

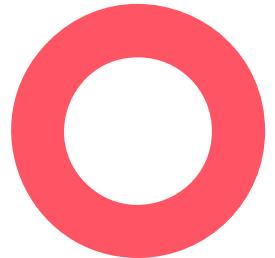
# **DESENVOLVIMENTO DE SISTEMAS**

## **UC13**

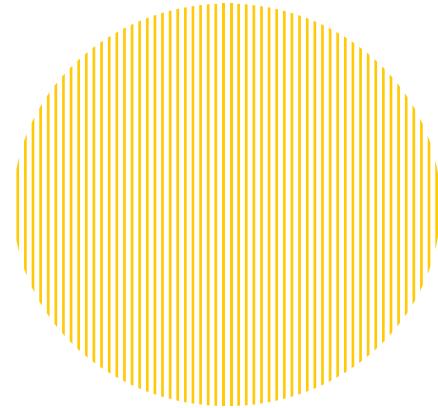
**Prof. Viviane de Lima**

[viviane.lfrancelino@sp.senac.br](mailto:viviane.lfrancelino@sp.senac.br)

**AULA 07**



# PROJETO PETSHOP



# BAIXE OS ARQUIVOS QUE A PROFESSORA DISPONIBILIZOU PARA INICIARMOS O PROJETO



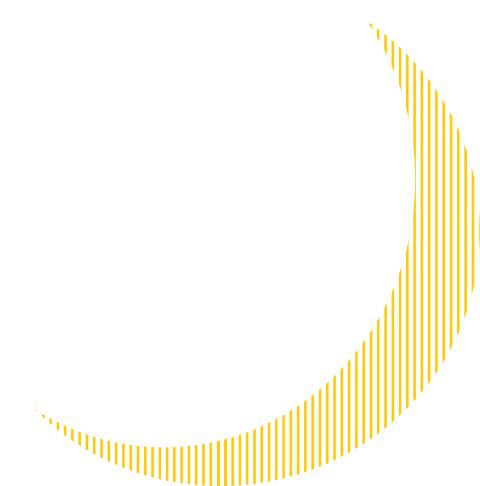
css



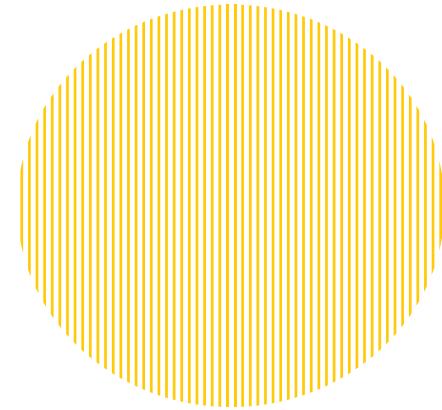
images



index

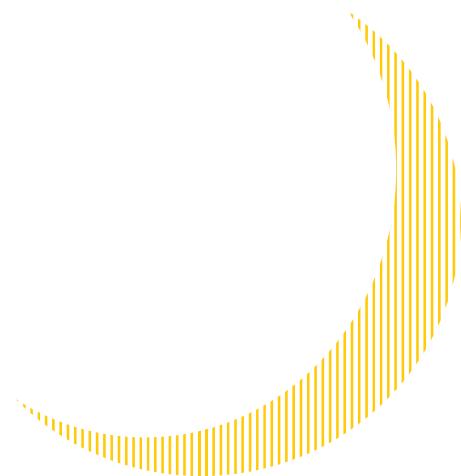


# **INICIANDO O PROJETO**





**CRIE UMA PASTA CHAMADA PROJETO EM DOWNLOADS**



# CRIE UM REPOSITÓRIO NO GITHUB COM README FILE E .GITIGNORE: NODE

## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?  
[Import a repository.](#)

Owner \*



zennom ▾

Repository name \*

/ petshop-nodets ✓

Great repository names are short and memorable. Need inspiration? How about [psychic-octo-spork](#)?

Description (optional)

Public

Anyone on the internet can see this repository. You choose who can commit.

Private

You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

Add a README file

This is where you can write a long description for your project. [Learn more](#).

Add .gitignore

Choose which files not to track from a list of templates. [Learn more](#).

.gitignore template: Node ▾

Choose a license

A license tells others what they can and can't do with your code. [Learn more](#).

This will set main as the default branch. Change the default name in your [settings](#).

VAMOS CLONAR O PROJETO PARA CONSEGUIRMOS USÁ-LO, COPIE A URL DO SEU

The screenshot shows a GitHub repository page for 'petshop-nodets'. At the top, there's a search bar with placeholder text 'Search or jump to...', and navigation links for 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. Below the header, the repository name 'petshop-nodets' is displayed, along with a 'Public' badge and a 'Star' icon. A navigation bar below the repository name includes 'Issues', 'Pull requests', 'Actions', 'Projects', 'Wiki', 'Security', 'Insights', and 'Settings'. On the left, there are buttons for 'main' (with a dropdown arrow), '1 branch', and '0 tags'. On the right, there are buttons for 'Go to file', 'Add file', and 'Code' (with a dropdown arrow). The main content area shows a commit from 'zennom' titled 'Initial commit' made 2 minutes ago. It includes files '.gitignore' and 'README.md', both of which are also listed as 'Initial commit' made 2 minutes ago. Below this, there's a preview of the 'README.md' file content.

https://github.com/zennom/petshop-nodets

Search or jump to... /

Pull requests Issues Marketplace Explore

zennom / petshop-nodets Public

Issues Pull requests Actions Projects Wiki Security Insights Settings

main ▾ 1 branch 0 tags

Go to file Add file ▾ Code ▾

zennom Initial commit c0210b3 2 minutes ago 1 commit

.gitignore Initial commit 2 minutes ago

README.md Initial commit 2 minutes ago

README.md

petshop-nodets

## ACESSE A SUA PASTA PROJETO ATRAVÉS DO CMD

Prompt de Comando

```
Microsoft Windows [versão 10.0.19044.1526]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\Users\Viviane>cd downloads

C:\Users\Viviane\Downloads>cd projeto

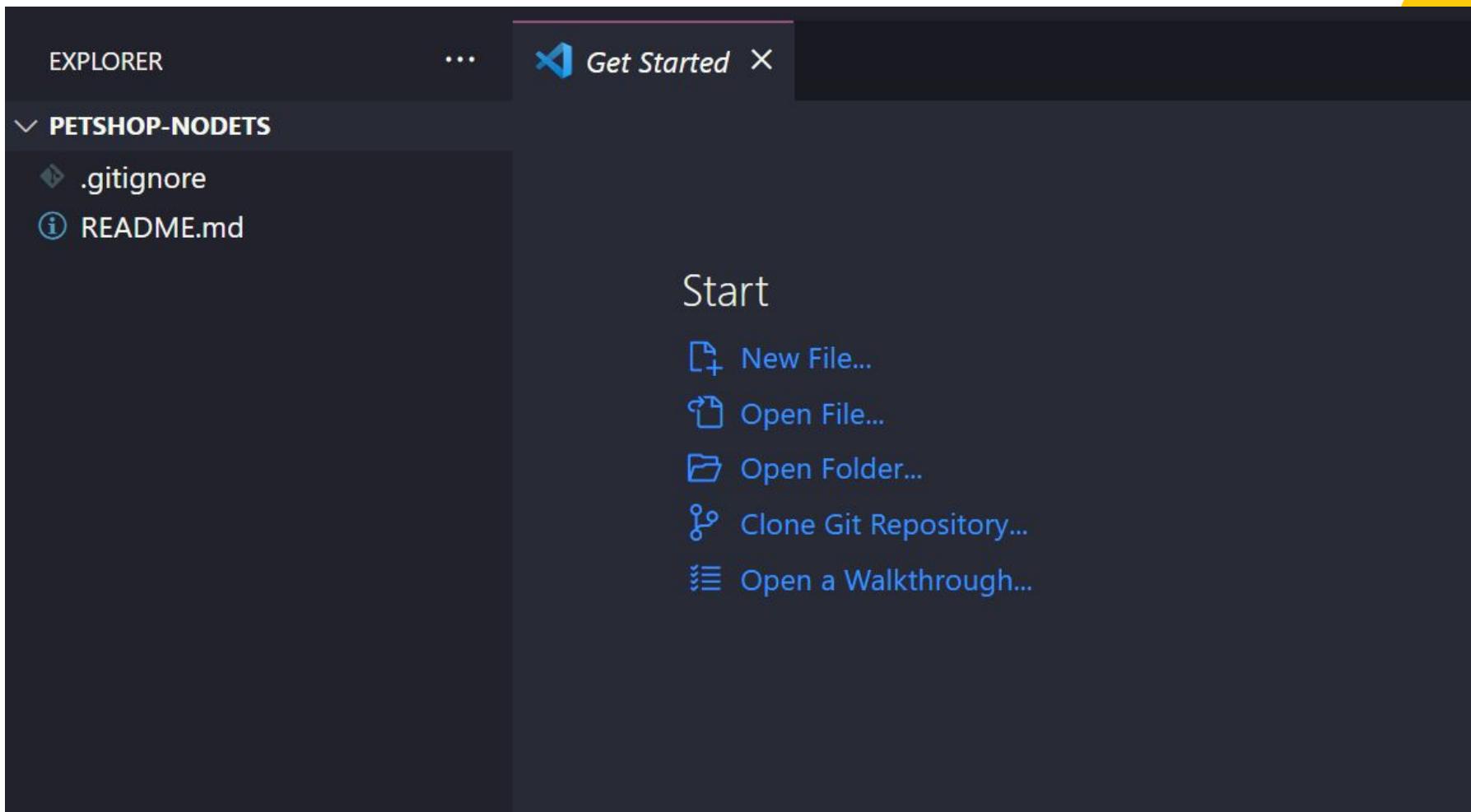
C:\Users\Viviane\Downloads\projeto>
```

