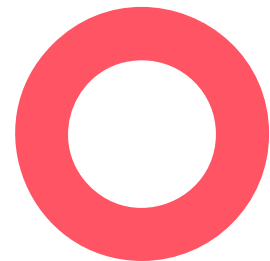


DESENVOLVIMENTO DE SISTEMAS

UC13

Prof. Viviane de Lima

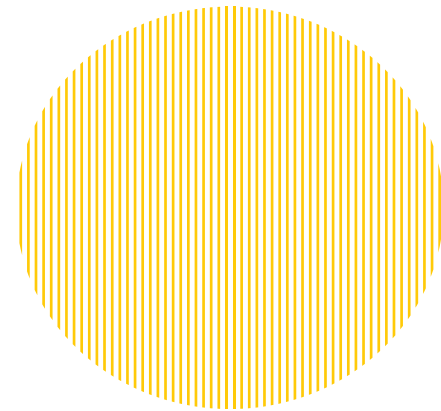
viviane.lfrancelino@sp.senac.br



AULA 08



ATIVANDO O MENU



**PRÓXIMO PASSO É REMOVER ESSA ATIVAÇÃO ‘TODOS’ DO MENU, DEIXAR
ATIVADO APENAS A PÁGINA ACESSADA**



PetShop Recanto dos Pets

Pesquise por raça

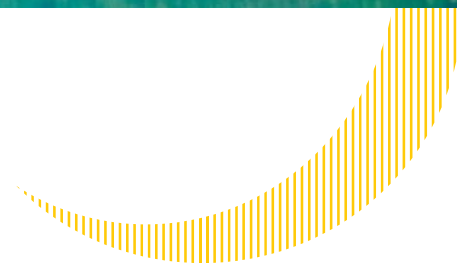
Todos

Cachorros

Gatos

Peixes

Todos os animais



DEPENDENDO DE QUAL MENU EU CLIQUEIM FICARÁ COMO 'ACTIVE'

TS pageController.ts

header.mustache ✕

pages.mustache

src > views > partials > header.mustache

```
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>PetShop Recanto dos Pets</title>
7      <link rel="stylesheet" href="/css/style.css" />
8      <link rel="icon" type="image/png" href="/images/favicon.png" />
9  </head>
10 <body>
11 <header>
12     <div class="container">
13         <a href="/">PetShop Recanto dos Pets</a>
14         <form method="GET" action="/search">
15             <input type="search" name="q" placeholder="Pesquise por raça" />
16         </form>
17     </div>
18 </header>
19 <nav>
20     <ul>
21         <li class="active"><a href="/">Todos</a></li>
22         <li class=""><a href="/dogs">Cachorros</a></li>
23         <li class=""><a href="/cats">Gatos</a></li>
24         <li class=""><a href="/fishes">Peixes</a></li>
25     </ul>
26 </nav>
```

VAMOS CRIAR BOOLEANS PARA CADA UM DOS MENUS

```
TS pageController.ts M X
src > controllers > TS pageController.ts > [🔍] dogs > [🔗] banner
1  import {Request, Response} from 'express'
2
3  export const home = (req:Request, res: Response) =>{
4      res.render("pages/pages",{
5          menu:{
6              all:true,
7              dog:false,
8              cat:false,
9              fish:false
10         },
11         banner:{
12             title:'Todos os animais',
13             background:'allanimals.jpg'
14         }
15     })
16 }
17 export const dogs = (req:Request, res: Response) =>{
18     res.render("pages/pages",{
19         menu:{
20             all:false,
21             dog:true,
22             cat:false,
23             fish:false
24         },
25         banner:{
26             title:'Todos os cachorros',
27             background:'banner_dog.jpg'
28         }
29     })
30 }
```

VAMOS CRIAR BOOLEANS PARA CADA UM DOS MENUS

```
TS pageController.ts M x
src > controllers > TS pageController.ts > [🔍] fishes
29     })
30   }
31   export const cats = (req: Request, res: Response) => {
32     res.render("pages/pages", {
33       menu: {
34         all: false,
35         dog: false,
36         cat: true,
37         fish: false
38       },
39       banner: {
40         title: 'Todos os gatos',
41         background: 'banner_cat.jpg'
42       }
43     })
44   }
45   export const fishes = (req: Request, res: Response) => {
46     res.render("pages/pages", {
47       menu: {
48         all: false,
49         dog: false,
50         cat: false,
51         fish: true
52       },
53       banner: {
54         title: 'Todos os peixes',
55         background: 'banner_fish.jpg'
56       }
57     })
58   }
```


AGORA TEMOS 4 BOOLEANS E CONSEGUIMOS VERIFICAR ESSES BOOLEANS NO NOSSO

TS pageController.ts M

header.mustache M X

src > views > partials > header.mustache

```
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>PetShop Recanto dos Pets</title>
7      <link rel="stylesheet" href="/css/style.css" />
8      <link rel="icon" type="image/png" href="/images/favicon.png" />
9  </head>
10 <body>
11 <header>
12     <div class="container">
13         <a href="/">PetShop Recanto dos Pets</a>
14         <form method="GET" action="/search">
15             <input type="search" name="q" placeholder="Pesquise por raça" />
16         </form>
17     </div>
18 </header>
19 <nav>
20     <ul>
21         <li class="{{#menu.all}}active{/menu.all}}"><a href="/">Todos</a></li>
22         <li class="{{#menu.dog}}active{/menu.dog}}"><a href="/dogs">Cachorros</a></li>
23         <li class="{{#menu.cat}}active{/menu.cat}}"><a href="/cats">Gatos</a></li>
24         <li class="{{#menu.fish}}active{/menu.fish}}"><a href="/fishes">Peixes</a></li>
25     </ul>
26 </nav>
```

se menu.all estiver como true, ele será ativado, a mesma lógica para menu.dog, menu.cat e menu.fish

Todos

Cachorros

Gatos

Peixes



Todos os gatos disponíveis para adoção



Pastor-alemão

Cor: Amarelo e Preto

Feito com ❤️ pela TI

E SE EU TIVER MAIS PÁGINAS? VOU TER QUE FICAR CRIANDO UM NOVO MENU SEMPRE?

