



PUC Minas

Sprint 3

Membros da equipe:

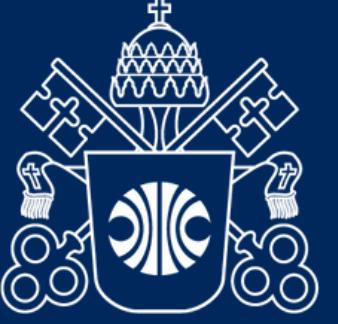
Erick Toshio

Leonardo Amaral

Gabriel Valedo

Henrique Giberti





PUC Minas



Patinhas felizes

CONEXÃO DE ONGS E ABRIGOS DE ANIMAIS

Patinhas
Felizes

Introduzindo

- Com o objetivo de mitigar o crescente número de animais abandonados no Brasil, nosso projeto conectará possíveis adotantes à ONG responsável por manter animais abandonados. Ele também visa facilitar o encontro e recolhimento de animais abandonados pela ONG por meio de reconhecimento de imagens por inteligência artificial.



Banco de Dados



```
SELECT * FROM ong;

CREATE TABLE ong
id VARCHAR(255) NOT NULL PRIMARY KEY,
nome VARCHAR(255) NOTNULL,
cnpj VARCHAR(255) NOTNULL,
endereco VARCHAR(255) NOTNULL,
telefone VARCHAR(255) NOTNULL,
email VARCHAR(255) NOTNULL,
senha VARCHAR(255) NOTNULL,
logo VARCHAR(511) NOTNULL,
historia VARCHAR(255) NOTNULL
);

CREATE TABLE usuarios(
id VARCHAR(255) NOT NULL PRIMARY KEY,
email VARCHAR(255) NOTNULL,
nome VARCHAR(255) NOTNULL,
senha VARCHAR(255) NOTNULL,
imagem VARCHAR(511),
etiquetas TEXT,
datanascimento DATE NOTNULL,
telefone VARCHAR(255) NOTNULL,
cpf VARCHAR(255) NOT NULL,
moradia VARCHAR(255) NOTNULL,
CONSTRAINT checarmoradia CHECK (moradia IN ('casa', 'apartamento'))
);
```



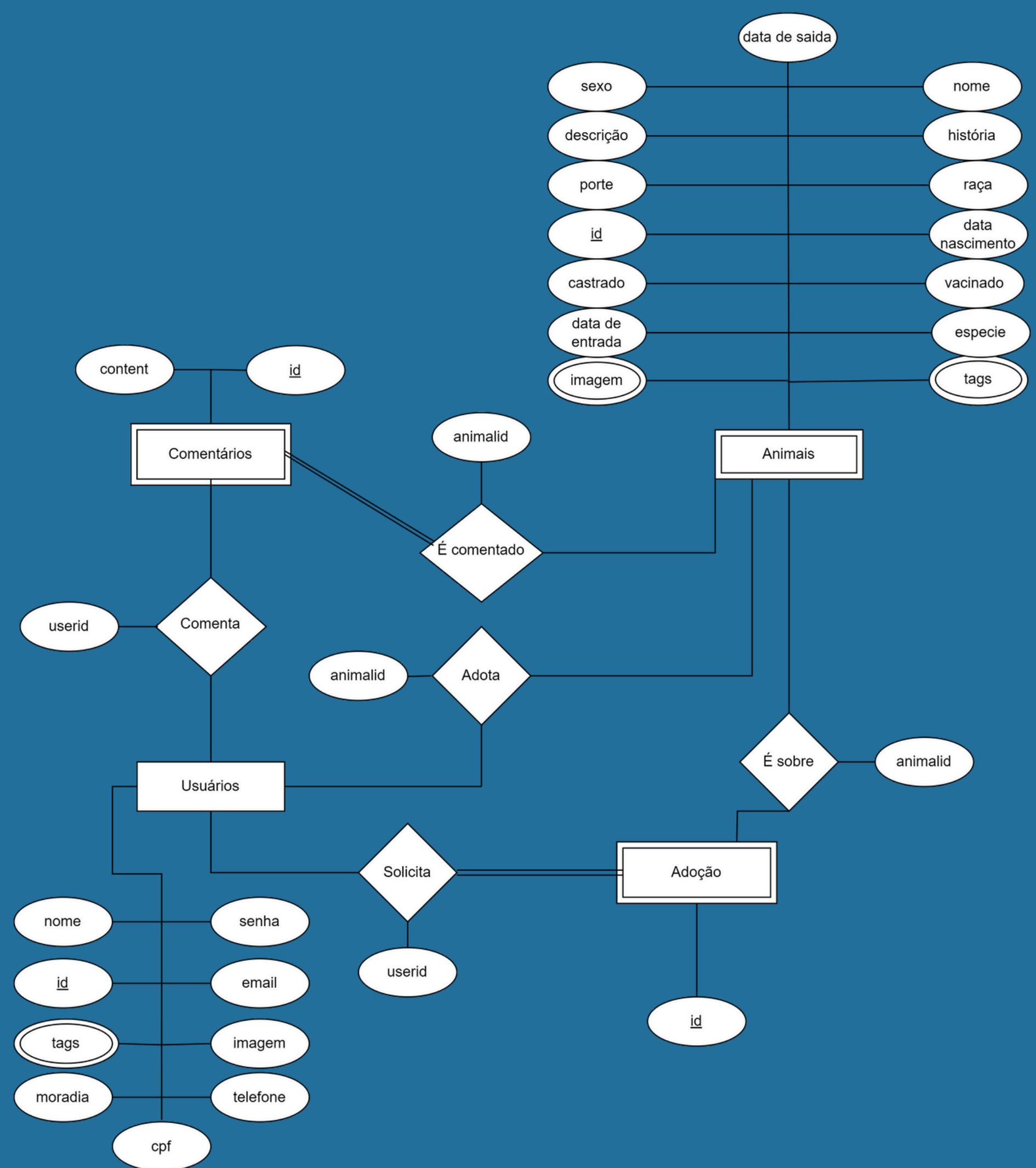
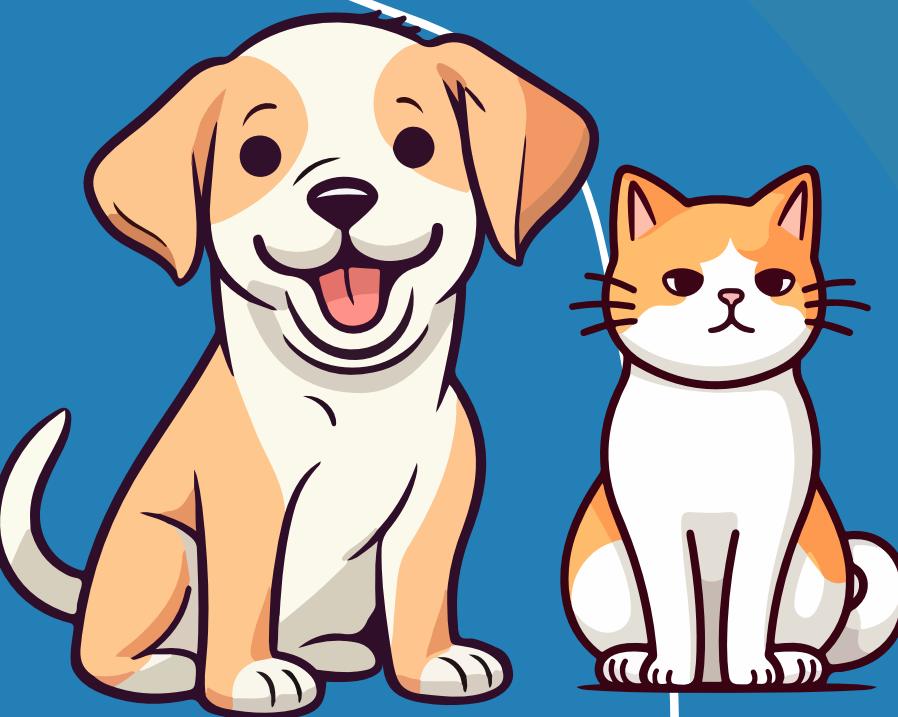
```
CREATE TABLE animais(
id VARCHAR(255) NOT NULL PRIMARY KEY,
nome VARCHAR(255) NOT NULL,
especie VARCHAR(255) NOT NULL,
porte VARCHAR(1) NOT NULL,
donoid VARCHAR(255),
imagem TEXT NOT NULL,
vacinas TEXT,
etiquetas TEXT NOT NULL,
datasaida DATE,
dataentrada DATE NOT NULL,
descricao TEXT NOT NULL,
historia TEXT NOTNULL,
raça VARCHAR(50) NOTNULL,
datanascimento DATE NOTNULL,
castrado BOOLEAN NOTNULL,
sexo VARCHAR(1) NOTNULL,
CONSTRAINT fkdonoanimal FOREIGN KEY(donoid) REFERENCES usuarios(id),
CONSTRAINT checarporte CHECK (porte IN ('P', 'M', 'G')),
CONSTRAINT checarsexo CHECK (sexo IN ('F', 'M'))
);
```

```
CREATE TABLE comentarios(
id VARCHAR(255) NOT NULL PRIMARY KEY,
usuarioid VARCHAR(255) NOT NULL,
animalid VARCHAR(255) NOT NULL,
conteudo VARCHAR(1023) NOT NULL,
CONSTRAINT fkcomentariouser FOREIGN KEY(usuarioid) REFERENCES usuarios(id),
CONSTRAINT fkcomentarioanimal FOREIGN KEY(animalid) REFERENCES animais(id)
);

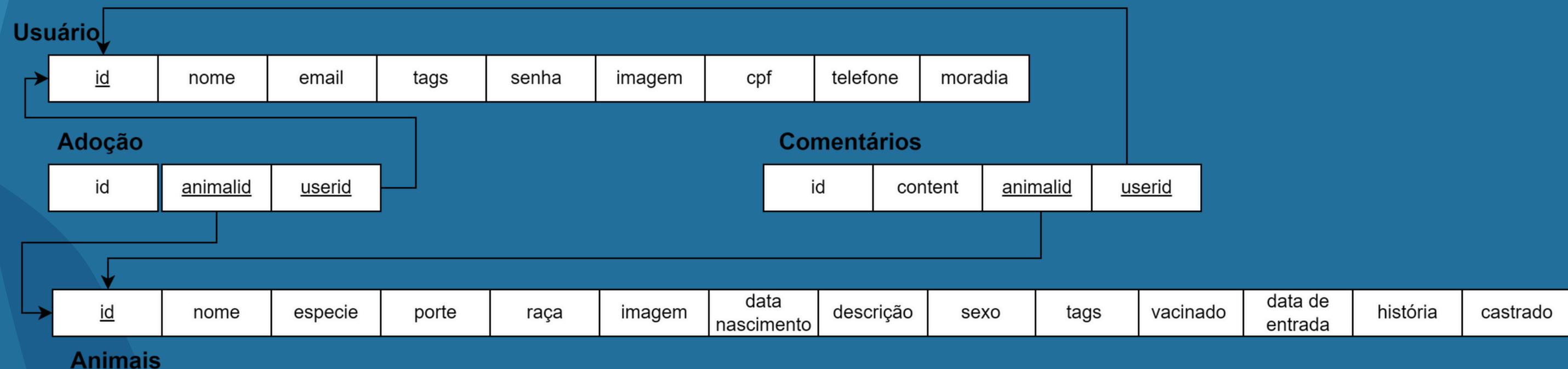
CREATE TABLE formularios(
id VARCHAR(255) NOT NULL PRIMARY KEY,
usuarioid VARCHAR(255) NOT NULL,
animalid VARCHAR(255) NOT NULL,
CONSTRAINT fkformularioanimal FOREIGN KEY(animalid) REFERENCES animais(id),
CONSTRAINT fkformulariouser FOREIGN KEY(usuarioid) REFERENCES usuarios(id)
);
```