## DIGITE GIT CLONE E COLE O ENDEREÇO DO SEU REPOSITÓRIO

```
C:\Users\Viviane\Downloads\projeto>git clone https://github.com/zennom/petshop-nodets
Cloning into 'petshop-nodets'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (4/4), done.
```

```
C:\Users\Viviane\Downloads\projeto>
```

# ABRA O ARQUIVO PETSHOP-NODETS NO VS CODE



**DIGITE NPM INIT E DÊ ENTER NAS INFORMAÇÕES**

EXPLORER

•

 Get Started 

## Start

## Recent

## PETSHOP-NODETS

## ❖ .gitignore

{ } package.json



Veja que o package.json  
foi criado

 New File...

 Open File...

 Open Folder...

 Clone Git Repository...

 Open a Walkthrough...

## PROBLEMS

## OUTPUT

TERMINAL

DEBUG CONSOLE

```
PS C:\Users\Viviane\Downloads\projeto\petshop-nodets> npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.
```

See `npm help init` for definitive documentation on these fields and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and save it as a dependency in the package.json file.

Press ^C at any time to quit.

package name: (petshop-nodets)

version: (1.0.0)

> OUTLINE

PARA CRIAR O ARQUIVO DE CONFIGURAÇÃO DO TYPESCRIPT DIGITE **TSC --INIT**

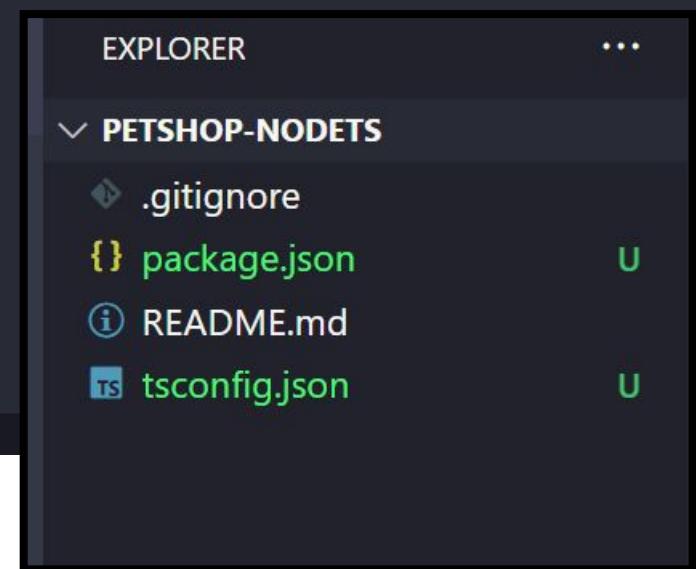
```
PS C:\Users\Viviane\Downloads\projeto\petshop-nodets> tsc --init
```

Created a new tsconfig.json with:

```
target: es2016
module: commonjs
strict: true
esModuleInterop: true
skipLibCheck: true
forceConsistentCasingInFileNames: true
```

You can learn more at <https://aka.ms/tsconfig.json>

```
PS C:\Users\Viviane\Downloads\projeto\petshop-nodets>
```



## VA EM TSCONFIG.JSON E FAÇA AS SEGUINTE ALTERAÇÕES

ts tsconfig.json U X

ts tsconfig.json > {} compilerOptions

```
13     /* Language and Environment */
14     "target": "es6",                                     /* Set t
15     // "lib": [],                                         /* Sp
16     // "jsx": "preserve",                                /* Sp
17     // "experimentalDecorators": true,                   /* En
18     // "emitDecoratorMetadata": true,                    /* Em
19     // "jsxFactory": "",                                /* Sp
20     // "jsxFragmentFactory": "",                        /* Sp
21     // "jsxImportSource": "",                           /* Sp
22     // "reactNamespace": "",                           /* Sp
23     // "noLib": true,                                    /* Di
24     // "useDefineForClassFields": true,                  /* Em
25
26     /* Modules */
27     "module": "commonjs",                               /* Sp
28     "outDir": "./dist",                                 /* Sp
29     "rootDir": "./src",                                /* Sp
30     "moduleResolution": "node",                         /* Speci
31     // "baseUrl": "./"                                /* Sp
```

VAMOS INSTALAR AS DEPENDÊNCIAS PARA O NOSSO PROJETO ABRA O TERMINAL  
E INSTALE AS DEPENDÊNCIAS ABAIXO:



# EXPRESS MUSTACHE-EXPRESS DOTENV

```
You can learn more at https://aka.ms/tsconfig.json
PS C:\Users\Viviane\Downloads\projeto\petshop-nodets> npm install express mustache-express dotenv

added 56 packages, and audited 57 packages in 19s

2 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS C:\Users\Viviane\Downloads\projeto\petshop-nodets>
```

VAMOS INSTALAR OS TYPES DAS NOSSAS DEPENDÊNCIAS :

--SAVE-DEV **@TYPES/EXPRESS**  
**@TYPES/MUSTACHE-EXPRESS**  
**@TYPES/NODE**

```
PS C:\Users\Viviane\Downloads\projeto\petshop-nodets> npm install --save-dev @types/express @types/mustache-express @types/node
added 11 packages, and audited 68 packages in 6s

2 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS C:\Users\Viviane\Downloads\projeto\petshop-nodets> []
```

PARA CONSEGUIRMOS USAR O SERVIDOR VAMOS INSTALAR O NODEMON E O TYPESCRIPT



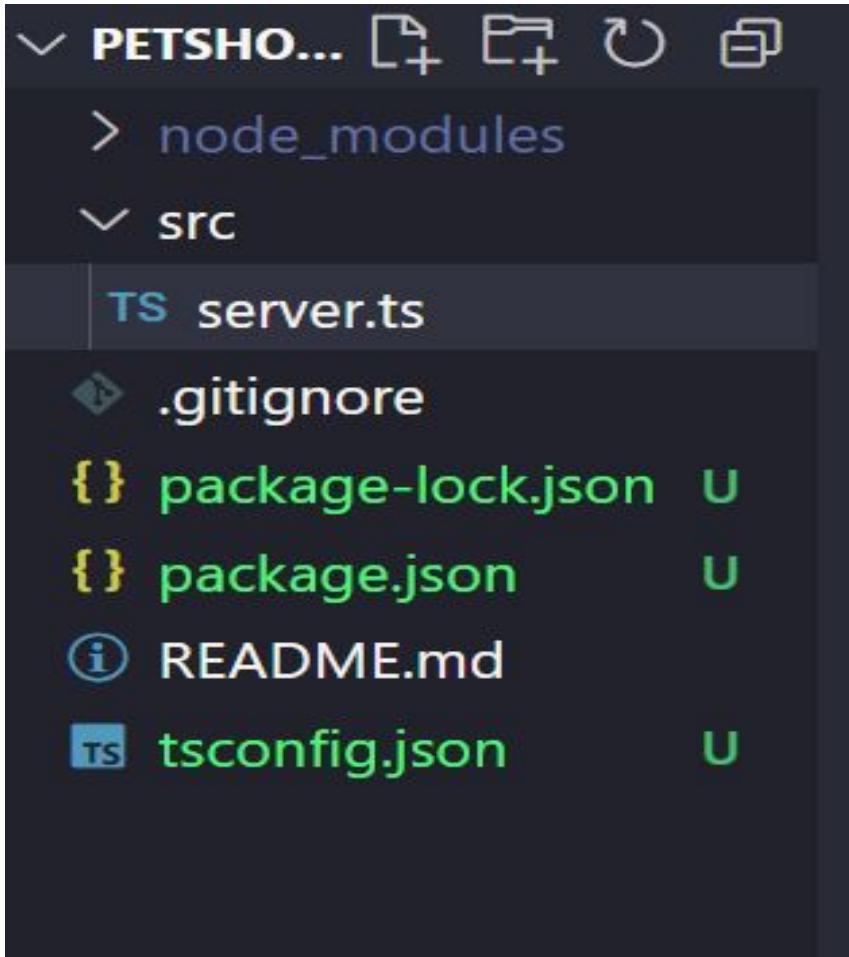
## -G NODEMON TYPESCRIPT TS-NODE

```
PS C:\Users\Viviane\Downloads\projeto\petshop-nodets> npm install -g nodemon typescript ts-node
changed 133 packages, and audited 134 packages in 8s
16 packages are looking for funding
  run `npm fund` for details
found 0 vulnerabilities
PS C:\Users\Viviane\Downloads\projeto\petshop-nodets> []
```