TS pageController.ts M X

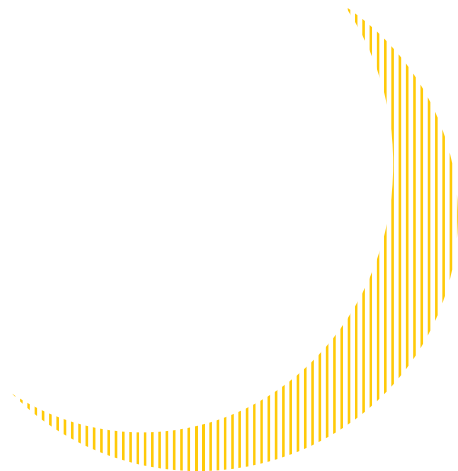
src > controllers > TS pageController.ts > [🔍] fishes

```
29     })
30   }
31   export const cats = (req: Request, res: Response) => {
32     res.render("pages/pages", {
33       menu: {
34         all: false,
35         dog: false,
36         cat: true,
37         fish: false
38       },
39       banner: {
40         title: 'Todos os gatos',
41         background: 'banner_cat.jpg'
42       }
43     })
44   }
45   export const fishes = (req: Request, res: Response) => {
46     res.render("pages/pages", {
47       menu: {
48         all: false,
49         dog: false,
50         cat: false,
51         fish: true
52       },
53       banner: {
54         title: 'Todos os peixes',
55         background: 'banner_fish.jpg'
56       }
57     })
58   }
```

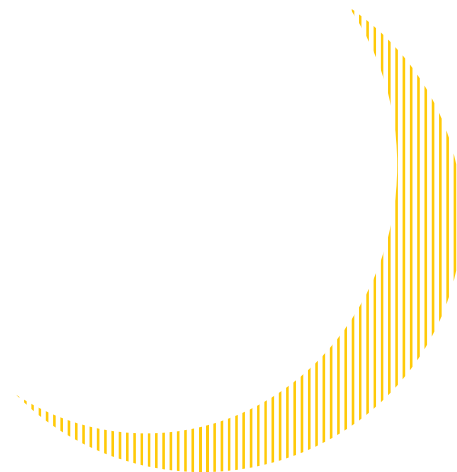
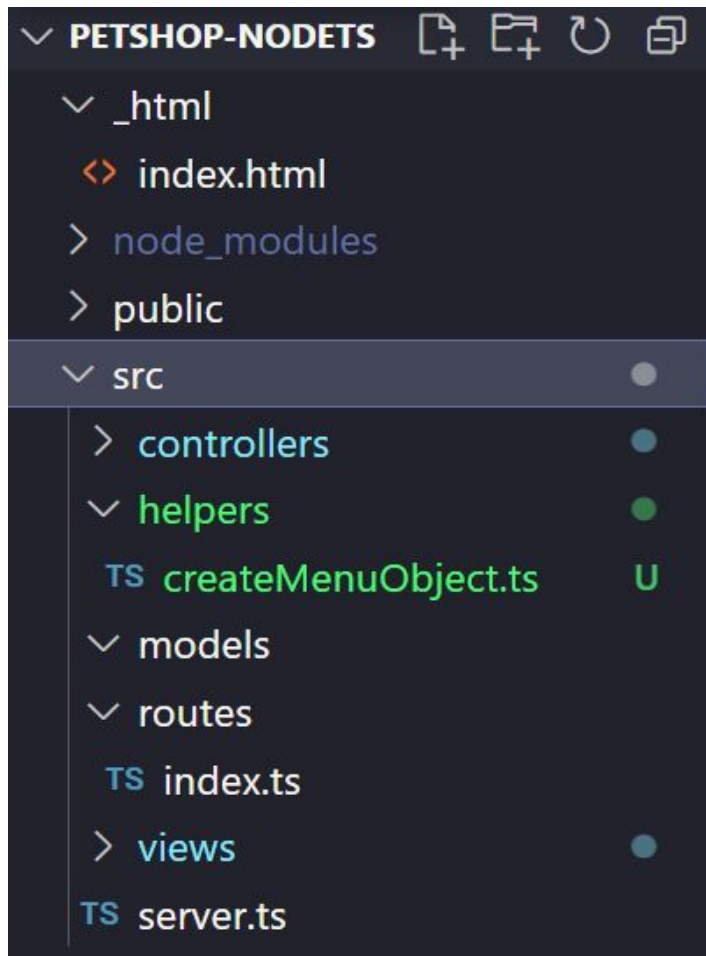
SIM, SERÁ NECESSÁRIO CRIAR ESSE MENU TODA VEZ QUE UMA PÁGINA FOR CRIADA, ENTÃO VAMOS PRECISAR MELHORAR ESSE CÓDIGO



**QUANDO QUEREMOS CRIAR UMA FUNÇÃO QUE VAI AUXILIAR EM VÁRIAS
PAGINAS DE UM CONTROLLER, CHAMAMOS ESSA FUNÇÃO DE **HELPER!****



**CRIE UMA PASTA NOVA DENTRO DE SRC CHAMADA HELPERS E DENTRO DA PASTA
HELPER CRIE UM ARQUIVO CHAMADO CREATEMENUOBJECT.TS**



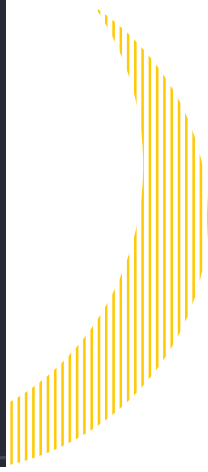
AQUI VAMOS RECEBER UMA STRING ACTIVEMENU E ELE VAI RECEBER O MENU QUE EU QUERO QUE ESTEJA ATIVO



TS createMenuObject.ts U X

src > helpers > TS createMenuObject.ts > ...

```
1  /*criando o type para o activeMenu
2  o menu pode estar em branco OU ele pode ter all OU ele pode ter dog
3  OU ele pode ter cat, OU ele pode ter fish */
4  type MenuOptions = '' | 'all' | 'dog' | 'cat' | 'fish'
5
6  export const createMenuObject = (activeMenu: MenuOptions) =>{
7
8      //agora vamos criar o objeto com todos do tipo FALSE
9      let returnObject = {
10         all:false,
11         dog:false,
12         cat:false,
13         fish:false
14     }
15     /*agora vamos transformar o item que queremos para true
16     SE activeMenu for diferente de vazio
17     então returnObject será true */
18     if(activeMenu !== ''){
19         returnObject[activeMenu] = true
20     }
21     //depois vamos retornar returnObject
22     return returnObject
23 }
```