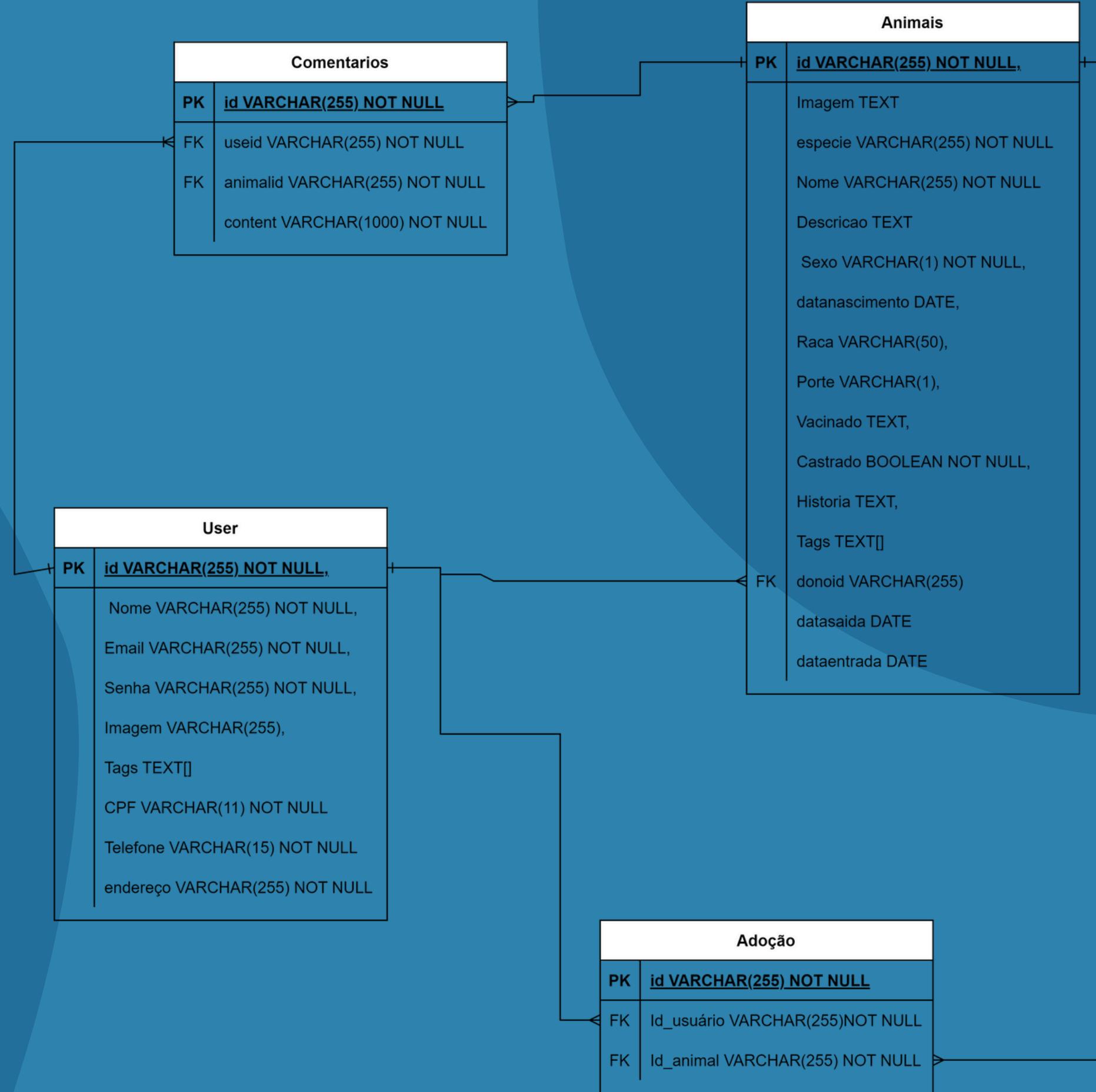
DER em notação Peter Chen



DSC em notação Elmasri & Navathe



DER em Pé de galinha



24 de out 21:56

Results: untitled-2 - plmg-cc-ti2-2024-2-g21-patinhasfelizes - Visual Studio Code

File Edit Selection View Go Run Terminal Help

SELECT * FROM animais;

	id	nome	especie	porte	donoid	imagem	vacinas	etiquetas	datasaida	dataentri
	character varying	text	text	text	date	date				
25	30	Simba	GA	M	null	https://s3.amazonaws.com/petz-cdm-stg/cf81a1e8-0d9f-4db8-b6db-224ed881cd1f.jpeg	Trivalente felina; Antirrábica	Extrovertido; Brincalhão; Gosta de caminhar	null	2022-12-0
26	1	Amora	CA	G	null	https://www.petz.com.br/blog/wp-content/uploads/2024/04/como-saber-se-o-cachorro-e-de-raca-interna2.jpg	Gripe canina; Glárdia; Antirrábica; Leishmaniose	Brincalhão; Gosta de caminhar	null	2024-07-2
27	33	Charlie	GA	M	null	https://s3.amazonaws.com/petz-cdm-stg/4b7d809d-8c80-4ff7-b7e3-03d1ad2bdb41.jpeg	Trivalente felina; Antirrábica	Extrovertido; Brincalhão; Gosta de caminhar	null	2023-01-2
28	7	Max	CA	G	null	https://th.bing.com/th/d/OIP.zlewrMx1krmznP1ltWGWHaHa?rs=1&pid=ImgDetMain	Gripe canina; Parvovírus; Antirrábica; Leptospirose	Brincalhão; Gosta de caminhar	null	2023-05-1

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

24 de out 21:57

Encontrei um Animal

Arquivo /home/gabriel/git/plmg-cc-ti2-2024-2-g21-patinhasfelizes/codigo/maven.patinhasfelizes/src/main/resources/front-end/cadastraranimal.html

Nome: Rufus

Entrada: 26/04/2024

Sexo: Macho Fêmea

Porte: Pequeno Médio Grande

Castrado: Sim Não

Espécie: Cachorro Coelho Roedor

Etiquetas: Brincalhão Calmo Caseiro Extrovertido

Raça: Rotwailer

Vacinas: Anti rabica, raiva

Descrição: Protetor e brincalhão

História: Rufus foi encontrado em situação de maus tratos, estava com medo, mas depois de muito carinho esta pronto para se divertir com uma nova família.

Enviar Limpar

© Patinhas Felizes
Conexão entre ONGs e Adotantes!

24 de out 21:57

cadastraranimal.html - maven.patinhasfelizes - Visual Studio Code

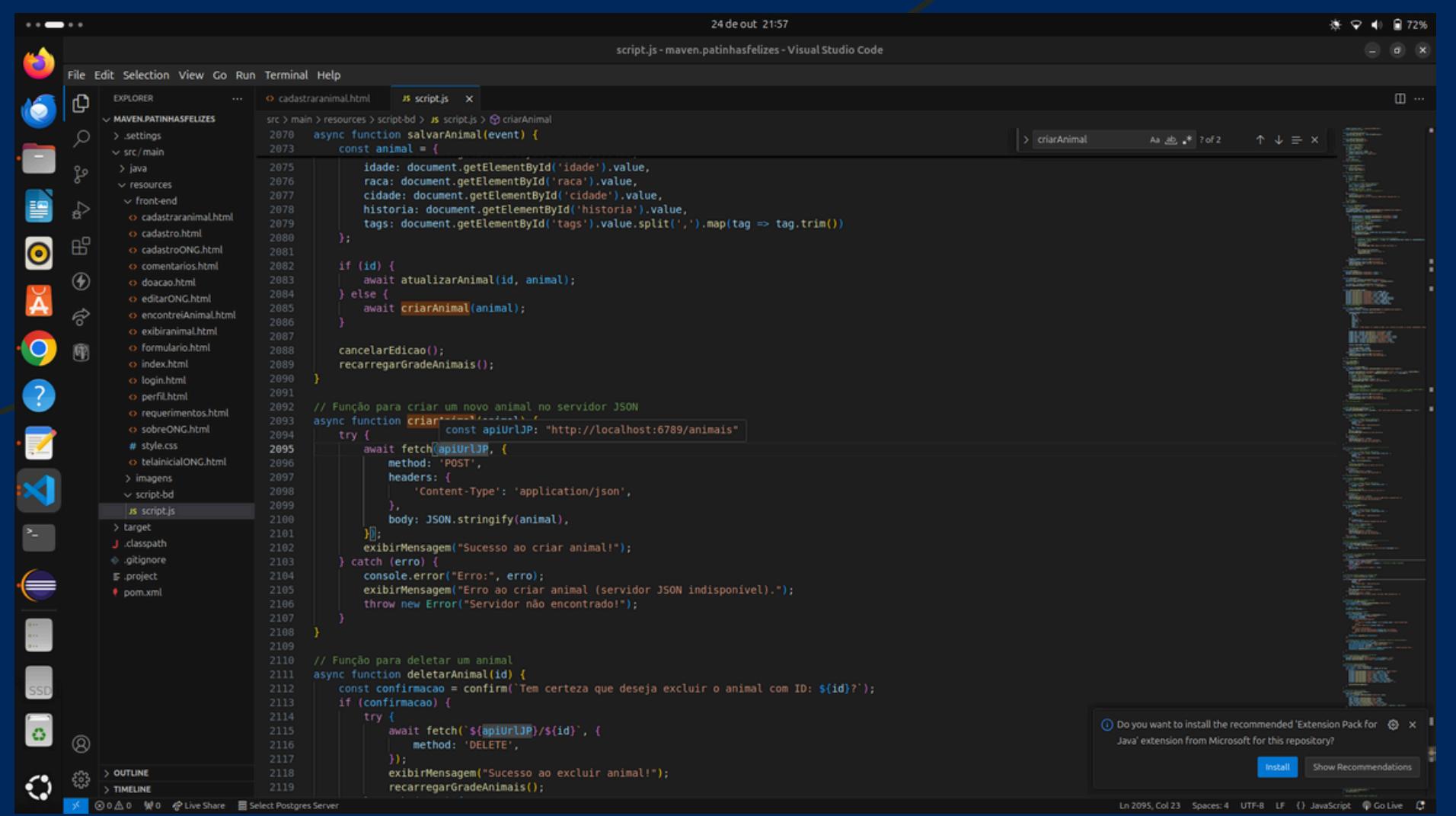
File Edit Selection View Go Run Terminal Help