# VEJA AS DEPENDENCIAS E OS TYPES INSTALADOS NO PACKAGE.JSON

```
{  
  "name": "petshop-nodets",  
  "version": "1.0.0",  
  "description": "",  
  "main": "index.js",  
  "scripts": {  
    "test": "echo \\\"Error: no test specified\\\" && exit 1"  
  },  
  "repository": {  
    "type": "git",  
    "url": "git+https://github.com/zennom/petshop-nodets.git"  
  },  
  "author": "",  
  "license": "ISC",  
  "bugs": {  
    "url": "https://github.com/zennom/petshop-nodets/issues"  
  },  
  "homepage": "https://github.com/zennom/petshop-nodets#readme",  
  "dependencies": {  
    "dotenv": "^16.0.0",  
    "express": "^4.17.3",  
    "mustache-express": "^1.3.2"  
  },  
  "devDependencies": {  
    "@types/express": "^4.17.13",  
    "@types/mustache-express": "^1.2.1",  
    "@types/node": "^17.0.19"  
  }  
}
```

CRIE A PASTA SRC E DENTRO DELA CRIE O SERVER.TS



# AINDA EM PACKAGE.JSON VAMOS CRIAR O NOSSO SCRIPT PARA INICIAR NOSSO SERVIDOR

```
{  
  "name": "petshop-nodets",  
  "version": "1.0.0",  
  "description": "",  
  "main": "index.js",  
  ▷ Debug  
  "scripts": {  
    "test": "echo \\"$Error: no test specified\\" && exit 1",  
    "start-dev": "nodemon -e ts,json,mustache src/server.ts"  
  },  
  "repository": {  
    "type": "git",  
    "url": "git+https://github.com/zennom/petshop-nodets.git"  
  },  
  "author": "",  
  "license": "ISC",  
  "bugs": {  
    "url": "https://github.com/zennom/petshop-nodets/issues"  
  },  
  "homepage": "https://github.com/zennom/petshop-nodets#readme",  
}
```

Não esqueça de colocar as extensões  
que o nodemon vai monitorar

# VÁ NO README.MD COLOCAR AS INSTRUÇÕES PARA RODAR O PROJETO



EXPLORER    ...

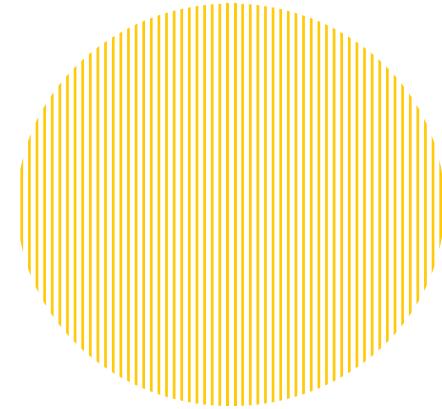
① README.md M X

① README.md > abc # petshop-nodets > abc ### Para rodar o projeto

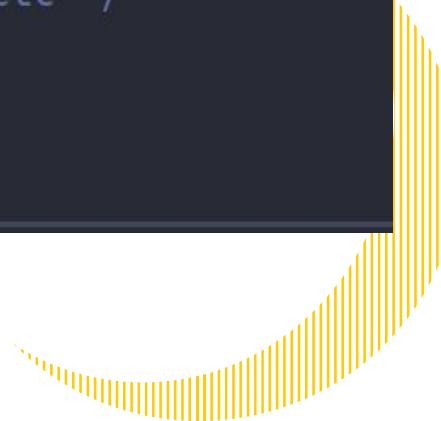
```
1 # petshop-nodets
2 Projeto feito em nodejs no de Desenvolvimento de Sistemas
3
4 #### Pré-requisitos globais:
5 `npm i -g nodemon typescript ts-node`
6
7 #### Instalação
8 `npm install`
9
10 #### Para rodar o projeto
11 `npm run start-dev`
```



# CRIANDO O SERVIDOR



## IMPORTE AS BIBLIOTECAS QUE VAMOS UTILIZAR



EXPLORER    ...    ⓘ README.md M    TS server.ts U X

▼ PETSHO... [+] ⏮ ⏹ ⏺ ⏻    src > TS server.ts

- > node\_modules
- ✓ src ●
  - TS server.ts U
- ↳ .gitignore
- { } package-lock.json U
- { } package.json U
- ⓘ README.md M

```
1 import express from 'express'
2 import dotenv from 'dotenv'
3 import mustache from 'mustache-express'
4 /* vamos por o path para conseguirmos usar a
5 pasta public e colocar html,css etc */
6 import path from 'path'
```

# CRIE O ARQUIVO .ENV E COLOQUE A PORTA QUE VOCÊ QUISER

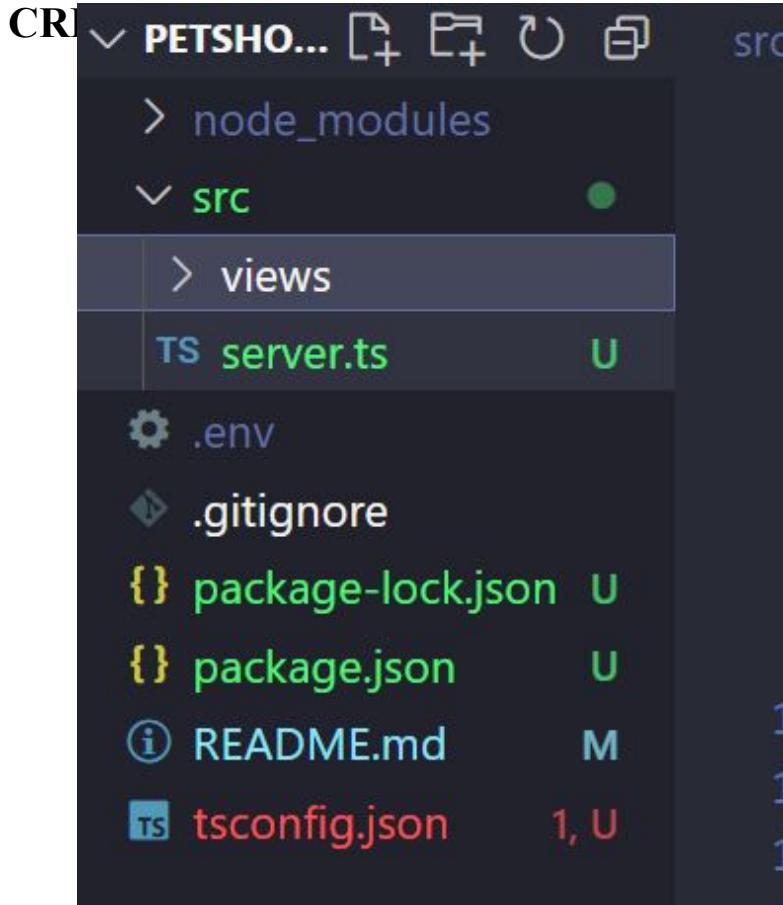


A screenshot of a dark-themed code editor interface, likely Visual Studio Code, showing a file named ".env" in the top right corner. The ".env" file is open in the center, displaying a single line of code: "PORT=3000". The left sidebar shows a project structure with files like "node\_modules", "src", "server.ts", ".env", ".gitignore", "package-lock.json", "package.json", and "README.md". The ".env" file in the sidebar is also highlighted.