VÁ EM PAGECONTROLLER E IMPORTE O HELPER QUE CRIAMOS

TS createMenuObject.ts U

TS pageController.ts M X

src > controllers > TS pageController.ts > [e] dogs

```
1  import {Request, Response} from 'express'
2  import { createMenuObject } from '../helpers/createMenuObject'
3
4  export const home = (req:Request, res: Response) =>{
5      res.render("pages/pages",{
6          menu:{
7              all:true,
8              dog:false,
9              cat:false,
10             fish:false
11         },
12         banner:{
13             title:'Todos os animais',
14             background:'allanimals.jpg'
15         }
16     })
17 }
18 export const dogs = (req:Request, res: Response) =>{
19     res.render("pages/pages",{
20         menu:{
```

VAMOS USAR O NOSSO HELPER DENTRO NO MENU

```
TS createMenuObject.ts U TS pageController.ts 1, M X
```

src > controllers > TS pageController.ts > home > menu

```
1 import {Request, Response} from 'express'
2 import { createMenuObject } from '../helpers/createMenuObject'
3
4 export const home = (req:Request, res:Response) =>{
5     res.render("pages/pages",{
6         menu:createMenuObject('')
7         banner:{
8             title:'Todos os ani
9             background:'allanim
10        }
11    })
12 }
13 export const dogs = (req:Request, res: Response) =>{
14     res.render("pages/pages",{
15         menu:{
16             all:false,
17             dog:true,
18             cat:false,
19             fish:false
20         },
21         banner:{
22             title:'Todos os cachorros',
23             background:'banner_dog.jpg'
```

createMenuObject(activeMenu: MenuOptions): { all: boolean; dog: boolean; cat: boolean; fish: boolean; }

all
cat
dog
fish

Assim que abrimos aspas ele já nos mostra qual menu deveremos selecionar, isso só é possível graças ao Typescript

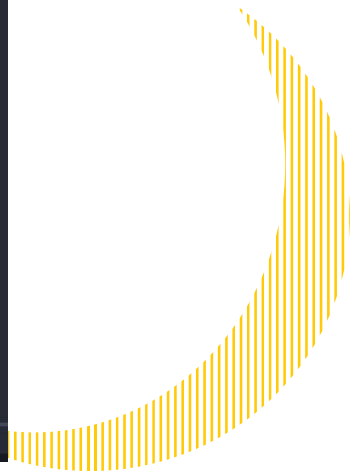
COLOQUE O MENU DE ACORDO COM AS PÁGINAS QUE CRIAMOS

```
export const home = (req: Request, res: Response) => {
  res.render("pages/pages", {
    menu: createMenuObject('all'),
    banner: {
      title: 'Todos os animais',
      background: 'allanimals.jpg'
    }
  })
}

export const dogs = (req: Request, res: Response) => {
  res.render("pages/pages", {
    menu: createMenuObject('dog'),
    banner: {
      title: 'Todos os cachorros',
      background: 'banner_dog.jpg'
    }
  })
}

export const cats = (req: Request, res: Response) => {
  res.render("pages/pages", {
    menu: createMenuObject('cat'),
    banner: {
      title: 'Todos os gatos',
      background: 'banner_cat.jpg'
    }
  })
}

export const fishes = (req: Request, res: Response) => {
  res.render("pages/pages", {
    menu: createMenuObject('fish'),
```



[Todos](#)[Cachorros](#)[Gatos](#)[Peixes](#)