cadastaranimal.html

```

src > main > resources > front-end > cadastraranimal.html > html > body > script > formulario_animal.addEventListener('submit') callback
13 <body>
14   <script src="..>
15     <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.8.3/jquery.min.js"></script>
16     <script src="..>
17       <script src="..>
18         const formulario_animal = document.getElementById('questionForm');
19         const GV_inputnascimento = document.querySelector('#inputNascimento');
20         const GV_inputentrada = document.querySelector('#inputEntrada');
21         const GV_imagemoststrar = document.getElementById('GV_imagemoststrar');
22         const GV_inputimgdoanimal = document.getElementById('inputImagem');
23         const GV_btnclear = document.getElementById('btnClear');
24
25         GV_inputimgdoanimal.addEventListener('blur', ()=>{
26           GV_imagemoststrar.setAttribute('src', GV_inputimgdoanimal.value);
27         });
28
29         formulario_animal.addEventListener('submit', async (event)=>{
30           let imagemcarregando = document.getElementById('GV_imagemdesitecarregando');
31           event.preventDefault();
32           try{
33             imagemcarregando.classList.remove("GV_esconder");
34             let GV_animal = new FormData(formulario_animal);
35             let GV_obj_animal = Object.fromEntries(GV_animal);
36             GV_obj_animal.etiquetas = new FormData(formulario_animal).getAll('etiquetas');
37             GV_obj_animal.id = Date.now().toString();
38             GV_obj_animal.donoid = null;
39             GV_obj_animal.datasaida = null;
40
41             await criarAnimal(GV_obj_animal);
42             console.log(GV_obj_animal);
43
44             window.location.href = "..>
45             imagemcarregando.classList.add("GV_esconder");
46           }catch(error){
47             imagemcarregando.classList.add("GV_esconder");
48             LCdisplayMessage('ERRO: Servidor não encontrado!');
49           }
50         });
51
52         function setaromaxdodata(campo){
53           let GV_inputdata = campo;
54           let GV_hoje = new Date();
55           let GV_diaatual = (GV_hoje.getDate()).toString().padStart(2, '0');
56           let GV_mesatual = (GV_hoje.getMonth() + 1).toString().padStart(2, '0');
57           let GV_anoatual = GV_hoje.getFullYear().toString();
58
59           GV_inputdata.value = `${GV_anoatual}-${GV_mesatual}-${GV_diaatual}`;
60         }
61
62         function setaromaxhora(campo){
63           let GV_inputhora = campo;
64           let GV_horaatual = new Date();
65           let GV_minutoatual = (GV_horaatual.getMinutes()).toString().padStart(2, '0');
66
67           GV_inputhora.value = `${GV_minutoatual}:00`;
68         }
69
70         function setaromaxdata(campo){
71           let GV_inputdata = campo;
72           let GV_hoje = new Date();
73           let GV_diaatual = (GV_hoje.getDate()).toString().padStart(2, '0');
74           let GV_mesatual = (GV_hoje.getMonth() + 1).toString().padStart(2, '0');
75           let GV_anoatual = GV_hoje.getFullYear().toString();
76
77           GV_inputdata.value = `${GV_anoatual}-${GV_mesatual}-${GV_diaatual}`;
78         }
79
80         function setaromaxhora(campo){
81           let GV_inputhora = campo;
82           let GV_horaatual = new Date();
83           let GV_minutoatual = (GV_horaatual.getMinutes()).toString().padStart(2, '0');
84
85           GV_inputhora.value = `${GV_minutoatual}:00`;
86         }
87
88         function setaromaxdata(campo){
89           let GV_inputdata = campo;
90           let GV_hoje = new Date();
91           let GV_diaatual = (GV_hoje.getDate()).toString().padStart(2, '0');
92           let GV_mesatual = (GV_hoje.getMonth() + 1).toString().padStart(2, '0');
93           let GV_anoatual = GV_hoje.getFullYear().toString();
94
95           GV_inputdata.value = `${GV_anoatual}-${GV_mesatual}-${GV_diaatual}`;
96         }
97
98         function setaromaxhora(campo){
99           let GV_inputhora = campo;
100          let GV_horaatual = new Date();
101          let GV_minutoatual = (GV_horaatual.getMinutes()).toString().padStart(2, '0');
102
103          GV_inputhora.value = `${GV_minutoatual}:00`;
104        }
105
106        function setaromaxdata(campo){
107          let GV_inputdata = campo;
108          let GV_hoje = new Date();
109          let GV_diaatual = (GV_hoje.getDate()).toString().padStart(2, '0');
110          let GV_mesatual = (GV_hoje.getMonth() + 1).toString().padStart(2, '0');
111          let GV_anoatual = GV_hoje.getFullYear().toString();
112
113          GV_inputdata.value = `${GV_anoatual}-${GV_mesatual}-${GV_diaatual}`;
114        }
115
116        function setaromaxhora(campo){
117          let GV_inputhora = campo;
118          let GV_horaatual = new Date();
119          let GV_minutoatual = (GV_horaatual.getMinutes()).toString().padStart(2, '0');
120
121          GV_inputhora.value = `${GV_minutoatual}:00`;
122        }
123
124        function setaromaxdata(campo){
125          let GV_inputdata = campo;
126          let GV_hoje = new Date();
127          let GV_diaatual = (GV_hoje.getDate()).toString().padStart(2, '0');
128          let GV_mesatual = (GV_hoje.getMonth() + 1).toString().padStart(2, '0');
129          let GV_anoatual = GV_hoje.getFullYear().toString();
130
131          GV_inputdata.value = `${GV_anoatual}-${GV_mesatual}-${GV_diaatual}`;
132        }
133
134        function setaromaxhora(campo){
135          let GV_inputhora = campo;
136          let GV_horaatual = new Date();
137          let GV_minutoatual = (GV_horaatual.getMinutes()).toString().padStart(2, '0');
138
139          GV_inputhora.value = `${GV_minutoatual}:00`;
140        }
141
142        function setaromaxdata(campo){
143          let GV_inputdata = campo;
144          let GV_hoje = new Date();
145          let GV_diaatual = (GV_hoje.getDate()).toString().padStart(2, '0');
146          let GV_mesatual = (GV_hoje.getMonth() + 1).toString().padStart(2, '0');
147          let GV_anoatual = GV_hoje.getFullYear().toString();
148
149          GV_inputdata.value = `${GV_anoatual}-${GV_mesatual}-${GV_diaatual}`;
150        }
151
152        function setaromaxhora(campo){
153          let GV_inputhora = campo;
154          let GV_horaatual = new Date();
155          let GV_minutoatual = (GV_horaatual.getMinutes()).toString().padStart(2, '0');
156
157          GV_inputhora.value = `${GV_minutoatual}:00`;
158        }
159
160        function setaromaxdata(campo){
161          let GV_inputdata = campo;
162          let GV_hoje = new Date();
163          let GV_diaatual = (GV_hoje.getDate()).toString().padStart(2, '0');
164          let GV_mesatual = (GV_hoje.getMonth() + 1).toString().padStart(2, '0');
165          let GV_anoatual = GV_hoje.getFullYear().toString();
166
167          GV_inputdata.value = `${GV_anoatual}-${GV_mesatual}-${GV_diaatual}`;
168        }
169
170        function setaromaxhora(campo){
171          let GV_inputhora = campo;
172          let GV_horaatual = new Date();
173          let GV_minutoatual = (GV_horaatual.getMinutes()).toString().padStart(2, '0');
174
175          GV_inputhora.value = `${GV_minutoatual}:00`;
176        }
177
178        function setaromaxdata(campo){
179          let GV_inputdata = campo;
180          let GV_hoje = new Date();
181          let GV_diaatual = (GV_hoje.getDate()).toString().padStart(2, '0');
182          let GV_mesatual = (GV_hoje.getMonth() + 1).toString().padStart(2, '0');
183          let GV_anoatual = GV_hoje.getFullYear().toString();
184
185          GV_inputdata.value = `${GV_anoatual}-${GV_mesatual}-${GV_diaatual}`;
186        }
187
188        function setaromaxhora(campo){
189          let GV_inputhora = campo;
190          let GV_horaatual = new Date();
191          let GV_minutoatual = (GV_horaatual.getMinutes()).toString().padStart(2, '0');
192
193          GV_inputhora.value = `${GV_minutoatual}:00`;
194        }
195
196        function setaromaxdata(campo){
197          let GV_inputdata = campo;
198          let GV_hoje = new Date();
199          let GV_diaatual = (GV_hoje.getDate()).toString().padStart(2, '0');
200          let GV_mesatual = (GV_hoje.getMonth() + 1).toString().padStart(2, '0');
201          let GV_anoatual = GV_hoje.getFullYear().toString();
202
203          GV_inputdata.value = `${GV_anoatual}-${GV_mesatual}-${GV_diaatual}`;
204        }
205
206        function setaromaxhora(campo){
207          let GV_inputhora = campo;
208          let GV_horaatual = new Date();
209          let GV_minutoatual = (GV_horaatual.getMinutes()).toString().padStart(2, '0');
210
211          GV_inputhora.value = `${GV_minutoatual}:00`;
212        }
213
214        function setaromaxdata(campo){
215          let GV_inputdata = campo;
216          let GV_hoje = new Date();
217          let GV_diaatual = (GV_hoje.getDate()).toString().padStart(2, '0');
218          let GV_mesatual = (GV_hoje.getMonth() + 1).toString().padStart(2, '0');
219          let GV_anoatual = GV_hoje.getFullYear().toString();
220
221          GV_inputdata.value = `${GV_anoatual}-${GV_mesatual}-${GV_diaatual}`;
222        }
223
224        function setaromaxhora(campo){
225          let GV_inputhora = campo;
226          let GV_horaatual = new Date();
227          let GV_minutoatual = (GV_horaatual.getMinutes()).toString().padStart(2, '0');
228
229          GV_inputhora.value = `${GV_minutoatual}:00`;
230        }
231
232        function setaromaxdata(campo){
233          let GV_inputdata = campo;
234          let GV_hoje = new Date();
235          let GV_diaatual = (GV_hoje.getDate()).toString().padStart(2, '0');
