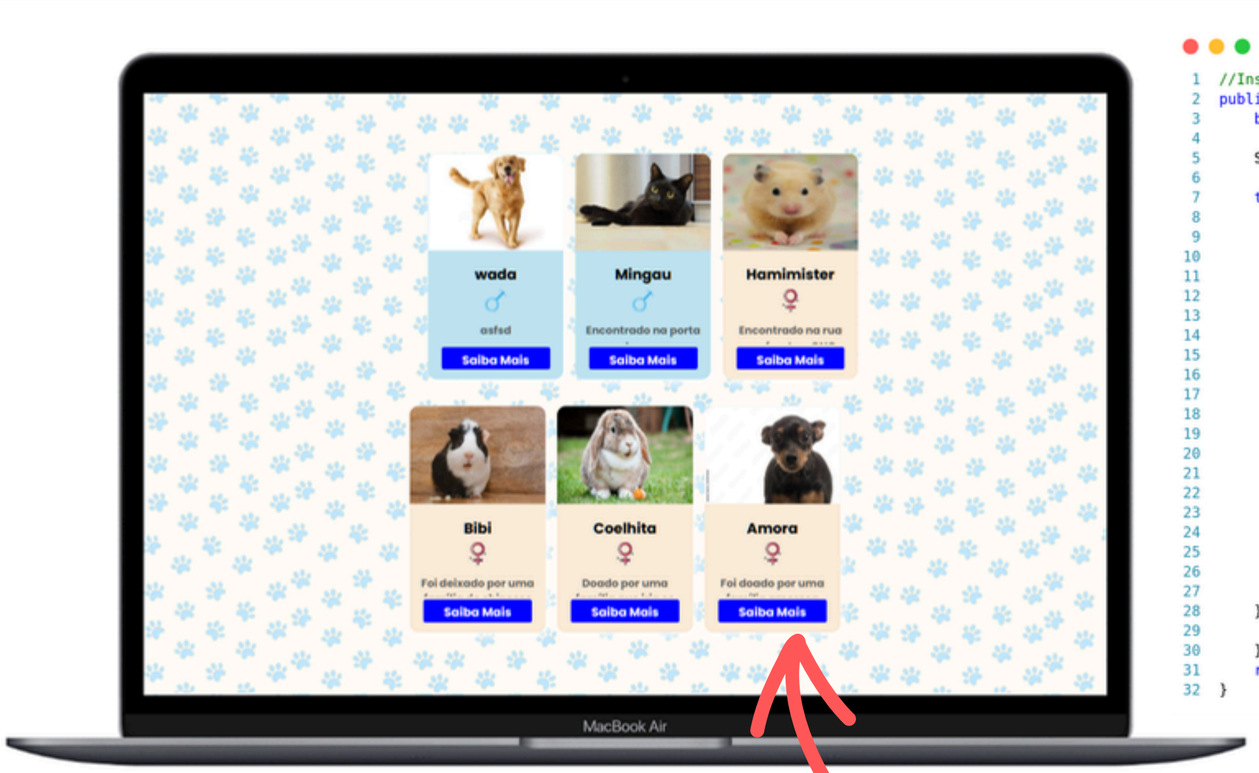
```
// ------------------------Aplicações CRUD Animal  ---------------------------------------- //


post("/animal", (request, response) -> animalService.add(request, response));

get("/animal/:id", (request, response) -> animalService.get(request, response));

post("/animal/update/:id", (request, response) -> animalService.update(request, response));

post("/animal/delete/:id", (request, response) -> animalService.remove(request, response));

get("/animal", (request, response) -> animalService.getAll(request, response));

```

AnimalService

```

public Object add(Request request, Response response) {
    Gson gson = new Gson();

    Animal registro = gson.fromJson(request.body(), Animal.class);

    int id = this.animalDAO.getMaxId() + 1;  // Corrigido o getMaxId

    registro.setId(id);

    animalDAO.inserirAnimal(registro);

    response.status(201); // 201 Created
    return id;
}
```