Todos os cachorros

Todos os cachorros disponíveis para adoção

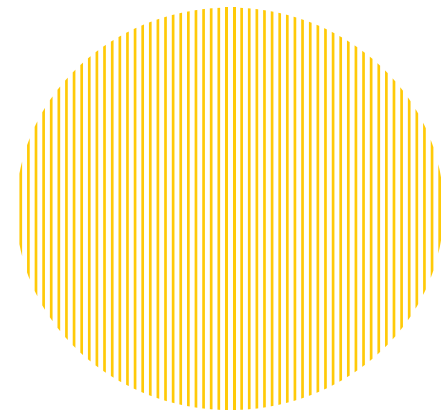


Pastor-alemão
Cor: Amarelo e Preto

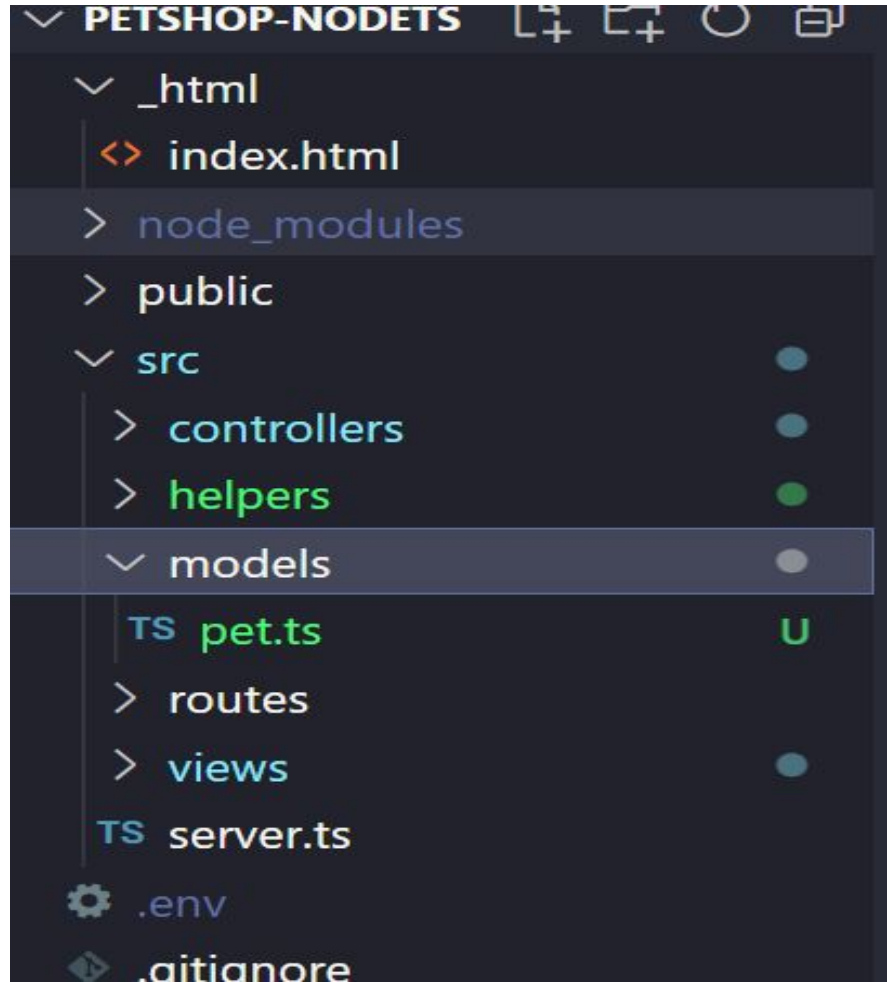
Feito com ❤️ pela TI



CRIANDO O MODEL PET 1



AGORA VAMOS CRIAR O MODEL DOS PETS PARA EXIBIR TODOS OS PETS NA NOSSA PÁGINA



**CRIE UM NOVO ARQUIVO
DENTRO DA PASTA MODEL
CHAMADA PET.TS**

O MODEL VAI SER UM OBJETO E ESSE OBJETO TERÁ FUNÇÕES PARA FAZERMOS COM OS PETS

TS createMenuObject.ts U

TS pageController.ts M

TS pet.ts U X

src > models > TS pet.ts

```
1  /* TEREMOS AO MENOS 3 FUNÇÕES DIFERENTES
2  1. PEGAR TODOS OS PETS
3  2. FILTRAR OS PETS POR TIPOS
4  3. FILTRAR OS PETS PELO NOME */
```

1 FUNÇÃO PEGAR TODOS OS PETS

VAMOS CRIAR UM MODELO DE BANCO DE DADOS

TS pet.ts 1, U X

src > models > TS pet.ts > ...

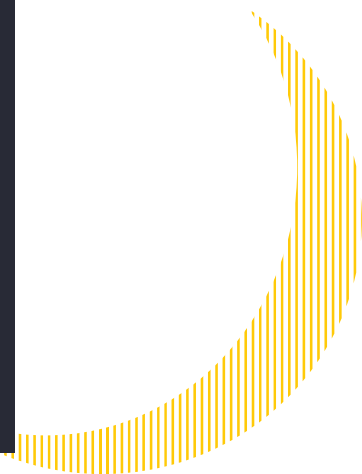
```
1  /*como ainda não sabemos banco de dados, vamos criar um array abaixo
2  e vamos fingir que esse array é o nosso banco de dados */
3
4  const data =
5  /*o arquivo de todos os pets é bem grande, a professora irá disponibilizar
6  esse arquivo para você */
7
```


O CÓDIGO VAI FICAR ASSIM

TS pet.ts U X

src > models > TS petts > [e] data

```
1 //copie e cole o arquivo disponibilizado pela prof
2 const data =
3 [
4   {
5     type: 'dog',
6     image: 'pastor-alemao.jpg',
7     name: 'Pastor-alemão',
8     color: 'Amarelo e Preto',
9     sex: 'Masculino'
10  },
11  {
12    type: 'dog',
13    image: 'labrador.jpg',
14    name: 'Labrador-retriever',
15    color: 'Branco',
16    sex: 'Masculino'
17  },
18  {
19    type: 'dog',
20    image: 'zwerdspitz.jpg',
21    name: 'Zwerdspitz',
22    color: 'Amarelo',
23    sex: 'Feminino'
24  },
```



VAMOS TIPAR OS DADOS DOS ANIMAIS PARA GARANTIR QUE TODOS TENHAM AS MESMAS INFORMAÇÕES



TS pet.ts U X

src > models > TS pet.ts > [X] data

```
1  type Pet = {
2    /*o type vamos deixar apenas para os tipos dos animais que
3    temos disponiveis(cachorro OU gato OU peixe),
4    imagem podemos deixar como string, nome e cor também como string.
5    Sexo só temos dois tipos, então será Masculino ou Feminino*/
6    type: 'dog' | 'cat' | 'fish',
7    image: string,
8    name: string,
9    color: string,
10   sex: 'Masculino' | 'Feminino'
11 }
12 //agora data é um ARRAY DE PET
13 const data: Pet[] =
14 [
15   {
16     type: 'dog',
17     image: 'pastor-alemao.jpg',
18     name: 'Pastor-alemão',
19     color: 'Amarelo e Preto',
20     sex: 'Masculino'
21   }
```

1 FUNÇÃO: EXIBIR TODOS OS PETS

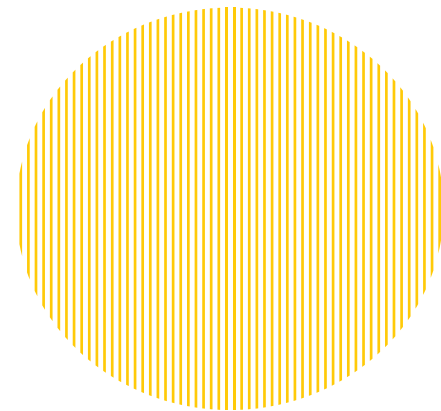
TS pet.ts U X

src > models > TS pet.ts > ...

```
1  type Pet = {
2    type: 'dog' | 'cat' | 'fish',
3    image: string,
4    name: string,
5    color: string,
6    sex: 'Masculino' | 'Feminino'
7  }
8  > const data: Pet[] = ...
129 ]
130 export const Pet = {
131   /*vamos retornar TODOS os pets para
132   exibir na página principal, então essa função
133   vai retornar data que é onde estão todos os nosso pets */
134   getAll: () =>{
135     return data
136   }
137 }
138
```