236          let GV_mesatual = (GV_hoje.getMonth() + 1).toString().padStart(2, '0');
237          let GV_anoatual = GV_hoje.getFullYear().toString();
238
239          GV_inputdata.value = `${GV_anoatual}-${GV_mesatual}-${GV_diaatual}`;
240        }
241
242        function setaromaxhora(campo){
243          let GV_inputhora = campo;
244          let GV_horaatual = new Date();
245          let GV_minutoatual = (GV_horaatual.getMinutes()).toString().padStart(2, '0');
246
247          GV_inputhora.value = `${GV_minutoatual}:00`;
248        }
249
250        function setaromaxdata(campo){
251          let GV_inputdata = campo;
252          let GV_hoje = new Date();
253          let GV_diaatual = (GV_hoje.getDate()).toString().padStart(2, '0');
254          let GV_mesatual = (GV_hoje.getMonth() + 1).toString().padStart(2, '0');
255          let GV_anoatual = GV_hoje.getFullYear().toString();
256
257          GV_inputdata.value = `${GV_anoatual}-${GV_mesatual}-${GV_diaatual}`;
258        }
259
260        function setaromaxhora(campo){
261          let GV_inputhora = campo;
262          let GV_horaatual = new Date();
263          let GV_minutoatual = (GV_horaatual.getMinutes()).toString().padStart(2, '0');
264
265          GV_inputhora.value = `${GV_minutoatual}:00`;
266        }
267
268        function setaromaxdata(campo){
269          let GV_inputdata = campo;
270          let GV_hoje = new Date();
271          let GV_diaatual = (GV_hoje.getDate()).toString().padStart(2, '0');
272          let GV_mesatual = (GV_hoje.getMonth() + 1).toString().padStart(2, '0');
273          let GV_anoatual = GV_hoje.getFullYear().toString();
274
275          GV_inputdata.value = `${GV_anoatual}-${GV_mesatual}-${GV_diaatual}`;
276        }
277
278        function setaromaxhora(campo){
279          let GV_inputhora = campo;
280          let GV_horaatual = new Date();
281          let GV_minutoatual = (GV_horaatual.getMinutes()).toString().padStart(2, '0');
282
283          GV_inputhora.value = `${GV_minutoatual}:00`;
284        }
285
286        function setaromaxdata(campo){
287          let GV_inputdata = campo;
288          let GV_hoje = new Date();
289          let GV_diaatual = (GV_hoje.getDate()).toString().padStart(2, '0');
290          let GV_mesatual = (GV_hoje.getMonth() + 1).toString().padStart(2, '0');
291          let GV_anoatual = GV_hoje.getFullYear().toString();
292
293          GV_inputdata.value = `${GV_anoatual}-${GV_mesatual}-${GV_diaatual}`;
294        }
295
296        function setaromaxhora(campo){
297          let GV_inputhora = campo;
298          let GV_horaatual = new Date();
299          let GV_minutoatual = (GV_horaatual.getMinutes()).toString().padStart(2, '0');
300
301          GV_inputhora.value = `${GV_minutoatual}:00`;
302        }
303
304        function setaromaxdata(campo){
305          let GV_inputdata = campo;
306          let GV_hoje = new Date();
307          let GV_diaatual = (GV_hoje.getDate()).toString().padStart(2, '0');
308          let GV_mesatual = (GV_hoje.getMonth() + 1).toString().padStart(2, '0');
309          let GV_anoatual = GV_hoje.getFullYear().toString();
310
311          GV_inputdata.value = `${GV_anoatual}-${GV_mesatual}-${GV_diaatual}`;
312        }
313
314        function setaromaxhora(campo){
315          let GV_inputhora = campo;
316          let GV_horaatual = new Date();
317          let GV_minutoatual = (GV_horaatual.getMinutes()).toString().padStart(2, '0');
318
319          GV_inputhora.value = `${GV_minutoatual}:00`;
320        }
321
322        function setaromaxdata(campo){
323          let GV_inputdata = campo;
324          let GV_hoje = new Date();
325          let GV_diaatual = (GV_hoje.getDate()).toString().padStart(2, '0');
326          let GV_mesatual = (GV_hoje.getMonth() + 1).toString().padStart(2, '0');
327          let GV_anoatual = GV_hoje.getFullYear().toString();
328
329          GV_inputdata.value = `${GV_anoatual}-${GV_mesatual}-${GV_diaatual}`;
330        }
331
332        function setaromaxhora(campo){
333          let GV_inputhora = campo;
334          let GV_horaatual = new Date();
335          let GV_minutoatual = (GV_horaatual.getMinutes()).toString().padStart(2, '0');
336
337          GV_inputhora.value = `${GV_minutoatual}:00`;
338        }
339
340        function setaromaxdata(campo){
341          let GV_inputdata = campo;
342          let GV_hoje = new Date();
343          let GV_diaatual = (GV_hoje.getDate()).toString().padStart(2, '0');
344          let GV_mesatual = (GV_hoje.getMonth() + 1).toString().padStart(2, '0');
345          let GV_anoatual = GV_hoje.getFullYear().toString();
346
347          GV_inputdata.value = `${GV_anoatual}-${GV_mesatual}-${GV_diaatual}`;
348        }
349
350        function setaromaxhora(campo){
351          let GV_inputhora = campo;
352          let GV_horaatual = new Date();
353          let GV_minutoatual = (GV_horaatual.getMinutes()).toString().padStart(2, '0');
354
355          GV_inputhora.value = `${GV_minutoatual}:00`;
356        }
357
358        function setaromaxdata(campo){
359          let GV_inputdata = campo;
360          let GV_hoje = new Date();
361          let GV_diaatual = (GV_hoje.getDate()).toString().padStart(2, '0');
362          let GV_mesatual = (GV_hoje.getMonth() + 1).toString().padStart(2, '0');
363          let GV_anoatual = GV_hoje.getFullYear().toString();
364
365          GV_inputdata.value = `${GV_anoatual}-${GV_mesatual}-${GV_diaatual}`;
366        }
367
368        function setaromaxhora(campo){
369          let GV_inputhora = campo;
370          let GV_horaatual = new Date();
371          let GV_minutoatual = (GV_horaatual.getMinutes()).toString().padStart(2, '0');
372
373          GV_inputhora.value = `${GV_minutoatual}:00`;
374        }
375
376        function setaromaxdata(campo){
377          let GV_inputdata = campo;
378          let GV_hoje = new Date();
379          let GV_diaatual = (GV_hoje.getDate()).toString().padStart(2, '0');
380          let GV_mesatual = (GV_hoje.getMonth() + 1).toString().padStart(2, '0');
381          let GV_anoatual = GV_hoje.getFullYear().toString();
382
383          GV_inputdata.value = `${GV_anoatual}-${GV_mesatual}-${GV_diaatual}`;
384        }
385
386        function setaromaxhora(campo){
387          let GV_inputhora = campo;
388          let GV_horaatual = new Date();
389          let GV_minutoatual = (GV_horaatual.getMinutes()).toString().padStart(2, '0');
390
391          GV_inputhora.value = `${GV_minutoatual}:00`;
392        }
393
394        function setaromaxdata(campo){
395          let GV_inputdata = campo;
396          let GV_hoje = new Date();
397          let GV_diaatual = (GV_hoje.getDate()).toString().padStart(2, '0');
398          let GV_mesatual = (GV_hoje.getMonth() + 1).toString().padStart(2, '0');
399          let GV_anoatual = GV_hoje.getFullYear().toString();
400
401          GV_inputdata.value = `${GV_anoatual}-${GV_mesatual}-${GV_diaatual}`;
402        }
403
404        function setaromaxhora(campo){
405          let GV_inputhora = campo;
406          let GV_horaatual = new Date();
407          let GV_minutoatual = (GV_horaatual.getMinutes()).toString().padStart(2, '0');
408
409          GV_inputhora.value = `${GV_minutoatual}:00`;
410        }
411
412        function setaromaxdata(campo){
413          let GV_inputdata = campo;
414          let GV_hoje = new Date();
415          let GV_diaatual = (GV_hoje.getDate()).toString().padStart(2, '0');
416          let GV_mesatual = (GV_hoje.getMonth() + 1).toString().padStart(2, '0');
417          let GV_anoatual = GV_hoje.getFullYear().toString();
418
419          GV_inputdata.value = `${GV_anoatual}-${GV_mesatual}-${GV_diaatual}`;
420        }
421
422        function setaromaxhora(campo){
423          let GV_inputhora = campo;
424          let GV_horaatual = new Date();
425          let GV_minutoatual = (GV_horaatual.getMinutes()).toString().padStart(2, '0');
426
427          GV_inputhora.value = `${GV_minutoatual}:00`;
428        }
429
430        function setaromaxdata(campo){
431          let GV_inputdata = campo;
432          let GV_hoje = new Date();
433          let GV_diaatual = (GV_hoje.getDate()).toString().padStart(2, '0');
434          let GV_mesatual = (GV_hoje.getMonth() + 1).toString().padStart(2, '0');
435          let GV_anoatual = GV_hoje.getFullYear().toString();
436
437          GV_inputdata.value = `${GV_anoatual}-${GV_mesatual}-${GV_diaatual}`;
438        }
439
440        function setaromaxhora(campo){
441          let GV_inputhora = campo;
442          let GV_horaatual = new Date();
443          let GV_minutoatual = (GV_horaatual.getMinutes()).toString().padStart(2, '0');
444
445          GV_inputhora.value = `${GV_minutoatual}:00`;
446        }
447
448        function setaromaxdata(campo){
449          let GV_inputdata = campo;
450          let GV_hoje = new Date();
451          let GV_diaatual = (GV_hoje.getDate()).toString().padStart(2, '0');
452          let GV_mesatual = (GV_hoje.getMonth() + 1).toString().padStart(2, '0');
453          let GV_anoatual = GV_hoje.getFullYear().toString();
454
455          GV_inputdata.value = `${GV_anoatual}-${GV_mesatual}-${GV_diaatual}`;
456        }
457
458        function setaromaxhora(campo){
459          let GV_inputhora = campo;
460          let GV_horaatual = new Date();
461          let GV_minutoatual = (GV_horaatual.getMinutes()).toString().padStart(2, '0');
462
463          GV_inputhora.value = `${GV_minutoatual}:00`;
464        }
465
466        function setaromaxdata(campo){
467          let GV_inputdata = campo;