Inserção no BD

AnimalDAO

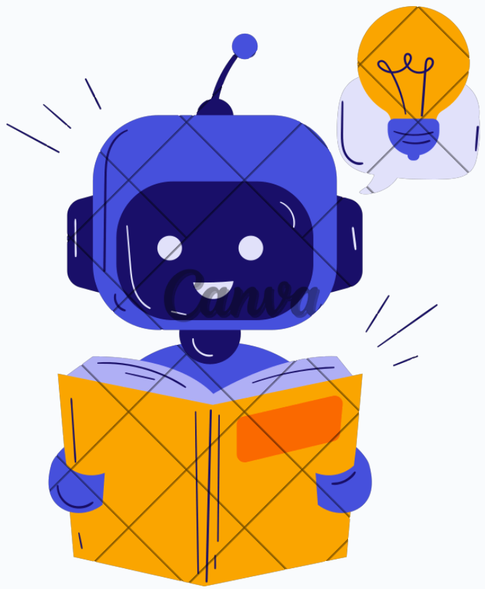



```java
//Inseri um animal no banco de dados
public boolean inserirAnimal(Animal animal){
    boolean status = false;

    System.out.println(animal);

    try {
        this.maxId = (animal.getId() > this.maxId) ? animal.getId() : this.maxId;
        Statement st = conexao.createStatement();
        String sql = "INSERT INTO animal (id_animal, imagem, nome, sexo, idade, raca, vacinas, castrado, historia, porte, especie) " +
                     "VALUES (" +
                     animal.getId() + ", '" +
                     animal.getImagem() + "', '" +
                     animal.getNome() + "', '" +
                     animal.getSexo() + "', '" +
                     animal.getIdade() + "', '" +
                     animal.getRaca() + "', '" +
                     animal.getVacinas() + "', " +
                     (animal.getCastrado() ? "TRUE" : "FALSE") + ", '" +  // Converte boolean para 'true' ou 'false'
                     animal.getHistoria() + "', '" +
                     animal.getPorte() + "', '" +
                     animal.getEspecie() + "')";

        //Executa o update com a variavel String sql
        st.executeUpdate(sql);
        st.close();
        status = true;
    } catch (SQLException u) {
        u.printStackTrace();
    }
    return status;
}
```

postgres
public
Functions
animal
id_animal : inte...
imagem : chara...
nome : characte...
sexo : character...
idade : characte...
castrado : boole...

	id_animal integer	imagem character varying	nome character varying	sexo bpchar	idade character varying	castrado boolean	raca character varying	vacinas character varying	historia text	porte bpchar	especie character varying
1	2	../../imgs/imagsInputs/Cachorro01.jpeg	wada	M	asfsd	false	asfsd	vacinas	asfsd	G	Cachorro
2	1	../../imgs/imagsInputs/GatoPreto01.jpg	Mingau	M	6 meses	false	gato preto	Astozoênica e puliga	Encontrado na porta de casa	P	Gato
3	3	../../imgs/imagsInputs/imageminexistente.png	Animal atualizado	M	18	true	animal teste de atualização	Não vacinado	Foi deixado	G	Coelho
4	4	https://i.ibb.co/PM6JYCt/hamster01.jpg	Hamimister	F	3 anos	false	Hamster holandês	Puliga	Encontrado na rua em frente a ONG	P	Roedor
5	5	https://i.ibb.co/5YzjnrZ/Bibi.jpg	Bibi	F	5 meses	false	Hamster Chinês	Não possui vacinas	Foi doado por uma familia de chineses para encontrar novo lar	P	Roedor
6	6	https://i.ibb.co/qn7zvVH/Coelho.png	Coelhita	F	1 ano	false	coelho lical	Não possui	Doado por uma familia que iria se mudar para um apartamento	M	Coelho
7	7	https://i.ibb.co/zQZPnzt/pincher.jpg	Amora	F	2 anos	true	pincher	Puliga e virose	Foi doado por uma familia amorosa	P	Cachorro