CRIANDO O MODEL PET 2



VAMOS CRIAR UM TYPE QUE VAI RECEBER O TIPO DE PET QUE EU QUERO

TS pet.ts M X

src > models > TS pet.ts > [data] sex

```
1 //CRIAR UM TYPE PARA BUSCARMOS O TIPO DE PET
2 type PetType = 'dog' | 'cat' | 'fish'
3
4 type Pet = {
5     /*como criamos um type específico para pet,
6     podemos dizer que type recebe, PetType */
7     type: PetType,
8     image: string,
9     name: string,
10    color: string,
11    sex: 'Masculino' | 'Feminino'
12 }
13 const data: Pet[] =
14 [
15     {
16         type: 'dog',
17         image: 'pastor-alemao.jpg',
18         name: 'Pastor-alemao'
```

2 FUNÇÃO: FILTRAR OS PETS POR TIPO

FILTRAGEM DOS PETS



```
TS pet.ts  M X
src > models > TS pet.ts > ...
 4 |         type: PetType,
 5 |         image: string,
 6 |         name: string,
 7 |         color: string,
 8 |         sex: 'Masculino' | 'Feminino'
 9 |     }
10 > const data: Pet[] = ...
131 |
132 |
133 | export const Pet = {
134 |     getAll: () =>{
135 |         return data
136 |     },
137 |     getFromType: (type: PetType): Pet[] =>{
138 |         /*como ainda não temos um banco de dados
139 |         vamos dar um return data.filter*/
140 |         return data.filter(item =>{
141 |
142 |
143 |         })
144 |     }
145 | }
```


FAZENDO A FILTRAGEM DOS PETS PELO TIPO

```
TS pet.ts  M X
src > models > TS pet.ts > ...
4 |     type: PetType,
5 |     image: string,
6 |     name: string,
7 |     color: string,
8 |     sex: 'Masculino' | 'Feminino'
9 |   }
10 > const data: Pet[] = ...
131 | ]
132 |
133 | export const Pet = {
134 |   getAll: () =>{
135 |     return data
136 |   },
137 |   getFromType: (type: PetType): Pet[] =>{
138 |
139 |     return data.filter(item =>{
140 |       /*se item.type for igual ao type
141 |       que eu recebi no meu parâmetro(que é o petType)
142 |       então retorne true para esse item */
143 |       if(item.type == type){
144 |         return true
145 |       } //caso contrário retorne falso para esse item
146 |       else{
147 |         return false
148 |       }
149 |     })
150 |   }
151 | }
```

3 FUNÇÃO: FILTRAR O PET PELO NOME DELE

```
export const Pet = {  
  getAll: () =>{  
    return data  
  },  
  getFromType: (type: PetType): Pet[] =>{  
  
    return data.filter(item =>{  
      if(item.type == type){  
        return true  
      }else{  
        return false  
      }  
    })  
  },  
  //filtrar o nome do pet  
  getFromName: (name: string): Pet[] =>{  
    return data.filter(item => {  
      /*SE item.name.indexOf retornar MAIS DO QUE menos 1  
      significa que eu achei a palavra que o usuário digitou*/  
      if(item.name.indexOf(name) > -1){  
        //se ele achou a palavra retorna true  
        return true  
      } else {  
        //se não achou retorne false  
        return false  
      }  
    })  
  }  
}
```

Exemplo: se eu digitei a palavra BULL, ele vai encontrar todos os pets que tenham BULL no nome, então, ele encontrará o pet BULLdog, por exemplo.

Por que menos 1? Quando o indexOf não acha nenhuma palavra que você buscou ele retorna menos 1

EXPLICAÇÃO DO -1 (JAVASCRIPT BÁSICO)



Lembra da posição dos arrays? Usamos essa lógica para usar o -1 quando queremos buscar algo. Vamos supor que você digitou na barra de pesquisa a palavra: BULL

BULL

Qual a posição da letra B?

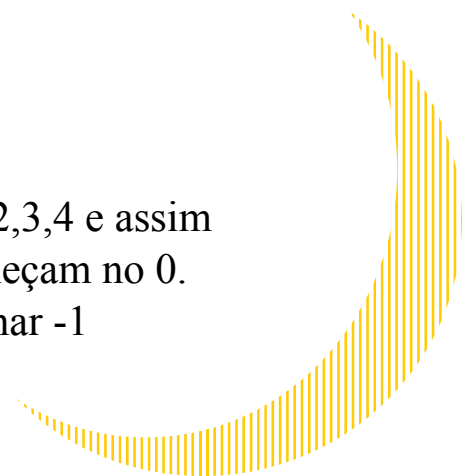
Se você disse 0, acertou

Qual a posição da letra U?

Se você disse 1, acertou

Então tudo que for **MAIOR DO QUE -1** consequentemente será 0,1,2,3,4 e assim sucessivamente, não existe posição de array com -1, eles sempre começam no 0.

Se você digitar algum nome de pet que não existe, ele vai retornar -1



USANDO TOLOWERCASE()

TS pet.ts M X

src > models > TS pet.ts > ...

```
138
139   getFromType: (type: PetType): Pet[] =>{
140
141       return data.filter(item =>{
142           if(item.type == type){
143               return true
144           }else{
145               return false
146           }
147       })
148   },
149   /*precisamos arrumar a busca que o usuário vai fazer do pet em letra minúscula
150   se o usuário digitar poodle, ele não vai encontrar, ele só vai encontrar
151   se o usuário digitar Poodle, por exemplo. */
152   getFromName: (name: string): Pet[] =>{
153       return data.filter(item => {
154           if(item.name.toLowerCase().indexOf(name.toLowerCase()) > -1){
155               return true
156           } else {
157               return false
158           }
159       })
160   }
161 }
```

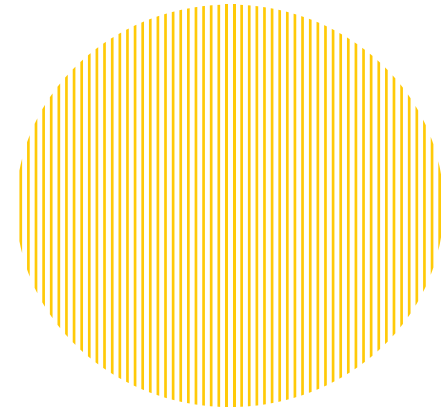
MODEL COMPLETO, 3 FUNÇÕES CRIADAS

```
export const Pet = {
  //1 buscar todos os pets
  getAll: () =>{
    return data
  },
  //2 buscar os pets por tipo
  getFromType: (type: PetType): Pet[] =>{

    return data.filter(item =>{
      if(item.type == type){
        return true
      }else{
        return false
      }
    })
  },
  //3 buscar o tipo de pet pelo nome
  getFromName: (name: string): Pet[] =>{
    return data.filter(item => {
      if(item.name.toLowerCase().indexOf(name.toLowerCase()) > -1){
        return true
      } else {
        return false
      }
    })
  }
}
```