# VOLTE EM SERVER.TS E VAMOS CONTINUAR AS CONFIGURAÇÕES

```
① README.md M    TS server.ts U X    .env
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
6 //dotenv para nossa porta
7 dotenv.config()
8
9 //configurando servidor
10 const server = express()
11
12 //configurando o mustache
13 server.set('view engine','mustache')
14 //diretório em que colocaremos nossos arquivos mustache
15 server.set('views',path.join(__dirname, 'views'))
16
```

COMO JÁ CONFIGURAMOS O MUSTACHE E COLOCAMOS O NOME DA NOSSA PASTA,



## E POR ÚLTIMO VAMOS SETAR O TEMPLATE ENGINES DO MUSTACHE

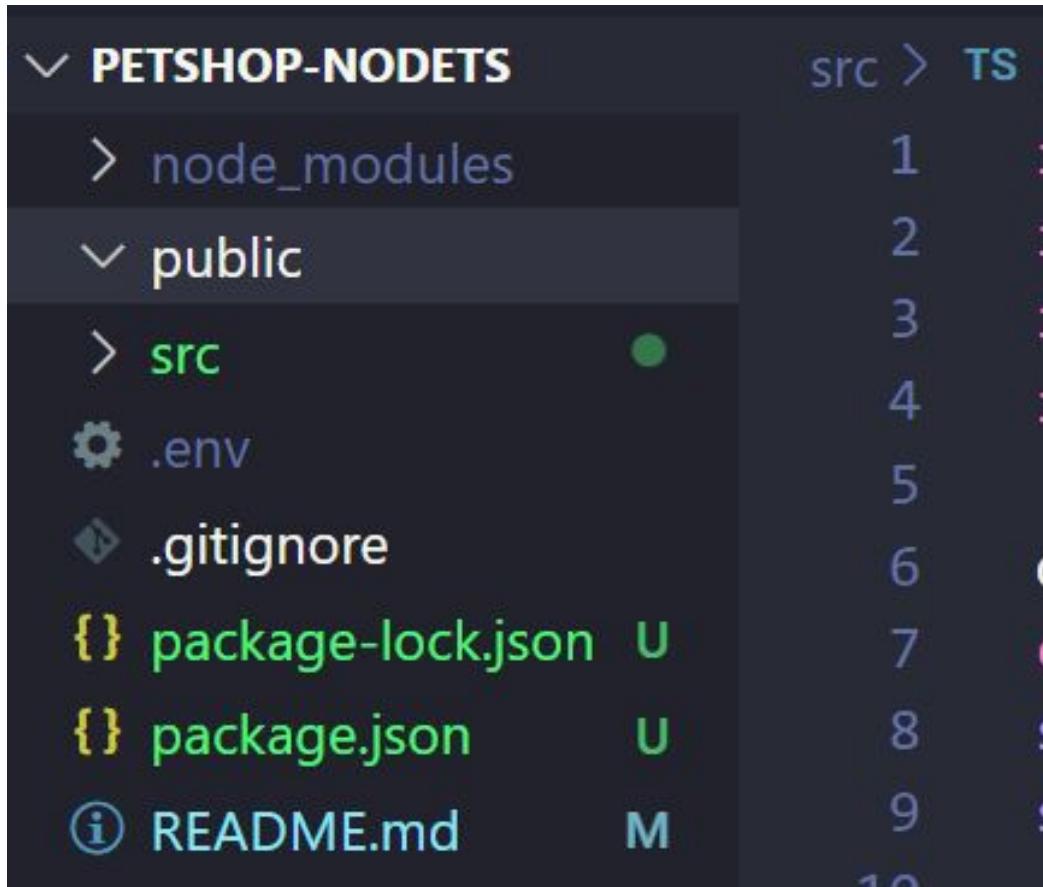
① README.md M    TS server.ts U X    .env

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
6  dotenv.config()
7  const server = express()
8  server.set('view engine', 'mustache')
9  server.set('views', path.join(__dirname, 'views'))
10 //template engines do mustache
11 server.engine('mustache', mustache())
12
```

# AGORA VAMOS CONFIGURAR A PASTA PÚBLICA, CRIE UMA NOVA PASTA NA RAÍZ DO PROJETO CHAMADA DE PUBLIC



```
PETSHOP-NODETS
  > node_modules
  < public
    > src
  .env
  .gitignore
  {} package-lock.json
  {} package.json
  ⓘ README.md
```

DENTRO DESSA PASTA  
TEREMOS O CSS, IMAGENS ETC

# CONFIGURAR PÁGINA DE ARQUIVOS ESTÁTICOS



```
① README.md M    TS server.ts U X   ⚙ .env
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
6 dotenv.config()
7 const server = express()
8 server.set('view engine', 'mustache')
9 server.set('views', path.join(__dirname, 'views'))
10 server.engine('mustache', mustache())
11
12 //diretorio da nossa pasta estática
13 server.use(express.static(path.join(__dirname, '../public')))
```

# COLOCAR O SERVIDOR PARA RODAR

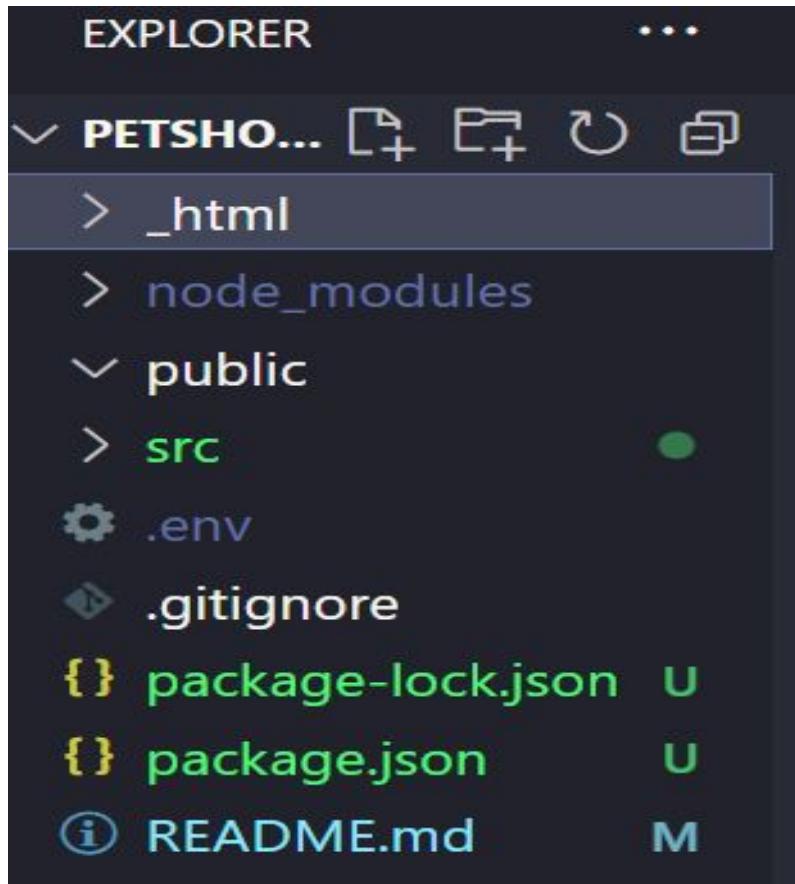
① README.md M

TS server.ts U X

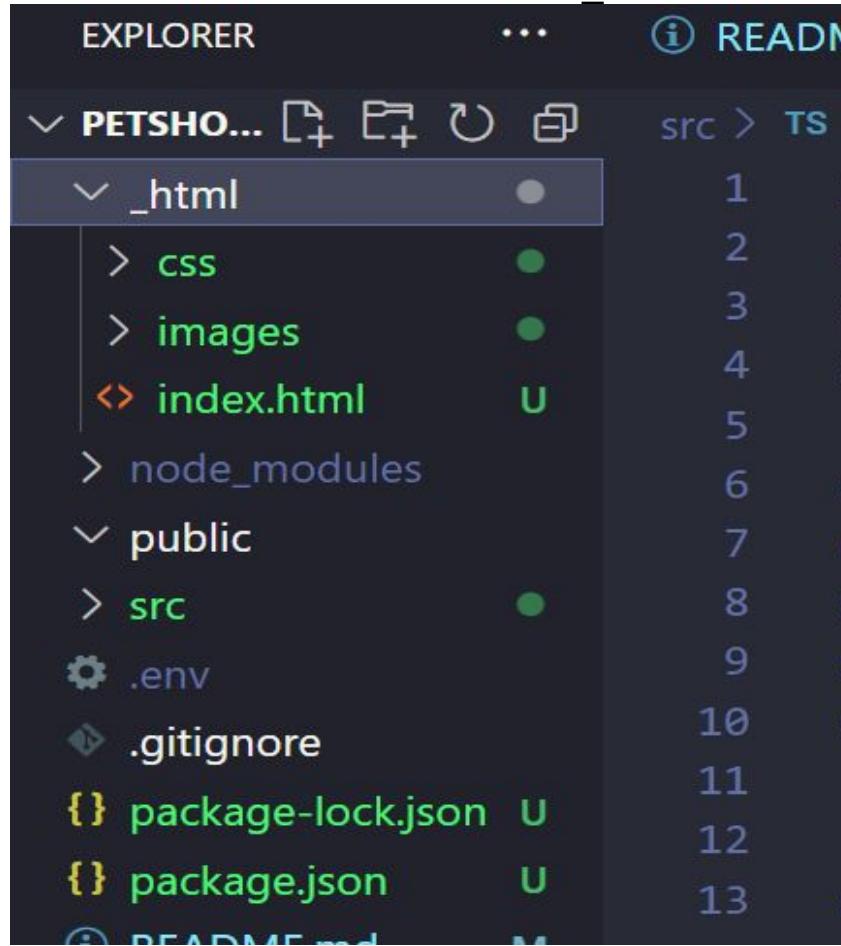
⚙ .env