```



24 de out 21:57

script.js - maven.patinhasfelizes - Visual Studio Code

```
File Edit Selection View Go Run Terminal Help
```

EXPLORER

MAVEN.PATINHASFELIZES

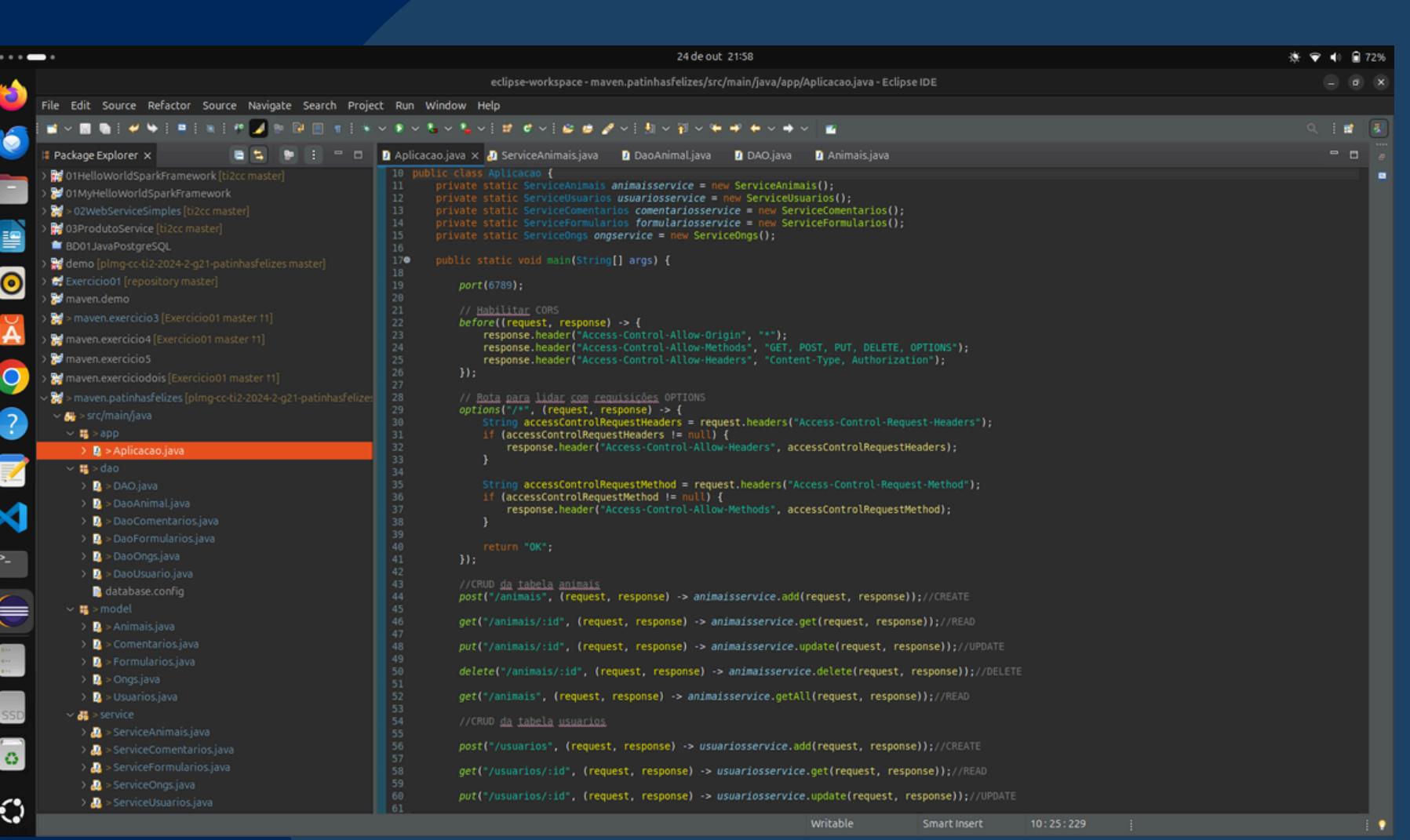
src > main > resources > script-bd > js script.js > criarAnimal

```
2070 	async function salvarAnimal(event) {
2071    const animal = {
2072      idade: document.getElementById('idade').value,
2073      raca: document.getElementById('raca').value,
2074      cidade: document.getElementById('cidade').value,
2075      historia: document.getElementById('historia').value,
2076      tags: document.getElementById('tags').value.split(',') .map(tag => tag.trim())
2077    };
2078
2079    if (id) {
2080      await atualizarAnimal(id, animal);
2081    } else {
2082      await criarAnimal(animal);
2083    }
2084
2085    cancelarEdicao();
2086
2087    recarregarGradeAnimais();
2088
2089  }
2090
2091
2092 // Função para criar um novo animal no servidor JSON
2093 async function criarAnimal() {
2094   const apiUrlJP: "http://localhost:6789/animais"
2095   try {
2096     await fetch(apiUrlJP, {
2097       method: 'POST',
2098       headers: {
2099         'Content-Type': 'application/json',
2100       },
2101       body: JSON.stringify(animal),
2102     });
2103     exibirMensagem("Sucesso ao criar animal!");
2104   } catch (erro) {
2105     console.error("Erro:", erro);
2106     exibirMensagem("Erro ao criar animal (servidor JSON indisponível).");
2107   }
2108 }
2109
2110 // Função para deletar um animal
2111 async function deletarAnimal(id) {
2112   const confirmacao = confirm('Tem certeza que deseja excluir o animal com ID: ${id}?');
2113
2114   if (confirmacao) {
2115     try {
2116       await fetch(`$apiUrlJP/${id}`, {
2117         method: 'DELETE',
2118       });
2119     exibirMensagem("Sucesso ao excluir animal!");
2120   }
2121   recarregarGradeAnimais();
2122 }
```

Do you want to install the recommended 'Extension Pack for Java' extension from Microsoft for this repository?

Install Show Recommendations

Ln 2095, Col 23 Spaces: 4 UTF-8 LF () JavaScript Go Live



24 de out 21:58

eclipse-workspace - maven.patinhasfelizes/src/main/java/app/Aplicacao.java - Eclipse IDE

```
File Edit Source Refactor Source Navigate Search Project Run Window Help
```