LISTANDO OS PETS NO CONTROLLER



AGORA VAMOS COLOCAR NOSSA MODEL DENTRO DO PAGECONTROLLER

TS pageController.ts M X

src > controllers > TS pageController.ts > [🔗] dogs > 🔑 banner

```
1  import {Request, Response} from 'express'
2  import { createMenuObject } from '../helpers/createMenuObject'
3  //importe a model pet
4  import { Pet } from '../models/pet'
5
6  export const home = (req:Request, res: Response) =>{
7      /*na página inicial vamos exibir todos os pets,
8      então use a função getAll*/
9      let list = Pet.getAll()
10     res.render("pages/pages",{
11         menu:createMenuObject('all'),
12         banner:{
13             title:'Todos os animais',
14             background:'allanimals.jpg'
15         },
16         //declare a list dentro de pages
17         list
18     })
19 }
```

AGORA TEMOS ACESSO A LET LIST DENTRO DA VIEW, VÁ EM PAGES.MUSTACHE

pages.mustache M X

src > views > pages > pages.mustache

```
1  {{>partials/header}}
2
3  {{#banner}}
4  <section class="banner" style="background-image: url('images/{{background}}')">{{title}}</section>
5  <h2>{{title}} disponíveis para adoção</h2>
6  {{/banner}}
7  <div class="container list">
8      {{#list}}
9      <div class="item">
10         
11         <div class="item--name">{{ name }}</div>
12         <div class="item--color">Cor:{{ color }}</div>
13         <div class="item--genre">Gênero: {{ sex }}</div>
14     </div>
15     {{/list}}
16 </div>
17
18 {{>partials/footer}}
```

ATUALIZE O SERVIDOR E VEJA SE ESTÁ FUNCIONANDO



Todos os animais disponíveis para adoção



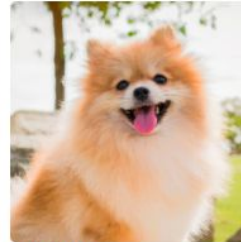
Pastor-alemão

Cor: Amarelo e Preto
Gênero: Masculino



Labrador-retriever

Cor: Branco
Gênero: Masculino



Zwergspitz

Cor: Amarelo
Gênero: Feminino



Husky Siberiano

Cor: Branco e Preto
Gênero: Masculino

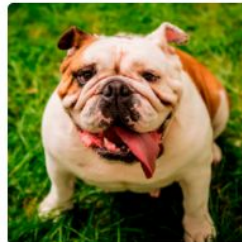


Golden Retriever

Cor: Amarelo
Gênero: Masculino



Poodle



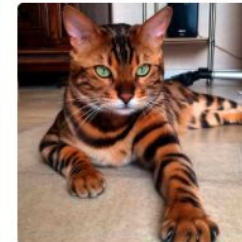
Bulldog



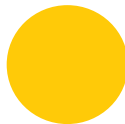
Persa



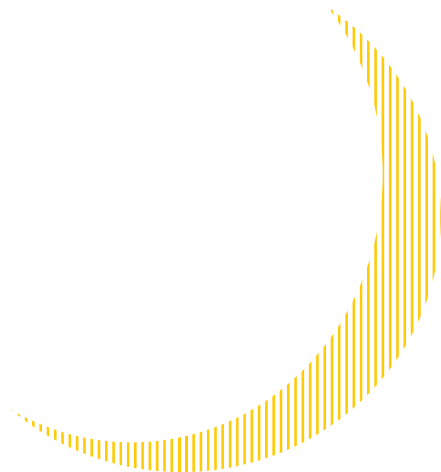
Maine Coon



Bengal



AGORA É SUA VEZ
TENTE A FAZER AS ROTAS: DOG, CAT E FISH



RESOLUÇÃO DOG E CAT

TS pageController.ts M X

src > controllers > TS pageController.ts > [🔗] cats > 🔑 list

```
16 export const dogs = (req: Request, res: Response) =>{
17   let list = Pet.getFromType('dog')
18
19   res.render("pages/pages",{
20     menu:createMenuObject('dog'),
21     banner:{
22       title:'Todos os cachorros',
23       background:'banner_dog.jpg'
24     },
25     list
26   })
27 }
28 export const cats = (req: Request, res: Response) =>{
29   let list = Pet.getFromType('cat')
30   res.render("pages/pages",{
31     menu:createMenuObject('cat'),
32     banner:{
33       title:'Todos os gatos',
34       background:'banner_cat.jpg'
35     },
36     list
37   })
38 }
```


RESOLUÇÃO FISH



TS pageController.ts M X

src > controllers > TS pageController.ts > [0] fishes

```
26     })
27   }
28   export const cats = (req: Request, res: Response) => {
29     let list = Pet.getFromType('cat')
30     res.render("pages/pages", {
31       menu: createMenuObject('cat'),
32       banner: {
33         title: 'Todos os gatos',
34         background: 'banner_cat.jpg'
35       },
36       list
37     })
38   }
39   export const fishes = (req: Request, res: Response) => {
40     let list = Pet.getFromType('fish')
41     res.render("pages/pages", {
42       menu: createMenuObject('fish'),
43       banner: {
44         title: 'Todos os peixes',
45         background: 'banner_fish.jpg'
46       },
47       list
48     })
49   }
```