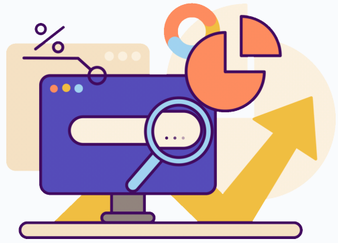
Sobre a Inteligência do Sistema

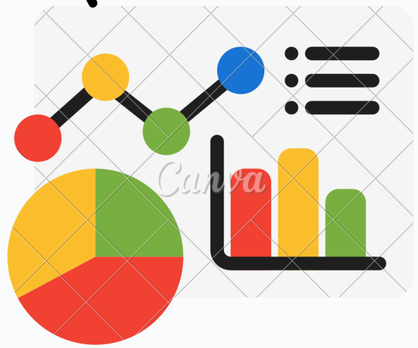


Gemini

Vision

Recursos e Técnicas





Características Gerais

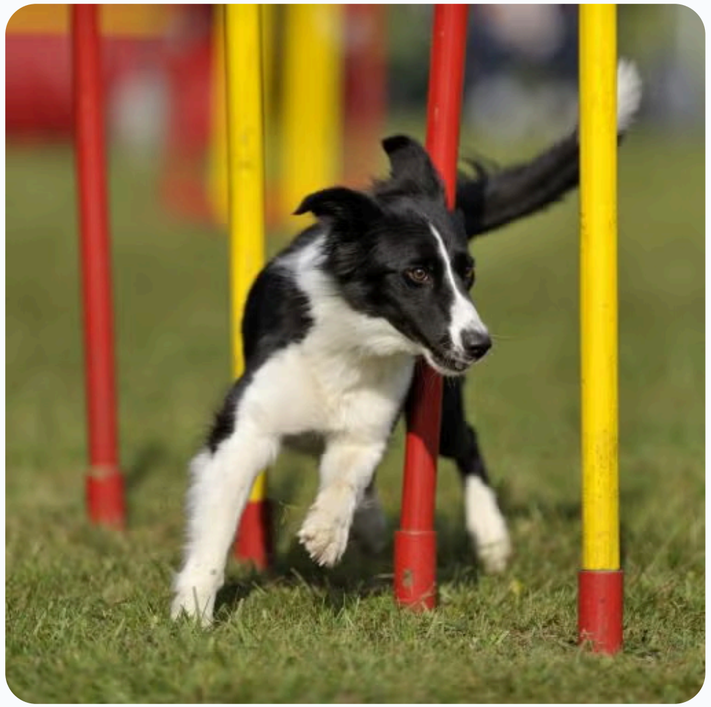
Possível idade

Possível temperamento

Especie

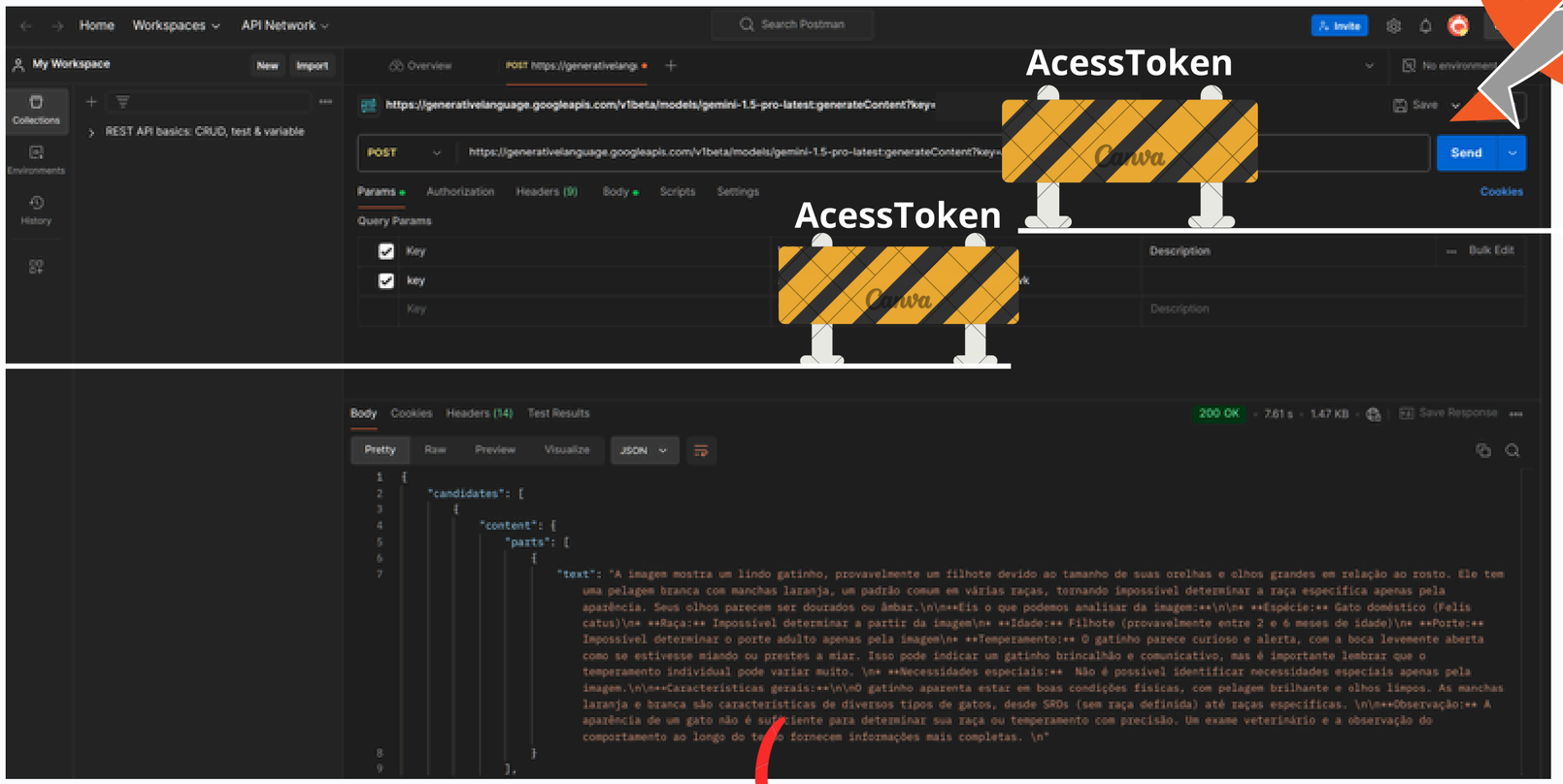
Raça

Valor Acrescentado pelo S.I



- **Agilidade**
- **Praticidade**
- **Comodidade**

Teste da Api no postman



Resposta

Aprendizado do Sistema & Entrada e Saída

TesteApi.java

AzureVisionService

GeminiVisionService

Gemini Vision

```
cliente@cliente-pc:~/Documentos/GitHub/plmg-cc-ti2-2024-2-g02-sosbichinhos/Codigo/SOSBichinhos$  /usr/bin/env MICRONAUT_CONFIG_FILES=/home/cliente/.confi
g/Code/User/workspaceStorage/b53d6819d6eca72522962695e1856054/asf.apache-netbeans-java/userdir/var/cache/nbls.db.connection/db-98171614826312736 6.propert
ies /home/cliente/apps/jdk22/bin/java @/tmp/cp_bczrpxhw6u8bi715ixk0kxg46.argfile app.TesteApi

-------------- Analise feita pela Azure --------------

{"categories":[{"name":"animal_gato","score":0.99609375}],"color":{"dominantColorForeground":"Grey","dominantColorBackground":"Grey","dominantColors":["G
rey","White"],"accentColor":"5F492F","isBwImg":false,"isBWImg":false},"tags":[{"name":"gato","confidence":0.9999920129776001},{"name":"sentando","confide
nce":0.9858494997024536},{"name":"gato doméstico","confidence":0.9745514392852783},{"name":"interior","confidence":0.9378886222839355},{"name":"branco","
confidence":0.9229899644851685},{"name":"animal","confidence":0.7531513571739197},{"name":"laranja","confidence":0.722207248210907},{"name":"carnívoro","
confidence":0.641948938369751}],"description":{"tags":["gato","mamífero","animal","no interior","laranja","olhando","mesa","pequeno","frente","em pé","ba