Package Explorer

Aplicacao.java

```
10 public class Aplicacao {
11   private static ServiceAnimais animaiservice = new ServiceAnimais();
12   private static ServiceUsuarios usuariosservice = new ServiceUsuarios();
13   private static ServiceComentarios comentariosservice = new ServiceComentarios();
14   private static ServiceFormularios formulariosservice = new ServiceFormularios();
15   private static ServiceOngs ongservice = new ServiceOngs();
16
17   public static void main(String[] args) {
18
19     port(6789);
20
21     // Habilitar CORS
22     before((request, response) -> {
23       response.header("Access-Control-Allow-Origin", "*");
24       response.header("Access-Control-Allow-Methods", "GET, POST, PUT, DELETE, OPTIONS");
25       response.header("Access-Control-Allow-Headers", "Content-Type, Authorization");
26     });
27
28     // Rota para lidar com requisições OPTIONS
29     options("*", (request, response) -> {
30       String accessControlRequestHeaders = request.headers("Access-Control-Request-Headers");
31       if (accessControlRequestHeaders != null) {
32         response.header("Access-Control-Allow-Headers", accessControlRequestHeaders);
33       }
34
35       String accessControlRequestMethod = request.headers("Access-Control-Request-Method");
36       if (accessControlRequestMethod != null) {
37         response.header("Access-Control-Allow-Methods", accessControlRequestMethod);
38       }
39
40       return "OK";
41     });
42
43     //CRUD da tabela animais
44     post("/animais", (request, response) -> animaiservice.add(request, response));//CREATE
45     get("/animais/:id", (request, response) -> animaiservice.get(request, response));//READ
46     put("/animais/:id", (request, response) -> animaiservice.update(request, response));//UPDATE
47     delete("/animais/:id", (request, response) -> animaiservice.delete(request, response));//DELETE
48
49     get("/animais", (request, response) -> animaiservice.getAll(request, response));//READ
50
51     //CRUD da tabela usuarios
52     post("/usuarios", (request, response) -> usuariosservice.add(request, response));//CREATE
53     get("/usuarios/:id", (request, response) -> usuariosservice.get(request, response));//READ
54     put("/usuarios/:id", (request, response) -> usuariosservice.update(request, response));//UPDATE
55     delete("/usuarios/:id", (request, response) -> usuariosservice.delete(request, response));//DELETE
56
57     //CRUD da tabela ongs
58     post("/ongs", (request, response) -> ongservice.add(request, response));//CREATE
59     get("/ongs/:id", (request, response) -> ongservice.get(request, response));//READ
60     put("/ongs/:id", (request, response) -> ongservice.update(request, response));//UPDATE
61     delete("/ongs/:id", (request, response) -> ongservice.delete(request, response));//DELETE
62
63     //CRUD da tabela comentarios
64     post("/comentarios", (request, response) -> comentariosservice.add(request, response));//CREATE
65     get("/comentarios/:id", (request, response) -> comentariosservice.get(request, response));//READ
66     put("/comentarios/:id", (request, response) -> comentariosservice.update(request, response));//UPDATE
67     delete("/comentarios/:id", (request, response) -> comentariosservice.delete(request, response));//DELETE
68
69     //CRUD da tabela formularios
70     post("/formularios", (request, response) -> formulariosservice.add(request, response));//CREATE
71     get("/formularios/:id", (request, response) -> formulariosservice.get(request, response));//READ
72     put("/formularios/:id", (request, response) -> formulariosservice.update(request, response));//UPDATE
73     delete("/formularios/:id", (request, response) -> formulariosservice.delete(request, response));//DELETE
74
75     //CRUD da tabela usuarios
76     post("/usuarios", (request, response) -> usuariosservice.add(request, response));//CREATE
77     get("/usuarios/:id", (request, response) -> usuariosservice.get(request, response));//READ
78     put("/usuarios/:id", (request, response) -> usuariosservice.update(request, response));//UPDATE
79     delete("/usuarios/:id", (request, response) -> usuariosservice.delete(request, response));//DELETE
80
81     //CRUD da tabela ongs
82     post("/ongs", (request, response) -> ongservice.add(request, response));//CREATE
83     get("/ongs/:id", (request, response) -> ongservice.get(request, response));//READ
84     put("/ongs/:id", (request, response) -> ongservice.update(request, response));//UPDATE
85     delete("/ongs/:id", (request, response) -> ongservice.delete(request, response));//DELETE
86
87     //CRUD da tabela animais
88     post("/animais", (request, response) -> animaiservice.add(request, response));//CREATE
89     get("/animais/:id", (request, response) -> animaiservice.get(request, response));//READ
90     put("/animais/:id", (request, response) -> animaiservice.update(request, response));//UPDATE
91     delete("/animais/:id", (request, response) -> animaiservice.delete(request, response));//DELETE
92
93     //CRUD da tabela comentarios
94     post("/comentarios", (request, response) -> comentariosservice.add(request, response));//CREATE
95     get("/comentarios/:id", (request, response) -> comentariosservice.get(request, response));//READ
96     put("/comentarios/:id", (request, response) -> comentariosservice.update(request, response));//UPDATE
97     delete("/comentarios/:id", (request, response) -> comentariosservice.delete(request, response));//DELETE
98
99     //CRUD da tabela formularios
100    post("/formularios", (request, response) -> formulariosservice.add(request, response));//CREATE
101    get("/formularios/:id", (request, response) -> formulariosservice.get(request, response));//READ
102    put("/formularios/:id", (request, response) -> formulariosservice.update(request, response));//UPDATE
103    delete("/formularios/:id", (request, response) -> formulariosservice.delete(request, response));//DELETE
104
105    //CRUD da tabela ongs
106    post("/ongs", (request, response) -> ongservice.add(request, response));//CREATE
107    get("/ongs/:id", (request, response) -> ongservice.get(request, response));//READ
108    put("/ongs/:id", (request, response) -> ongservice.update(request, response));//UPDATE
109    delete("/ongs/:id", (request, response) -> ongservice.delete(request, response));//DELETE
110
111    //CRUD da tabela animais
112    post("/animais", (request, response) -> animaiservice.add(request, response));//CREATE
113    get("/animais/:id", (request, response) -> animaiservice.get(request, response));//READ
114    put("/animais/:id", (request, response) -> animaiservice.update(request, response));//UPDATE
115    delete("/animais/:id", (request, response) -> animaiservice.delete(request, response));//DELETE
116
117    //CRUD da tabela comentarios
118    post("/comentarios", (request, response) -> comentariosservice.add(request, response));//CREATE
119    get("/comentarios/:id", (request, response) -> comentariosservice.get(request, response));//READ
120    put("/comentarios/:id", (request, response) -> comentariosservice.update(request, response));//UPDATE
121    delete("/comentarios/:id", (request, response) -> comentariosservice.delete(request, response));//DELETE
122
123    //CRUD da tabela formularios
124    post("/formularios", (request, response) -> formulariosservice.add(request, response));//CREATE
125    get("/formularios/:id", (request, response) -> formulariosservice.get(request, response));//READ
126    put("/formularios/:id", (request, response) -> formulariosservice.update(request, response));//UPDATE
127    delete("/formularios/:id", (request, response) -> formulariosservice.delete(request, response));//DELETE
128
129    //CRUD da tabela ongs
130    post("/ongs", (request, response) -> ongservice.add(request, response));//CREATE
131    get("/ongs/:id", (request, response) -> ongservice.get(request, response));//READ
132    put("/ongs/:id", (request, response) -> ongservice.update(request, response));//UPDATE
133    delete("/ongs/:id", (request, response) -> ongservice.delete(request, response));//DELETE
134
135    //CRUD da tabela animais
136    post("/animais", (request, response) -> animaiservice.add(request, response));//CREATE
137    get("/animais/:id", (request, response) -> animaiservice.get(request, response));//READ
138    put("/animais/:id", (request, response) -> animaiservice.update(request, response));//UPDATE
139    delete("/animais/:id", (request, response) -> animaiservice.delete(request, response));//DELETE
140
141    //CRUD da tabela comentarios
142    post("/comentarios", (request, response) -> comentariosservice.add(request, response));//CREATE
143    get("/comentarios/:id", (request, response) -> comentariosservice.get(request, response));//READ
144    put("/comentarios/:id", (request, response) -> comentariosservice.update(request, response));//UPDATE
145    delete("/comentarios/:id", (request, response) -> comentariosservice.delete(request, response));//DELETE
146
147    //CRUD da tabela formularios
148    post("/formularios", (request, response) -> formulariosservice.add(request, response));//CREATE
149    get("/formularios/:id", (request, response) -> formulariosservice.get(request, response));//READ
150    put("/formularios/:id", (request, response) -> formulariosservice.update(request, response));//UPDATE
151    delete("/formularios/:id", (request, response) -> formulariosservice.delete(request, response));//DELETE
152
153    //CRUD da tabela ongs
154    post("/ongs", (request, response) -> ongservice.add(request, response));//CREATE
155    get("/ongs/:id", (request, response) -> ongservice.get(request, response));//READ
156    put("/ongs/:id", (request, response) -> ongservice.update(request, response));//UPDATE
157    delete("/ongs/:id", (request, response) -> ongservice.delete(request, response));//DELETE
158
159    //CRUD da tabela animais
160    post("/animais", (request, response) -> animaiservice.add(request, response));//CREATE
161    get("/animais/:id", (request, response) -> animaiservice.get(request, response));//READ
162    put("/animais/:id", (request, response) -> animaiservice.update(request, response));//UPDATE
163    delete("/animais/:id", (request, response) -> animaiservice.delete(request, response));//DELETE
164
165    //CRUD da tabela comentarios
166    post("/comentarios", (request, response) -> comentariosservice.add(request, response));//CREATE
167    get("/comentarios/:id", (request, response) -> comentariosservice.get(request, response));//READ
168    put("/comentarios/:id", (request, response) -> comentariosservice.update(request, response));//UPDATE
169    delete("/comentarios/:id", (request, response) -> comentariosservice.delete(request, response));//DELETE
170
171    //CRUD da tabela formularios
172    post("/formularios", (request, response) -> formulariosservice.add(request, response));//CREATE
173    get("/formularios/:id", (request, response) -> formulariosservice.get(request, response));//READ
174    put("/formularios/:id", (request, response) -> formulariosservice.update(request, response));//UPDATE
175    delete("/formularios/:id", (request, response) -> formulariosservice.delete(request, response));//DELETE
176
177    //CRUD da tabela ongs
178    post("/ongs", (request, response) -> ongservice.add(request, response));//CREATE
179    get("/ongs/:id", (request, response) -> ongservice.get(request, response));//READ
180    put("/ongs/:id", (request, response) -> ongservice.update(request, response));//UPDATE
181    delete("/ongs/:id", (request, response) -> ongservice.delete(request, response));//DELETE
182
183    //CRUD da tabela animais
184    post("/animais", (request, response) -> animaiservice.add(request, response));//CREATE
185    get("/animais/:id", (request, response) -> animaiservice.get(request, response));//READ
186    put("/animais/:id", (request, response) -> animaiservice.update(request, response));//UPDATE
187    delete("/animais/:id", (request, response) -> animaiservice.delete(request, response));//DELETE
188
189    //CRUD da tabela comentarios
190    post("/comentarios", (request, response) -> comentariosservice.add(request, response));//CREATE
191    get("/comentarios/:id", (request, response) -> comentariosservice.get(request, response));//READ
192    put("/comentarios/:id", (request, response) -> comentariosservice.update(request, response));//UPDATE
193    delete("/comentarios/:id", (request, response) -> comentariosservice.delete(request, response));//DELETE
194
195    //CRUD da tabela formularios
196    post("/formularios", (request, response) -> formulariosservice.add(request, response));//CREATE
197    get("/formularios/:id", (request, response) -> formulariosservice.get(request, response));//READ
198    put("/formularios/:id", (request, response) -> formulariosservice.update(request, response));//UPDATE
199    delete("/formularios/:id", (request, response) -> formulariosservice.delete(request, response));//DELETE
200
201    //CRUD da tabela ongs
202    post("/ongs", (request, response) -> ongservice.add(request, response));//CREATE
203    get("/ongs/:id", (request, response) -> ongservice.get(request, response));//READ
204    put("/ongs/:id", (request, response) -> ongservice.update(request, response));//UPDATE
205    delete("/ongs/:id", (request, response) -> ongservice.delete(request, response));//DELETE
206
207    //CRUD da tabela animais
208    post("/animais", (request, response) -> animaiservice.add(request, response));//CREATE
209    get("/animais/:id", (request, response) -> animaiservice.get(request, response));//READ
210    put("/animais/:id", (request, response) -> animaiservice.update(request, response));//UPDATE
211    delete("/animais/:id", (request, response) -> animaiservice.delete(request, response));//DELETE
212
213    //CRUD da tabela comentarios
214    post("/comentarios", (request, response) -> comentariosservice.add(request, response));//CREATE
215    get("/comentarios/:id", (request, response) -> comentariosservice.get(request, response));//READ
216    put("/comentarios/:id", (request, response) -> comentariosservice.update(request, response));//UPDATE
217    delete("/comentarios/:id", (request, response) -> comentariosservice.delete(request, response));//DELETE
218
219    //CRUD da tabela formularios
220    post("/formularios", (request, response) -> formulariosservice.add(request, response));//CREATE
221    get("/formularios/:id", (request, response) -> formulariosservice.get(request, response));//READ
222    put("/formularios/:id", (request, response) -> formulariosservice.update(request, response));//UPDATE
223    delete("/formularios/:id", (request, response) -> formulariosservice.delete(request, response));//DELETE
224
225    //CRUD da tabela ongs
226    post("/ongs", (request, response) -> ongservice.add(request, response));//CREATE
227    get("/ongs/:id", (request, response) -> ongservice.get(request, response));//READ
228    put("/ongs/:id", (request, response) -> ongservice.update(request, response));//UPDATE
229    delete("/ongs/:id", (request, response) -> ongservice.delete(request, response));//DELETE
230
231    //CRUD da tabela animais
232    post("/animais", (request, response) -> animaiservice.add(request, response));//CREATE
233    get("/animais/:id", (request, response) -> animaiservice.get(request, response));//READ
234    put("/animais/:id", (request, response) -> animaiservice.update(request, response));//UPDATE
235    delete("/animais/:id", (request, response) -> animaiservice.delete(request, response));//DELETE
236
237    //CRUD da tabela comentarios
238    post("/comentarios", (request, response) -> comentariosservice.add(request, response));//CREATE
239    get("/comentarios/:id", (request, response) -> comentariosservice.get(request, response));//READ
240    put("/comentarios/:id", (request, response) -> comentariosservice.update(request, response));//UPDATE
241    delete("/comentarios/:id", (request, response) -> comentariosservice.delete(request, response));//DELETE
242
243    //CRUD da tabela formularios
244    post("/formularios", (request, response) -> formulariosservice.add(request, response));//CREATE
245    get("/formularios/:id", (request, response) -> formulariosservice.get(request, response));//READ
246    put("/formularios/:id", (request, response) -> formulariosservice.update(request, response));//UPDATE
247    delete("/formularios/:id", (request, response) -> formulariosservice.delete(request, response));//DELETE
248
249    //CRUD da tabela ongs
250    post("/ongs", (request, response) -> ongservice.add(request, response));//CREATE
251    get("/ongs/:id", (request, response) -> ongservice.get(request, response));//READ
252    put("/ongs/:id", (request, response) -> ongservice.update(request, response));//UPDATE
253    delete("/ongs/:id", (request, response) -> ongservice.delete(request, response));//DELETE
254
255    //CRUD da tabela animais
256    post("/animais", (request, response) -> animaiservice.add(request, response));//CREATE
257    get("/animais/:id", (request, response) -> animaiservice.get(request, response));//READ
258    put("/animais/:id", (request, response) -> animaiservice.update(request, response));//UPDATE
259    delete("/animais/:id", (request, response) -> animaiservice.delete(request, response));//DELETE
260
261    //CRUD da tabela comentarios
262    post("/comentarios", (request, response) -> comentariosservice.add(request, response));//CREATE
263    get("/comentarios/:id", (request, response) -> comentariosservice.get(request, response));//READ
264    put("/comentarios/:id", (request, response) -> comentariosservice.update(request, response));//UPDATE
265    delete("/comentarios/:id", (request, response) -> comentariosservice.delete(request, response));//DELETE
266
267    //CRUD da tabela formularios
268    post("/formularios", (request, response) -> formulariosservice.add(request, response));//CREATE
269    get("/formularios/:id", (request, response) -> formulariosservice.get(request, response));//READ
270    put("/formularios/:id", (request, response) -> formulariosservice.update(request, response));//UPDATE
271    delete("/formularios/:id", (request, response) -> formulariosservice.delete(request, response));//DELETE
272
273    //CRUD da tabela ongs
274    post("/ongs", (request, response) -> ongservice.add(request, response));//CREATE
275    get("/ongs/:id", (request, response) -> ongservice.get(request, response));//READ
276    put("/ongs/:id", (request, response) -> ongservice.update(request, response));//UPDATE
277    delete("/ongs/:id", (request, response) -> ongservice.delete(request, response));//DELETE
278
279    //CRUD da tabela animais
280    post("/animais", (request, response) -> animaiservice.add(request, response));//CREATE
281    get("/animais/:id", (request, response) -> animaiservice.get(request, response));//READ
282    put("/animais/:id", (request, response) -> animaiservice.update(request, response));//UPDATE
283    delete("/animais/:id", (request, response) -> animaiservice.delete(request, response));//DELETE
284
285    //CRUD da tabela comentarios
286    post("/comentarios", (request, response) -> comentariosservice.add(request, response));//CREATE
287    get("/comentarios/:id", (request, response) -> comentariosservice.get(request, response));//READ
288    put("/comentarios/:id", (request, response) -> comentariosservice.update(request, response));//UPDATE
289    delete("/comentarios/:id", (request, response) -> comentariosservice.delete(request, response));//DELETE
290
291    //CRUD da tabela formularios
292    post("/formularios", (request, response) -> formulariosservice.add(request, response));//CREATE
293    get("/formularios/:id", (request, response) -> formulariosservice.get(request, response));//READ
294    put("/formularios/:id", (request, response) -> formulariosservice.update(request, response));//UPDATE
295    delete("/formularios/:id", (request, response) -> formulariosservice.delete(request, response));//DELETE
296
297    //CRUD da tabela ongs
298    post("/ongs", (request, response) -> ongservice.add(request, response));//CREATE
299    get("/ongs/:id", (request, response) -> ongservice.get(request, response));//READ
300    put("/ongs/:id", (request, response) -> ongservice.update(request, response));//UPDATE
301    delete("/ongs/:id", (request, response) -> ongservice.delete(request, response));//DELETE
302
303    //CRUD da tabela animais
304    post("/animais", (request, response) -> animaiservice.add(request, response));//CREATE
305    get("/animais/:id", (request, response) -> animaiservice.get(request, response));//READ
306    put("/animais/:id", (request, response) -> animaiservice.update(request, response));//UPDATE
307    delete("/animais/:id", (request, response) -> animaiservice.delete(request, response));//DELETE
308
309    //CRUD da tabela comentarios
310    post("/comentarios", (request, response) -> comentariosservice.add(request, response));//CREATE
311    get("/comentarios/:id", (request, response) -> comentariosservice.get(request, response));//READ
312    put("/comentarios/:id", (request, response) -> comentariosservice.update(request, response));//UPDATE
313    delete("/comentarios/:id", (request, response) -> comentariosservice.delete(request, response));//DELETE
314
315    //CRUD da tabela formularios
316    post("/formularios", (request, response) -> formulariosservice.add(request, response));//CREATE
317    get("/formularios/:id", (request, response) -> formulariosservice.get(request, response));//READ
318    put("/formularios/:id", (request, response) -> formulariosservice.update(request, response));//UPDATE
319    delete("/formularios/:id", (request, response) -> formulariosservice.delete(request, response));//DELETE
320
321    //CRUD da tabela ongs
322    post("/ongs", (request, response) -> ongservice.add(request, response));//CREATE
323    get("/ongs/:id", (request, response) -> ongservice.get(request, response));//READ
324    put("/ongs/:id", (request, response) -&
```