VERIFIQUE SE AS ROTAS ESTÃO FUNCIONANDO

PetShop Recanto dos Pets

Todos

Cachorros

Gatos

Peixes



Todos os peixes disponíveis para adoção



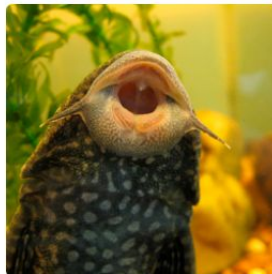
Tetra Neon

Cor: Vermelho e Azul
Gênero: Masculino



Mato Grosso

Cor: Laranja
Gênero: Masculino



Limpa Vidro

Cor: Verde e Branco
Gênero: Masculino



Tanictis

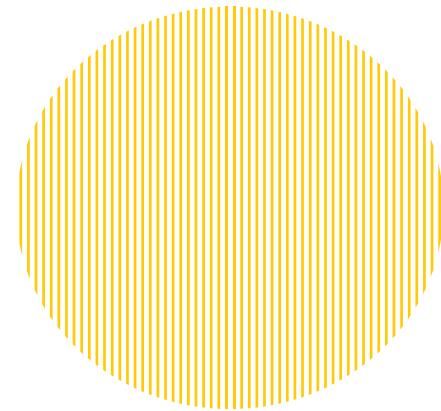
Cor: Vermelho
Gênero: Masculino



Acará Bandeira

Cor: Preto
Gênero: Masculino

FAZENDO A BUSCA E O 404



VAMOS FAZER A BUSCA, VÁ EM SEARCHCONTROLLER

TS pageController.ts

TS searchController.ts M X

pages.mustache

src > controllers > TS searchController.ts > ...

```
1 import {Request, Response} from 'express'
2
3 //importar o model
4 import { Pet } from '../models/pet'
5
6 //importar o menu
7 import { createMenuObject } from '../helpers/createMenuObject'
8
9 export const search = (req:Request, res: Response) =>{
10     //renderizar a página
11     res.render('pages/pages',{
12         //aqui dentro vamos mandar os nosso itens
13         menu:createMenuObject('')
14     })
15 }
16
```

DIGITE QUALQUER COISA NA BUSCA E VEJA SE APARECE DESSA FORMA:



PetShop Recanto dos Pets

Pesquise por raça

Todos

Cachorros

Gatos

Peixes

Feito com ❤️ pela TI

ELE NÃO EXIBIU NADA POIS NÃO MANDAMOS A LISTA DE ITENS AINDA

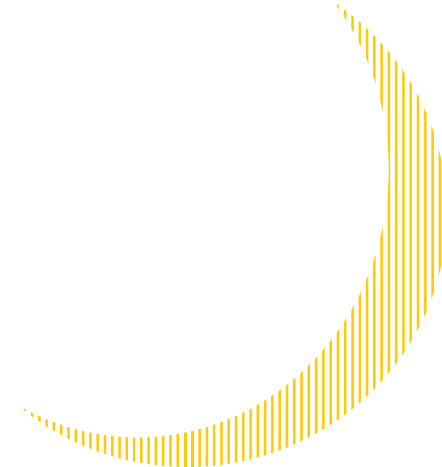
PRECISAMOS PEGAR A INFORMAÇÃO QUE O USUÁRIO DIGITOU, NESSE CASO O q



localhost/search?q=spitz



PetShop Recanto dos Pets



PEGANDO Q

TS pageController.ts

TS searchController.ts M X

pages.mustache

src > controllers > TS searchController.ts > [🔍] search

```
1  import {Request, Response} from 'express'
2  import { Pet } from '../models/pet'
3  import { createMenuObject } from '../helpers/createMenuObject'
4
5
6  export const search = (req:Request, res: Response) =>{
7      /*vamos usar o request para pegar o q
8      let query vai receber req.query.q que vai ser uma string */
9      let query: string = req.query.q as string
10
11      /*em seguida pegamos o que o usuário digitou e exibimos a lista
12      ele vai fazer a filtragem e gerar uma lista com os nomes parecidos dos pets */
13      let list = Pet.getFromName(query)
14
15      res.render('pages/pages',{
16          menu:createMenuObject(''),
17          //em seguida mando a minha lista para render
18          list
19      })
20 }
```

TESTE A BUSCA NO NAVEGADOR

PetShop Recanto dos Pets

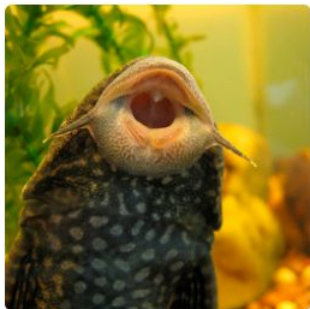
Pesquise por raça

Todos

Cachorros

Gatos

Peixes



Limpa Vidro

Cor: Verde e Branco

Gênero: Masculino

Feito com ❤️ pela TI

AGORA VAMOS EXIBIR UMA MENSAGEM AO USUÁRIO

CASO NÃO TENHA NENHUM ANIMAL COM AQUELE NOME OU LETRA



PetShop Recanto dos Pets

Pesquise por raça

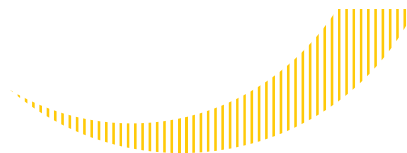
Todos

Cachorros

Gatos

Peixes

Feito com ♥ pela TI



EM PAGE.MUSTACHE DEIXE ASSIM

pages.mustache M X

src > views > pages > pages.mustache

```
4 <section class="banner" style="background-image: url('{{background}}')">{{title}}
5 <h2>{{title}} disponíveis para adoção</h2>
6 {{/banner}}
7 <div class="container list">
8   {{#list}}
9   <div class="item">
10     
11     <div class="item--name">{{ name }}</div>
12     <div class="item--color">Cor:{{ color }}o</div>
13     <div class="item--genre">Gênero: {{ sex }}</div>
14   </div>
15   {{/list}}
16 </div>
17
18 <!-- quando list não existir exibimos outra coisa -->
19 {{^list}}
20   <div class="container no-list">
21     <h2>Não existem animais disponíveis com esse nome</h2>
22   </div>
23 {{/list}}
24
25 {{>partials/footer}}
```