lcão","cinza","cozinha","empoleirado"],"captions":[{"text":"gato com a boca aberta","confidence":0.6803683412349936}]},"objects":[{"rectangle":{"x":22,"y
":54,"w":142,"h":220},"object":"cat","confidence":0.81,"parent":{"object":"mammal","confidence":0.859,"parent":{"object":"animal","confidence":0.859}}}],
"requestId":"44654c3e-331c-4c18-b8dd-730644c30869","metadata":{"height":275,"width":183,"format":"Jpeg"}}

-------------- Analise feita pela Gemini --------------

{
  "candidates": [
    {
      "content": {
        "parts": [
          {
            "text": "A imagem mostra um gato, provavelmente da **raça mestiça**, devido à sua pelagem que mistura laranja e branco. Pelas suas caracterís
ticas físicas, como o tamanho das orelhas e dos olhos, ele aparenta ser um **filhote**, com idade entre **2 e 4 meses**.\n\n**Não é possível determinar:*
*\n\n* **Porte:** a imagem não oferece referência de tamanho para saber se ele será um gato pequeno, médio ou grande quando adulto.\n* **Temperamento:**
 a expressão facial do gato pode indicar curiosidade, mas não é possível determinar seu temperamento apenas por uma foto.  \n* **Necessidades especiais:*
* a imagem não fornece informações suficientes para determinar se o gato possui alguma necessidade especial. \n\n**Características gerais:**\n\n* **Pelag
em:** curta, laranja e branca.\n* **Olhos:** grandes e expressivos.\n* **Postura:** Alerta e curioso.\n\nVale ressaltar que essas são apenas **suposições
** baseadas na observação da imagem. Para uma análise mais precisa, seria necessário um exame físico feito por um veterinário. \n"
          }
        ],
        "role": "model"
      }
```

Aprendizado offline

Resposta

+detalhada e precisa

Fornecedor do Sistema Inteligente

Ajude uma ONG, ajude uma vida, ajude um Animal!