24 de out 21:58

eclipse-workspace - maven.patinhasfelizes/src/main/java/dao/DaoAnimal.java - Eclipse IDE

```

File Edit Source Refactor Source Navigate Search Project Run Window Help

Package Explorer X
  01HelloWorldSparkFramework [ti2cc master]
  02WebServiceSimples [ti2cc master]
  03ProdutoService [ti2cc master]
  BD01JavaPostgreSQL
  demo [plmg-cc-ti2-2024-2-g21-patinhasfelizes master]
  Exercicio01 [repository master]
  maven.demo
  maven.exercicio3 [Exercicio01 master 11]
  maven.exercicio4 [Exercicio01 master 11]
  maven.exercicio5
  maven.exercicioodois [Exercicio01 master 11]
  maven.patinhasfelizes [plmg-cc-ti2-2024-2-g21-patinhasfelizes master]
  src/main/java
    app
      Aplicacao.java
    dao
      DaoAnimal.java
      DAO.java
      Animais.java
    model
      Animais.java
      Comentarios.java
      Formularios.java
      Ongs.java
      Usuarios.java
    service
      ServiceAnimais.java
      ServiceComentarios.java
      ServiceFormularios.java
      ServiceOngs.java
      ServiceUsuarios.java
  database.config

  package dao;
  import java.sql.ResultSet;
  public class DaoAnimal extends DAO{
    //Metodo para inserir um objeto do tipo Animais na tabela animais do banco de dados postgresql
    public boolean inserirAnimais(Animais animalInserir) {
      //Varáveis
      boolean status = false;
      //Conectar com o servidor sql
      conectar();
      try {
        //Criar objeto do tipo Statement para realizar consultas no banco de dados
        Statement st = conexao.createStatement();
        //Executar determinada linha de código em sql para inserir os atributos do objeto do tipo Animais na tabela animais no banco de dados
        st.executeUpdate("INSERT INTO animais (id, nome, imagem, especie, raca, vacinas, datanascimento, dataentrada, sexo, porte, castrado, etiquetas, descricao) VALUES ('" + animalInserir.getId() + "','" + animalInserir.getNome() + "','" + animalInserir.getImagen() + "','" + animalInserir.getEspecie() + "','" + animalInserir.getRaca() + "','" + animalInserir.getVaccinas() + "','" + animalInserir.getDataNascimento() + "','" + animalInserir.getDataEntrada() + "','" + animalInserir.getSexo() + "','" + animalInserir.getPorte() + "','" + animalInserir.isCastrado() + "','" + String.join(", ", animalInserir.getEtiquetas()) + "','" + animalInserir.getDescricao() + "','" + animalInserir.getHistoria() + "','" + animalInserir.getDonoid() + null ? "','" + animalInserir.getDonoid() + "','" + animalInserir.getDataSaida() + "')");
      } catch (SQLException u) {
        throw new RuntimeException(u);
      } finally{
        //Encerrar a conexão com o banco de dados
        close();
      }
      return status;
    }
    //Metodo para atualizar os valores de um item da tabela animais para os atributos do objeto do tipo Animais passado como parametros
    public boolean atualizaranimais(Animais animalAtualizar) {
      //Varáveis
      boolean status = false;
      //Conectar com o servidor sql
      conectar();
      try {
        //Criar objeto do tipo Statement para realizar consultas no banco de dados
        Statement st = conexao.createStatement();
        //Criar uma linha de código em sql para atualizar um item da tabela animais no banco de dados para os atributos do objeto do tipo Animais
        st.executeUpdate("UPDATE animais SET nome = '" + animalAtualizar.getNome() + "', especie = '" + animalAtualizar.getEspecie() + "', raca = '" + animalAtualizar.getRaca() + "', vacinas = '" + animalAtualizar.getVaccinas() + "', dataascimento = '" + animalAtualizar.getDataNascimento() + "', dataentrada = '" + animalAtualizar.getDataEntrada() + "', sexo = '" + animalAtualizar.getSexo() + "', porte = '" + animalAtualizar.getPorte() + "', castrado = '" + animalAtualizar.isCastrado() + "', etiquetas = '" + String.join(", ", animalAtualizar.getEtiquetas()) + "', descricao = '" + animalAtualizar.getDescricao() + "', historia = '" + animalAtualizar.getHistoria() + "' WHERE id = " + animalAtualizar.getId());
      } catch (SQLException u) {
        throw new RuntimeException(u);
      } finally{
        //Encerrar a conexão com o banco de dados
        close();
      }
      return status;
    }
  }

```

Writable Smart Insert 1:1:0

24 de out 21:58

eclipse-workspace - maven.patinhasfelizes/src/main/java/dao/DAO.java - Eclipse IDE

```

File Edit Source Refactor Source Navigate Search Project Run Window Help

Package Explorer X
  01HelloWorldSparkFramework [ti2cc master]
  02WebServiceSimples [ti2cc master]
  03ProdutoService [ti2cc master]
  BD01JavaPostgreSQL
  demo [plmg-cc-ti2-2024-2-g21-patinhasfelizes master]
  Exercicio01 [repository master]
  maven.demo
  maven.exercicio3 [Exercicio01 master 11]
  maven.exercicio4 [Exercicio01 master 11]
  maven.exercicio5
  maven.exercicioodois [Exercicio01 master 11]
  maven.patinhasfelizes [plmg-cc-ti2-2024-2-g21-patinhasfelizes master]
  src/main/java
    app
      Aplicacao.java
    dao
      DAO.java
      DaoAnimal.java
      Animais.java
    model
      Animais.java
      Comentarios.java
      Formularios.java
      Ongs.java
      Usuarios.java
    service
      ServiceAnimais.java
      ServiceComentarios.java
      ServiceFormularios.java
      ServiceOngs.java
      ServiceUsuarios.java
  database.config

  package dao;
  import java.sql.Connection;
  public class DAO {
    //Atributo de conexão com servidor sql
    protected Connection conexao;
    //Construtor padrão
    public DAO() {
      conexao = null;
    }
    //Metodo para realizar a conexão com o servidor sql
    public boolean conectar() {
      boolean status = false;
      try {
        //Ler o arquivo database.config com os endereços a senha e o usuário do servidor sql
        Properties config = new Properties();
        FileInputStream fis = new FileInputStream("src/main/java/dao/database.config");
        config.load(fis);
        //Atribuir os endereços e as senhas as suas respectivas variáveis
        String drivername = "org.postgresql.Driver";
        String servername = config.getProperty("db.servername");
        String mydatabase = config.getProperty("db.mydatabase");
        int porta = 5432;
        String url = "jdbc:postgresql://" + servername + ":" + porta + "/" + mydatabase;
        String username = config.getProperty("db.username");
        String password = config.getProperty("db.password");
        //Realizar as conexões com o servidor sql
        Class.forName(drivername);
        conexao = DriverManager.getConnection(url, username, password);
        status = (conexao == null);
        System.out.println("Conexão efetuada com o postgres!");
      } catch (ClassNotFoundException e){
        System.err.println("Conexão NÃO efetuada com o postgres -- Driver não encontrado -- " + e.getMessage());
      } catch (SQLException e) {
        System.err.println("Conexão NÃO efetuada com o postgres -- " + e.getMessage());
      } catch (IOException e) {
        System.err.println("Não foi possível ler o .config -- " + e.getMessage());
      }
      return status;
    }
    //Metodo para encerrar a conexão com o servidor sql
    public boolean close() {
      boolean status = false;
      try {
        //Encerrar a conexão com o servidor sql
      } catch (SQLException e) {
        System.out.println("Conexão encerrada com sucesso!");
      }
      return status;
    }
  }

```

Writable Smart Insert 1:1:0

24 de out 22:04

Results: untitled:untitled-2 - plmg-cc-ti2-2024-2-g21-patinhasfelizes - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER PLMG-CC-TI2-2024-2-G21-PATINH...