VEJA NO NAVEGADOR



PetShop Recanto dos Pets

Pesquise por raça

Todos

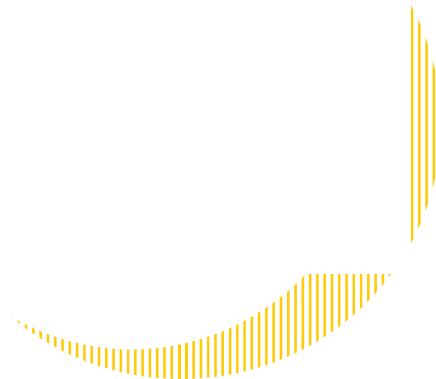
Cachorros

Gatos

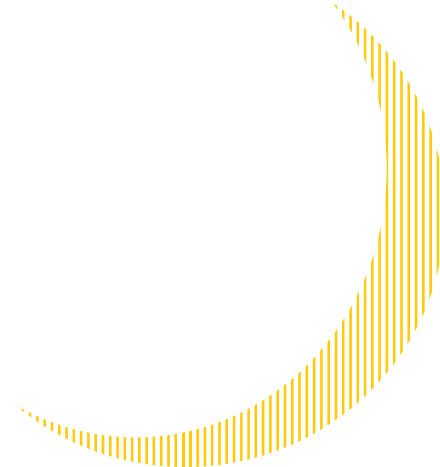
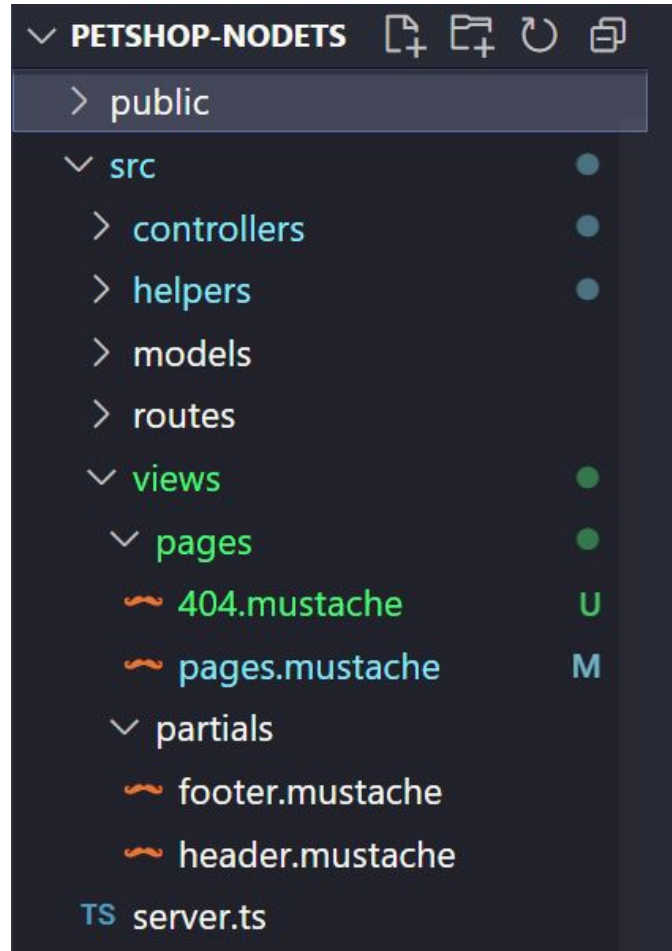
Peixes

**Não existem
animais
disponíveis
com esse
nome**

Feito com ❤️ pela TI



VAMOS CRIAR A PÁGINA 404.MUSTACHE



DEIXE A PÁGINA ASSIM



pages.mustache M

404.mustache U X

src > views > pages > 404.mustache

```
1  {{>partials/header}}
2
3  <div class="container list">
4
5  </div>
6
7  <div class="container no-list">
8    <h2>Página não encontrada</h2>
9  </div>
10
11 {{>partials/footer}}
```

VÁ EM SERVER.TS E COLOQUE A PÁGINA 404



```
TS server.ts M ×  pages.mustache M  404.mustache U
src > TS server.ts > ...
1  import express from 'express'
2  import dotenv from 'dotenv'
3  import mustache from 'mustache-express'
4  import path from 'path'
5  import mainRoutes from './routes/index'
6
7  dotenv.config()
8  const server = express()
9  server.set('view engine', 'mustache')
10 server.set('views', path.join(__dirname, 'views'))
11 server.engine('mustache', mustache())
12
13 server.use(express.static(path.join(__dirname, '../public')))
14
15 server.use(mainRoutes)
16
17 server.use((req, res) => {
18
19     //res.send('Página não encontrada!')
20     res.render('pages/404')
21 })
22
23 server.listen(process.env.PORT)
```

VEJA SE ESTÁ FUNCIONANDO



PetShop Recanto dos Pets

Todos

Cachorros

Gatos

Peixes

Página não encontrada

Feito com ❤️ pela TI

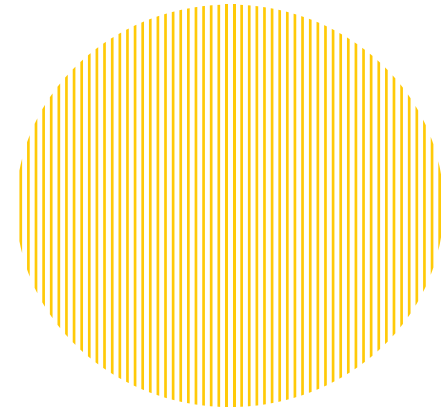


ATUALIZE SEU REPOSITÓRIO NO GITHUB

```
PS C:\Users\Viviane\Downloads\projeto\petshop-nodets> git add .
PS C:\Users\Viviane\Downloads\projeto\petshop-nodets> git commit -m "projeto finalizado"
[main 334ee13] projeto finalizado
21 files changed, 115 insertions(+), 35 deletions(-)
create mode 100644 public/images/arara-azul.jpg
create mode 100644 public/images/arara-caninde.jpg
create mode 100644 public/images/banner_bird.jpg
create mode 100644 public/images/cacatua-rosa.jpg
create mode 100644 public/images/calopsita.jpg
create mode 100644 public/images/pato-mandarim.jpg
create mode 100644 public/images/ringneck.jpg
create mode 100644 public/images/screenshot1.png
create mode 100644 public/images/screenshot2.png
create mode 100644 public/images/screenshot3.png
create mode 100644 public/images/screenshot4.png
create mode 100644 public/images/tucano.jpg
rewrite src/helpers/createMenuObject.ts (64%)
create mode 100644 src/views/pages/404.mustache
PS C:\Users\Viviane\Downloads\projeto\petshop-nodets> git push
```



MIDDLEWARES





SESSIONS E COOKIES