```
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
6 dotenv.config()
7 const server = express()
8 server.set('view engine','mustache')
9 server.set('views',path.join(__dirname, 'views'))
10 server.engine('mustache',mustache())
11
12
13 server.use(express.static(path.join(__dirname,'../public'))))
14
15 //habilitando servidor
16 server.listen(process.env.PORT)
17 |
```

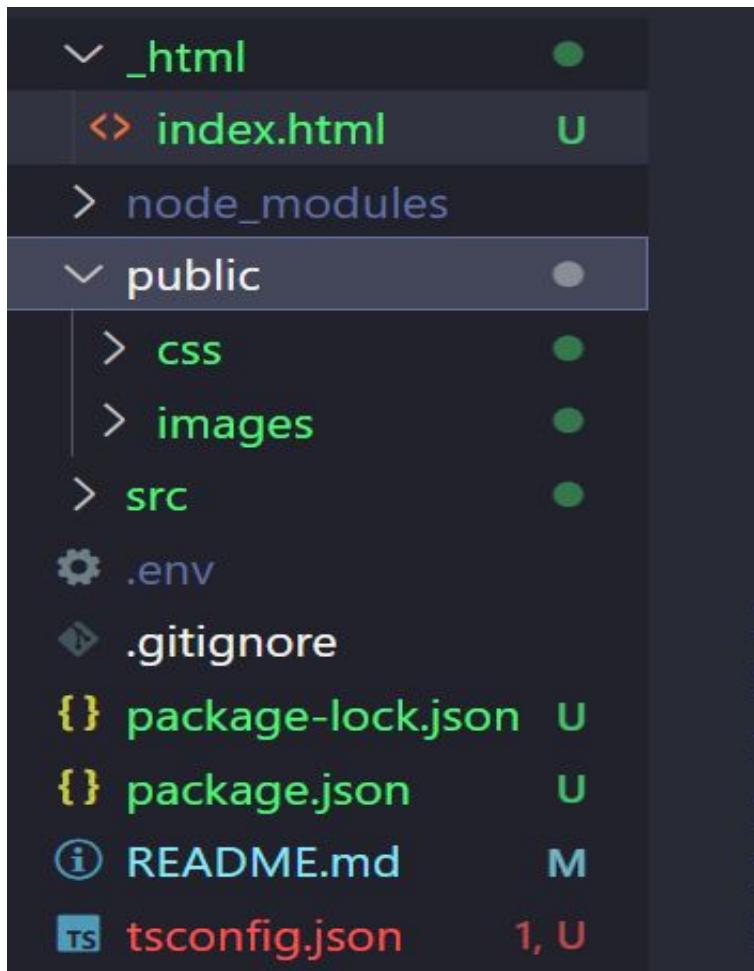
VAMOS CRIAR UMA PASTA DE AUXILIAR CHAMADA \_HTML (PASTA TEMPORÁRIA)  
(VAMOS POR O NOSSO ARQUIVO HTML DE AUXÍLIO)



**PEGUE OS ARQUIVOS QUE VOCÊ BAIXOU NO INICIO DA AULA E COLOQUE  
DENTRO DESSA PASTA HTML**



EM SEGUIDA PEGUE OS ARQUIVOS CSS E IMAGES E COLOQUE NA PASTA PUBLIC



# CRIANDO AS ROTAS E OS CONTROLLERS



DENTRO DA PASTA SRC CRIE UMA NOVA PASTA CHAMADA ROUTES E DENTRO DE ROUTES CRIE O ARQUIVO CHAMADO INDEX.TS



# CRIANDO AS ROTAS

The screenshot shows the Visual Studio Code interface with the title "CRIANDO AS ROTAS". The Explorer sidebar on the left displays the project structure of a Node.js application named "PETSHOP". The "src" directory contains "routes", "views", and "server.ts" files. The ".env" and ".gitignore" files are also listed. The "index.ts" file in the "routes" directory is currently open in the editor.

```
TS index.ts U ×
src > routes > TS index.ts > ...
1 import { Router } from 'express'
2
3 //criando a rota
4 const router = Router()
5
6 //criando a rota home
7 router.get('/',(req,res)=>{
8     res.send('home')
9 })
10
11
12 //exportando a rota
13 export default router
14
```

# VÁ EM SERVER.TS E IMPORTE A ROTA QUE CRIAMOS

TS index.ts U X

TS server.ts U X

```
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  //importando a mainRoutes
6  import mainRoutes from './routes/index'
7
8  dotenv.config()
9  const server = express()
10 server.set('view engine','mustache')
11 server.set('views',path.join(__dirname, 'views'))
12 server.engine('mustache',mustache())
13
14 server.use(express.static(path.join(__dirname,'../public')))
15
16 //criando a mainRoutes para ser usada
17 server.use(mainRoutes)
18
19 //já vamos aproveitar e criar a pagina not found
20 server.use((req,res)=>{
21   res.send('Página não encontrada!')
22 })
23
24 server.listen(process.env.PORT)
```

# RODE O PROJETO E VEJA SE ELE ESTÁ FUNCIONANDO!

TS index.ts U    TS server.ts U X

```
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'
```

PROBLEMS    OUTPUT    TERMINAL    DEBUG CONSOLE

```
PS C:\Users\Viviane\Downloads\projeto\petshop-nodets> npm run start-dev
```

```
> petshop-nodets@1.0.0 start-dev
> nodemon -e ts,json,mustache src/server.ts
```

```
[nodemon] 2.0.15
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): ***!
[nodemon] watching extensions: ts,json,mustache
[nodemon] starting `ts-node src/server.ts`
```



# LOCALHOST:3000

←

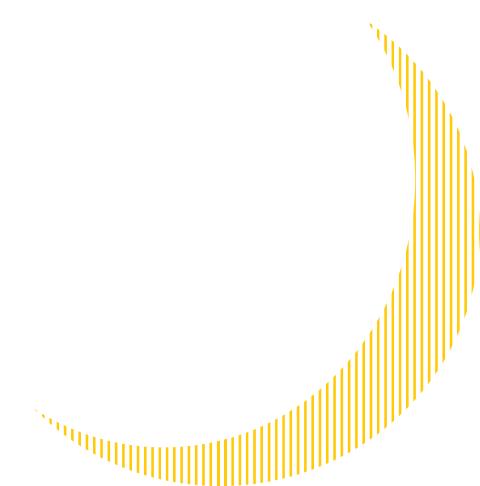
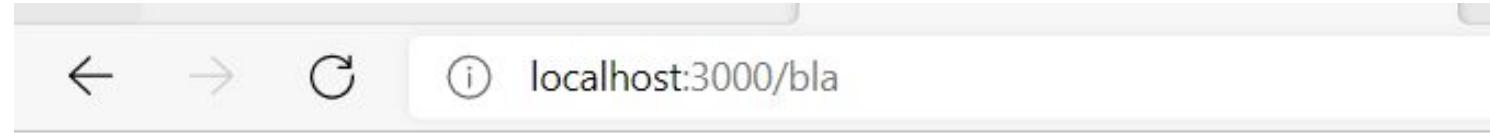
→



localhost:3000

home

**VEJA SE A PAGINA NOT FOUND ESTÁ FUNCIONANDO TAMBÉM**



AGORA VAMOS CRIAR OS CONTROLLERS, CRIE A PASTA CONTROLLERS DENTRO DE



# APROVEITE E CRIE TAMBÉM A PASTA MODELS DENTRO DE SRC

The screenshot shows a Visual Studio Code interface. The Explorer sidebar on the left displays a project structure:

- PETSHOP (root)
- \_html
- node\_modules
- public
- src
  - controllers
    - models
  - routes
    - index.ts
  - views
- server.ts
- .env

The 'models' folder under 'src/controllers' is currently selected. The Editor pane on the right shows the content of the 'index.ts' file:

```
src > TS server.ts > ...
1 import express
2 import dotenv
3 import mustache
4 import path
5 import mainRouter 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.__engine)
12
13 server.use(express.json())
14
15 const port = process.env.PORT || 3001
16
17 server.listen(port, () => {
18   console.log(`Server running on port ${port}`)
19 })
```

# QUANTAS ROTAS DEVEREMOS CRIAR?

PetShop Recanto dos Pets

Pesquise por raça

Todos

Cachorros

Gatos

Peixes



Todos os animais

5 ROTAS!

**ENTRETANTO, TODAS AS PÁGINAS SÃO IGUAIS, O QUE VAI MUDAR SERÁ O CONTEUDO DELAS  
VAMOS CRIAR UM CONTROLLER PARA TODAS AS PÁGINAS E UM CONTROLLER PARA A BUSCA**

### **Todos os animais disponíveis para adoção**



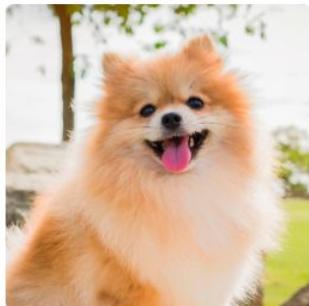
**Pastor-alemão**

Cor: Amarelo e Preto  
Gênero: Masculino



**Labrador-retriever**

Cor: Branco  
Gênero: Masculino



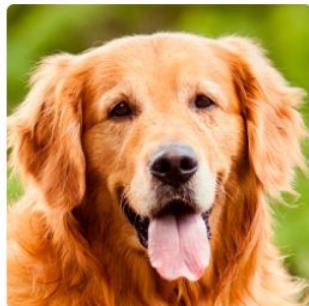
**Zwergspitz**

Cor: Amarelo  
Gênero: Feminino



**Husky Siberiano**

Cor: Branco e Preto  
Gênero: Masculino



**Golden Retriever**

Cor: Amarelo  
Gênero: Masculino



# CRIE OS DOIS CONTROLLERS (PAGECONTROLLER E SEARCH CONTROLLER)



```
✓ PETSHO... [+] [+] ⌂ ⌂ 1 src > controllers > TS searchController.ts
  > _html ●
  > node_modules
  > public ●
  < src ●
    < controllers ●
      TS pageControll... U
      TS searchControl... U
    > models
    > routes ●
    < views
      TS server.ts U
```

# FAZENDO O PAGECONTROLLER

TS index.ts U    TS server.ts U    TS pageController.ts U X    TS searchController.ts U

src > controllers > TS pageController.ts > [?] home

```
1 import {Request, Response} from 'express'  
2  
3 /*criando a rota, não vamos colocar o render ainda  
4 pois não temos a parte visual da página */  
5 export const home = (req:Request, res: Response) =>{  
6  
7     //res.render('pages/page')  
8  
9 }
```

## FAZENDO O SEARCHCONTROLLER

TS index.ts U

TS pageController.ts U

TS searchController.ts U X

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

```
1 import {Request, Response} from 'express'  
2  
3 export const search = (req:Request, res: Response) =>{  
4  
5     //res.render('pages/search')  
6  
7 }
```

# CRIANDO AS ROTAS EM PAGECONTROLLER

TS index.ts U

TS pageController.ts U X

TS searchController.ts U

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

```
1 import {Request, Response} from 'express'
2
3
4 export const home = (req:Request, res: Response) =>{
5 }
6
7 export const dogs = (req:Request, res: Response) =>{
8 }
9
10 export const cats = (req:Request, res: Response) =>{
11 }
12
13 export const fishes = (req:Request, res: Response) =>[
14 ]
```

# IMPORTANDO AS DUAS ROTAS EM INDEX.TS E CRIANDO AS ROTAS

TS index.ts U X

TS pageController.ts U

TS searchController.ts U

src > routes > TS index.ts > ...

```
1 import { Router } from 'express'
2 //importando os controllers
3 import * as PageController from '../controllers/pageController'
4 import * as SearchController from '../controllers/searchController'
5
6 const router = Router()
7
8 //criando as rotas
9 router.get('/',PageController.home)
10 router.get('/dogs',PageController.dogs)
11 router.get('/cats',PageController.cats)
12 router.get('/fishes',PageController.fishes)
13
14
15 export default router
16
```

# FAÇA A QUINTA ROTA QUE É A DA BUSCA

rs index.ts U X

TS pageController.ts U

TS searchController.ts U

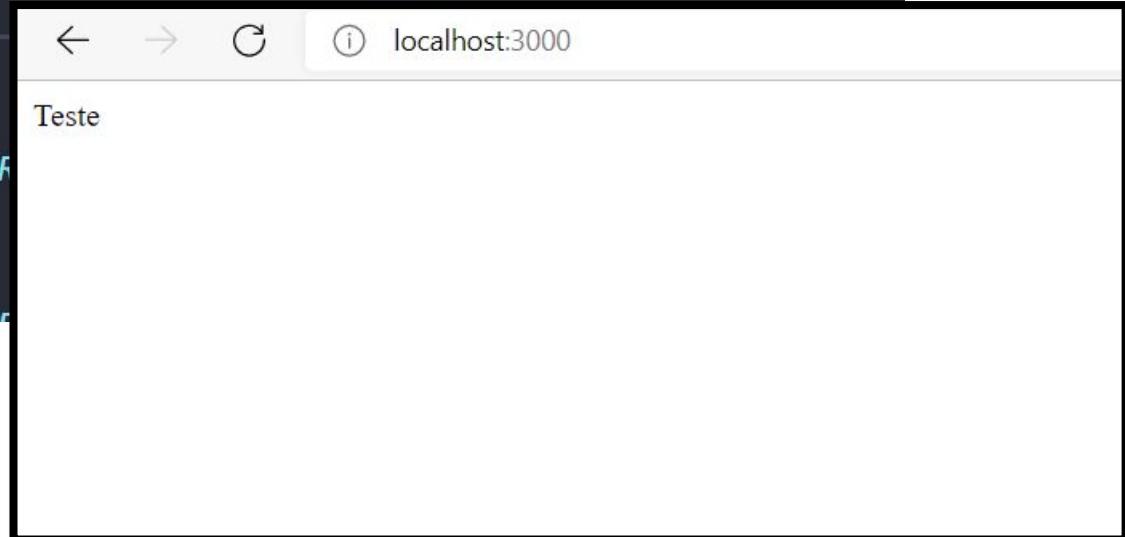
src > routes > TS index.ts > ...

```
1 import { Router } from 'express'
2 import * as PageController from '../controllers/pageController'
3 import * as SearchController from '../controllers/searchController'
4
5 const router = Router()
6
7 router.get('/',PageController.home)
8 router.get('/dogs',PageController.dogs)
9 router.get('/cats',PageController.cats)
10 router.get('/fishes',PageController.fishes)
11
12 //a quinta rota
13 router.get('/search',SearchController.search)
14
15 export default router
16 |
```

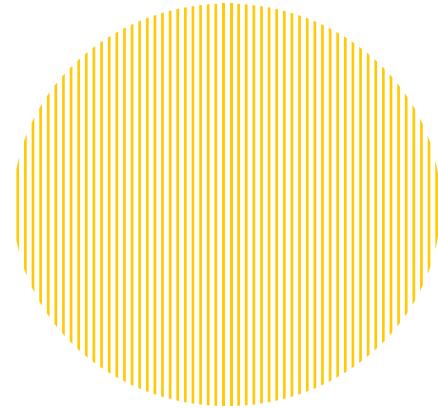
# TESTE SE ESTÁ TUDO FUNCIONANDO

src > controllers > **TS** pageController.ts U X TS searchController.ts U

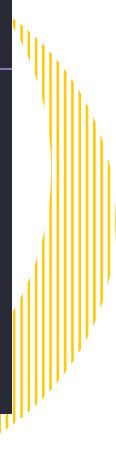
```
1 import {Request, Response} from 'express'
2
3
4 √ export const home = (req:Request, res: Response) =>{
5     res.send("Teste")
6 }
7
8 export const dogs = (req:R
9 }
10
11
```



# SEPARANDO AS VIEWS



VAMOS APROVEITAR QUE NOSSA BASE JÁ ESTÁ PRONTA E VAMOS DAR UM COMMIT NO GIT,  
FINALIZE O TERMINAL COM CTRL + C E DIGITE OS COMANDOS ABAIXO:



```
TS index.ts A    TS pageController.ts A X    TS searchController.ts A
src > controllers > TS pageController.ts > [?] home
1 import {Request, Response} from 'express'
2
3
4 export const home = (req:Request, res: Response) =>{
5     res.send("Teste")
6 }
7
8 export const dogs = (req:Request, res: Response) =>{
9 }
```

PROBLEMS    OUTPUT    TERMINAL    DEBUG CONSOLE