RESULTS: SELECT * FROM animais; Untitled-2

	id	nome	especie	porte	donoid	imagem	vacinas	etiquetas	datasaida	data
1	Amora	CA	G	null		https://www.petz.com.br/blog/wp-content/uploads/2024/04/como-saber-se-o-cachorro-e-de-raca-interna2.jpg	Gripe canina; Glárdia; Antirrábica; Leishmaniose	Brincalhão; Gosta de caminhar;	null	21
27	Charlie	GA	M	null	https://s3.amazonaws.com/petz-cdm-stg/4b7d809d-8c80-4ff7-b7e3-03d1ad2bdb41.jpeg	Trivalente; Felina; Antirrábica	Extrovertido; Brincalhão; Gosta de caminhar	null	21	
28	Max	CA	G	null	https://th.bing.com/th/id/OIP.zliewrMx1krmznPILtpWGwHaHa?rs=1&pid=ImgDetMain	Gripe canina; Parvovírus; Antirrábica; Leptospirose	Brincalhão; Gosta de caminhar;	null	21	
29	Rufus	CA	G	null	https://static.significados.com.br/foto/rottweiler-alemao-cke.jpg	Anti rabica, Raiva	Brincalhão; Caseiro; Introvertido	null	21	

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

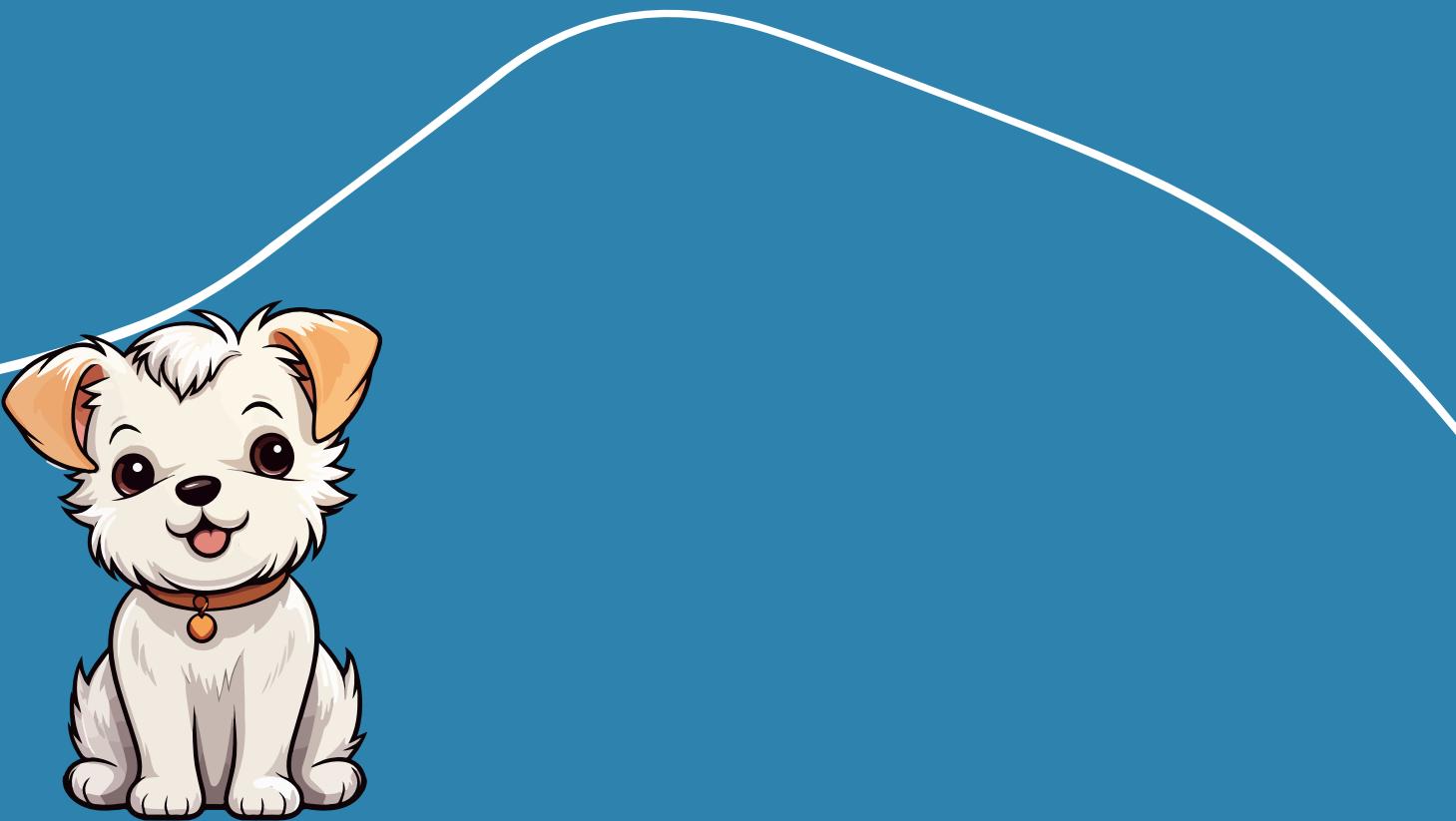
Query completed in 2.2 sec.



Sistema inteligente

Para nosso site, utilizaremos um sistema inteligente com base em reconhecimento e processamento de imagens, o Azure: Detect common objects in images.

Este será responsável pela identificação da espécies de animais enviados por foto e funcionara por aprendizado offline.





Ferramental de IA

AZURE: Detect common objects in images

Entradas

Imagens

Proposição de valor

Precisão da informação

Equipe

Time
Patinhas
Felizes

Clientes

Usuários que encontrarem animais abandonados

Saídas

Percentual de precisão em texto

Agilidade do processamento

Stakeholders Chaves

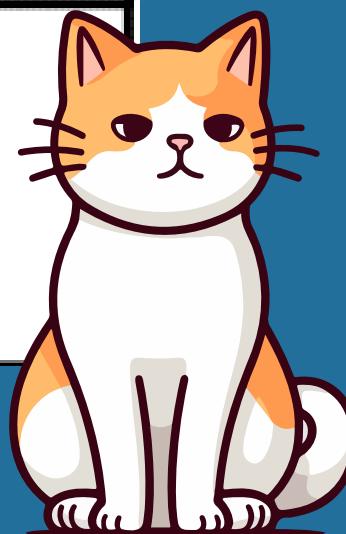
ONG e Adotantes

Custos

Preço do servidor

Receitas

Doações e patrocínios



Ferramental de Inteligência Artificial

AI Toolkit Conceptual Framework

Agentes Inteligentes

- Processamento de Imagem

Machine Learning

- Classificação
- Associação

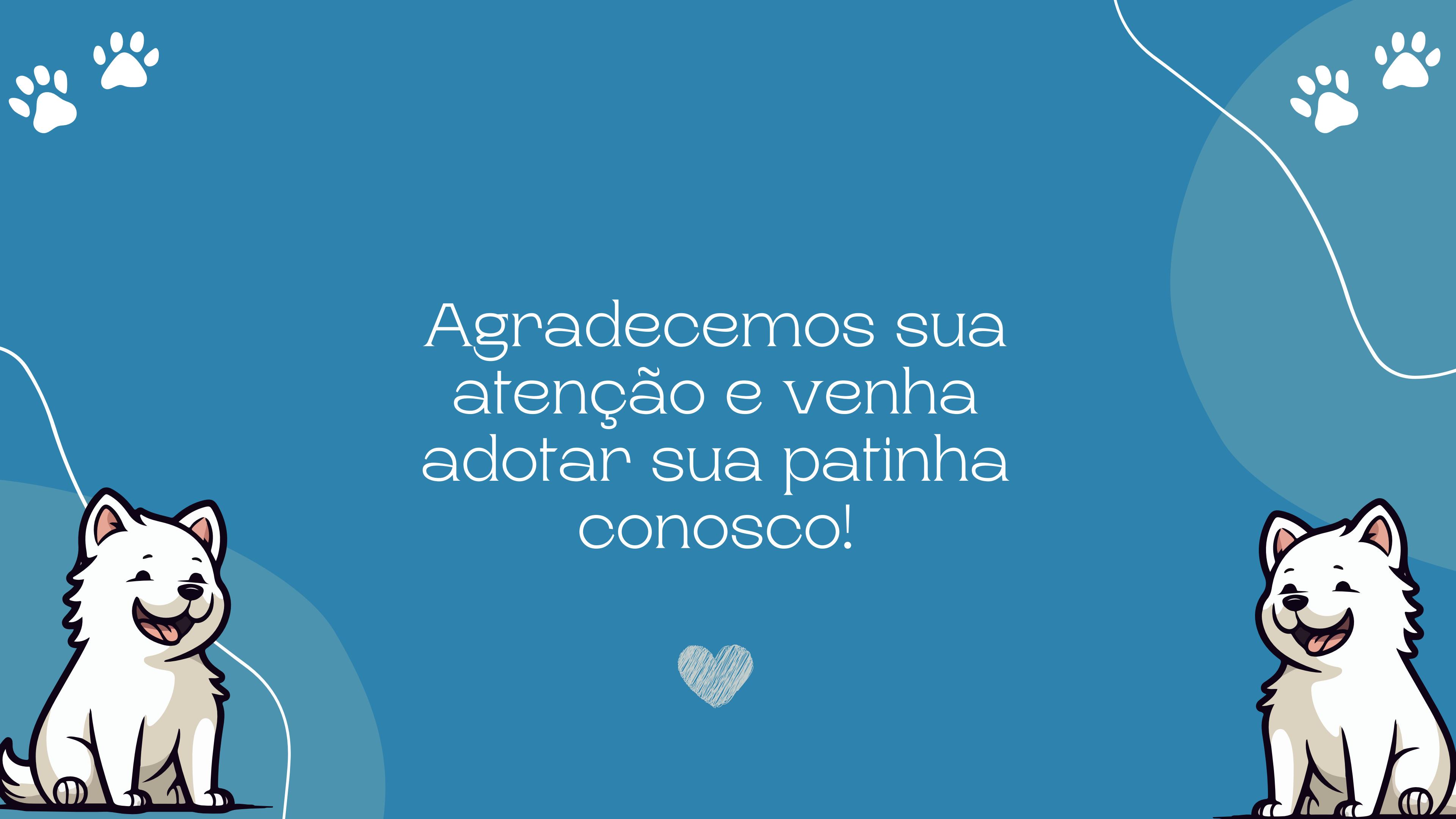
Fala & Texto

- Analise de espécies e raças
- Conversão

Visão Computacional

- Identificação de elementos visuais
- Reconhecimento de Padrões





Agradecemos sua
atenção e venha
adotar sua patinha
conosco!