```
PS C:\Users\Viviane\Downloads\projeto\petshop-nodets> git add .
PS C:\Users\Viviane\Downloads\projeto\petshop-nodets> git commit -m "base criada"
```

## EM SEGUIDA DE UM **GIT PUSH** PARA ELE ALTERAR O SEU REPOSITÓRIO



```
TS index.ts          TS pageController.ts X    TS searchController.ts
src > controllers > TS pageController.ts > [x] home
1 import {Request, Response} from 'express'
2
3
4 √ export const home = (req:Request, res: Response) =>{
5     res.send("Teste")
6 }
7
8 export const dogs = (req:Request, res: Response) =>{
9 }
```

PROBLEMS    OUTPUT    TERMINAL    DEBUG CONSOLE

```
PS C:\Users\Viviane\Downloads\projeto\petshop-nodets> git push
Enumerating objects: 43, done.
Counting objects: 100% (43/43), done.
Delta compression using up to 12 threads
Compressing objects: 100% (38/38), done.
Writing objects: 100% (41/41), 4.63 MiB | 1.32 MiB/s, done.
Total 41 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/zennom/petshop-nodets
  c0210b3..7458e5e main -> main
PS C:\Users\Viviane\Downloads\projeto\petshop-nodets>
```

# VEJA SE O SEU REPOSITÓRIO FOI ALTERADO:

 zennom	base criada	7458e5e 1 minute ago	 2 commits
 _html	base criada	1 minute ago	
 public	base criada	1 minute ago	
 src	base criada	1 minute ago	
 .gitignore	Initial commit	2 hours ago	
 README.md	base criada	1 minute ago	
 package-lock.json	base criada	1 minute ago	
 package.json	base criada	1 minute ago	
 tsconfig.json	base criada	1 minute ago	
 README.md			

## petshop-nodets

Projeto feito em nodejs no de Desenvolvimento de Sistemas

### Pré-requisitos globais:

```
npm i -g nodemon typescript ts-node
```

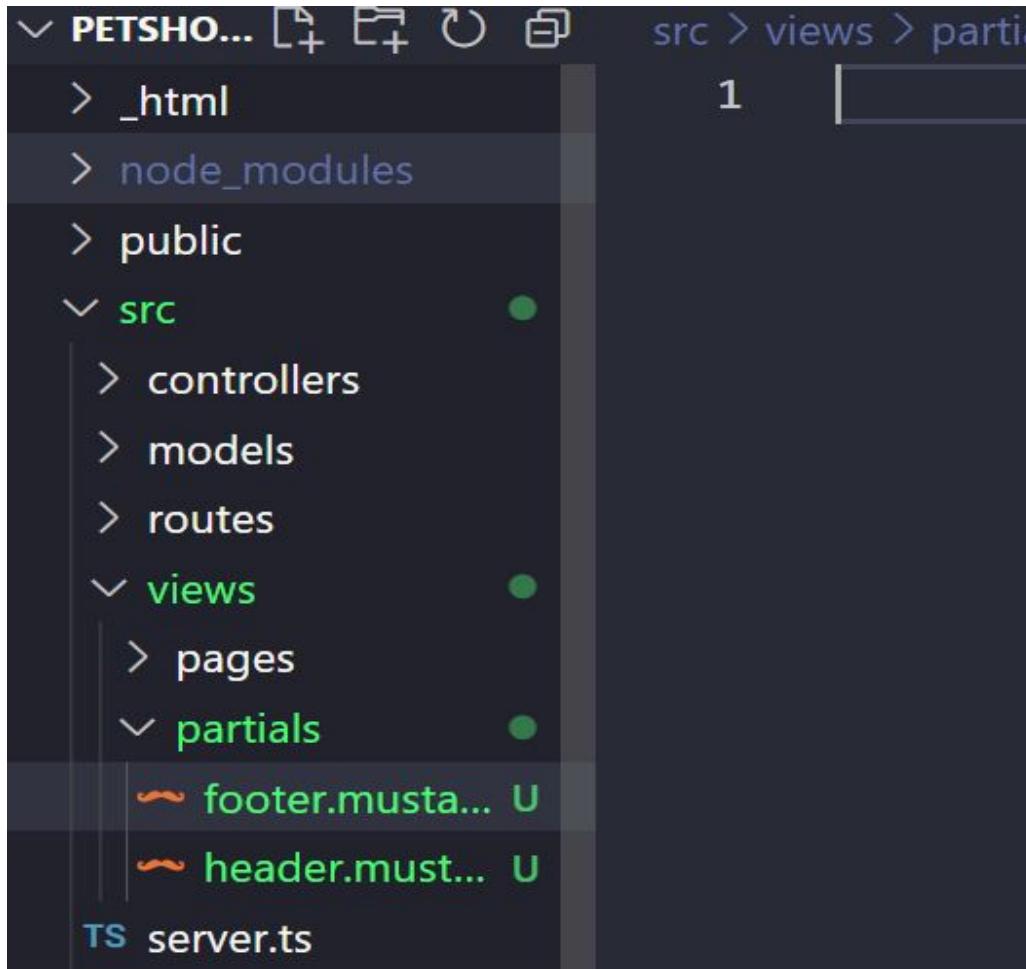
### Instalação

```
npm install
```

DENTRO DA PASTA VIEW CRIE DUAS PASTAS: PARTIALS E PAGES



DENTRO DA PASTA PARTIALS CRIE O HEADER.MUSTACHE E O FOOTER.MUSTACHE



# VÁ EM `HTML>INDEX E PEGUE O ATÉ O MENU (COPIE O CÓDIGO)`

```
~ header.mustache U ✖ ~ footer.mustache U ✖ ◊ index.html ✖
html > ◊ index.html > ...
1  <!DOCTYPE html>
2  <html lang="pt">
3  <head>
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="">
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="">Cachorros</a></li>
23         <li class=""><a href="">Gatos</a></li>
24         <li class=""><a href="">Peixes</a></li>
25     </ul>
26 </nav>
27 <section class="banner" style="background-image: url('images/allanimals.jpg')>Todos os animais</section>
28 <h2>Todos os animais disponíveis para adoção</h2>
```

# COLE EM HEADER.MUSTACHE

~ header.mustache U X

~ footer.mustache U

index.html

src > views > partials > ~ header.mustache

```
1   <!DOCTYPE html>
2   <html lang="pt">
3   <head>
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="">
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="">Cachorros</a></li>
23          <li class=""><a href="">Gatos</a></li>
24          <li class=""><a href="">Peixes</a></li>
25      </ul>
26  </nav>
```

# VAMOS FAZER ALGUMAS ALTERAÇÕES PARA NOSSA PÁGINA DE ACORDO COM O MUSTACHE

~ header.mustache U X

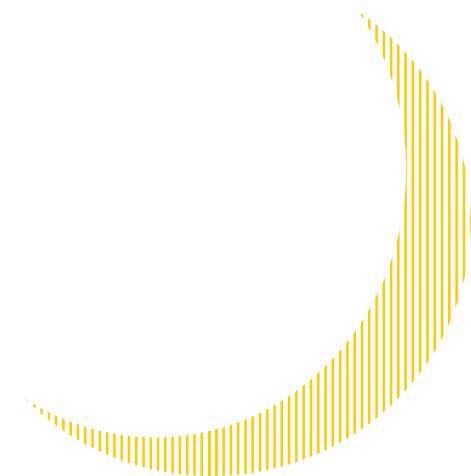
~ footer.mustache U

src > views > partials > ~ header.mustache

```
1  <!DOCTYPE html>
2  <html lang="pt">
3  <head>
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>
```

# AINDA EM HEADER.MUSTACHE COLOQUE OS LINKS DO NAV

```
~ header.mustache ✘ ~ footer.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>
```



**AGORA FAÇA A MESMA COISA COM O RODAPÉ DA PÁGINA, COLOQUE NA PARTIALS DO  
FOOTER.MUSTACHE**

## FOOTER.MUSTACHE

~ header.mustache U

~ footer.mustache U X

↔ index.html M

src > views > partials > ~ footer.mustache

```
1  <footer class="container">
2    Feito com ❤ pela TI
3  </footer>
4  </body>
5  </html>
```

## DENTRO DE PAGES CRIE O PAGES.MUSTACHE

```
▽ _html
  <> index.html      M
  > node_modules
  > public
  ▽ src
    > controllers
    > models
    > routes
    ▽ views
      ▽ pages
        ~ pages.musta... U
      ▽ partials
        ~ footer.musta... U
        ~ header.musta... U
```

## COPIE O RESTANTE DO LAYOUT QUE ESTÁ EM \_HTML PARA PAGES.MUSTACHE, IMPORTE AS PARTIALS DE HEADER E FOOTER

~ header.mustache U

~ footer.mustache U

~ pages.mustache U X

```
src > views > pages > ~ pages.mustache
1   {{>partials/header}}
2   <section class="banner" style="background-image: url('images/allanimals.jpg')>Todos os animais</section>
3   <h2>Todos os animais disponíveis para adoção</h2>
4
5   <div class="container list">
6       <div class="item">
7           
```

```

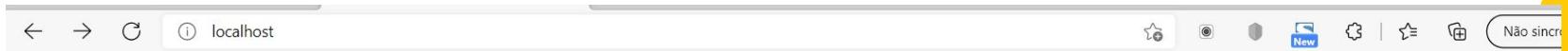
<div class="item--name">Tanictis</div>
<div class="item--color">Cor: Vermelho</div>
<div class="item--genre">Gênero: Masculino</div>
</div>
<div class="item">
    
    <div class="item--name">Acará Bandeira</div>
    <div class="item--color">Cor: Preto</div>
    <div class="item--genre">Gênero: Masculino</div>
</div>
</div>
{{>partials/footer}}
```

## VÁ EM PAGESCONTROLLER E DIRECIONE AS ROTAS

ts pageController.ts M X

```
src > controllers > ts pageController.ts > [e] fishes
1   import {Request, Response} from 'express'
2
3
4   √ export const home = (req:Request, res: Response) =>{
5     |   res.render("pages/pages")
6   }
7
8   √ export const dogs = (req:Request, res: Response) =>{
9     |   res.send("dogs")
10    }
11
12  √ export const cats = (req:Request, res: Response) =>{
13    |   res.send("cats")
14  }
15
16  √ export const fishes = (req:Request, res: Response) =>{
17    |   res.send("fishes")
18  }
```

# VEJA SE ESTÁ FUNCIONANDO



**Todos os animais disponíveis para adoção**



**Pastor-alemão**

Cor: Amarelo e Preto



**Labrador-retriever**

Cor: Branco



**Zwergspitz**

Cor: Amarelo



**Husky Siberiano**

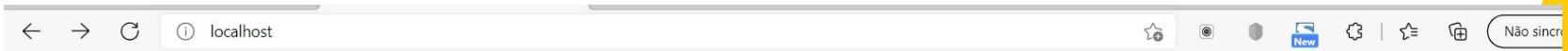
Cor: Branco e Preto



**Golden Retriever**

Cor: Amarelo

# AGORA PRECISAMOS ARRUMAR AS IMAGENS



PetShop Recanto dos Pets

Pesquise por raça

Todos

Cachorros

Gatos

Peixes

Todos os animais

Todos os animais disponíveis para adoção



**Pastor-alemão**

Cor: Amarelo e Preto



**Labrador-retriever**

Cor: Branco



**Zwergspitz**

Cor: Amarelo



**Husky Siberiano**

Cor: Branco e Preto



**Golden Retriever**

Cor: Amarelo

## REMOVA OS CARDS DA PAGES.MUSTACHE E DEIXE APENAS UM

TS pageController.ts M

~ pages.mustache U X

src > views > pages > ~ pages.mustache

```
1  {{>partials/header}}
2  <section class="banner" style="background-image: url('images/allanimals.jpg')>Todos os animais
3  <h2>Todos os animais disponíveis para adoção</h2>
4
5  <div class="container list">
6      <div class="item">
7          
8          <div class="item--name">Pastor-alemão</div>
9          <div class="item--color">Cor: Amarelo e Preto</div>
10         <div class="item--genre">Gênero: Masculino</div>
11     </div>
12 <div>
13 {{>partials/footer}}
```

# A PÁGINA IRÁ FICAR ASSIM

Todos

Cachorros

Gatos

Peixes

Todos os animais

Todos os animais disponíveis para adoção

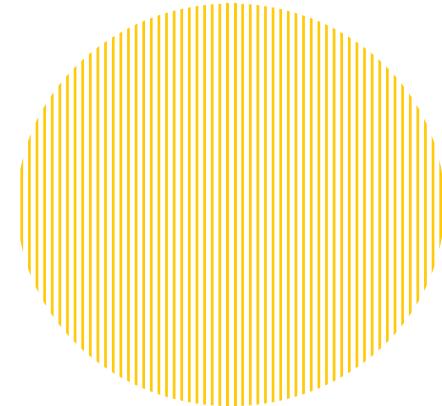


**Pastor-alemão**

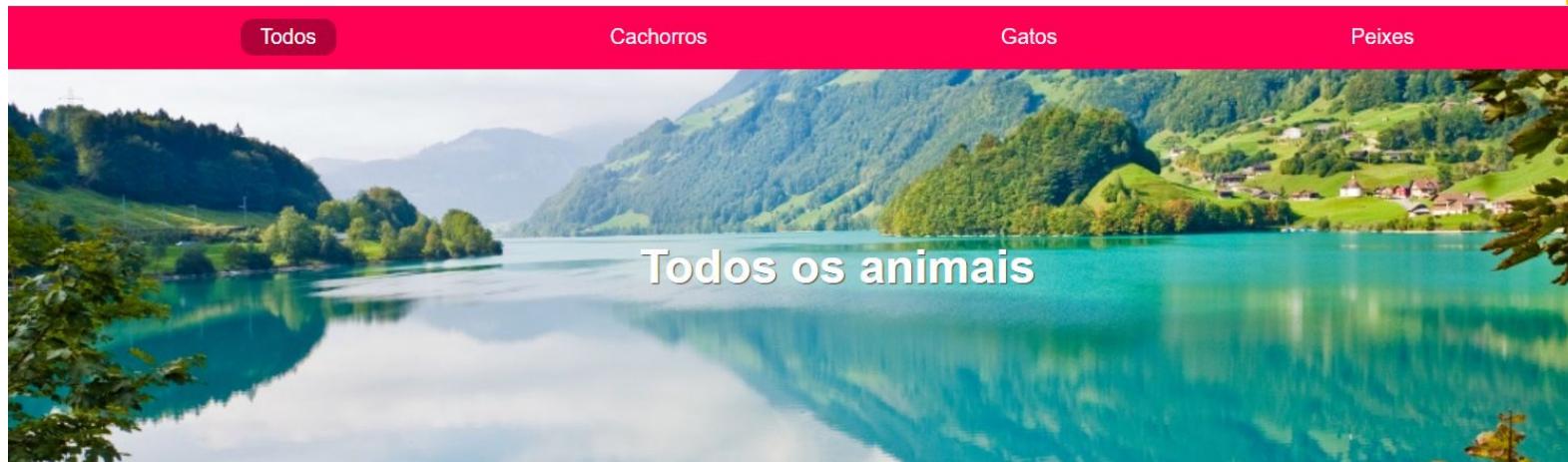
Cor: Amarelo e Preto  
Gênero: Masculino

Feito com ❤️ pela TI

# MODIFICANDO DADOS EM CADA PÁGINA



**AGORA VAMOS DEIXAR A PÁGINA DINÂMICA, TODA VEZ QUE MUDARMOS DE ROTA, A IMAGEM DE FUNDO MUDARÁ E O NOME DO TÍTULO TAMBÉM**



**Todos os animais disponíveis para adoção**



Feito com ❤️ pela TI

**Pastor-alemão**

Cor: Amarelo e Preto  
Gênero: Masculino

# ALTERANDO O BANNER E O TITULO DA PÁGINA HOME

```
TS pageController.ts M ✘ ~ pages.mustache U ~ footer.mustache U
src > controllers > TS pageController.ts > [o] fishes
  1 import {Request, Response} from 'express'
  2
  3
  4 export const home = (req:Request, res: Response) =>{
  5   res.render("pages/pages",{
  6     banner:{
  7       title:'Todos os animais',
  8       background:'allanimals.jpg'
  9     }
 10   })
 11 }
 12
 13 export const dogs = (req:Request, res: Response) =>{
 14   res.send("dogs")
 15 }
 16
 17 export const cats = (req:Request, res: Response) =>{
 18   res.send("cats")
 19 }
 20
 21 export const fishes = (req:Request, res: Response) =>{
 22   res.send("fishes")
 23 }
```

## VÁ EM PAGES.MUSTACHE E VAMOS POR O TITLE E O BACKGROUND



```
ts pageController.ts M ~ pages.mustache U X ~ footer.mustache U
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    <div class="item">
9      
10     <div class="item--name">Pastor-alemão</div>
11     <div class="item--color">Cor: Amarelo e Preto</div>
12     <div class="item--genre">Gênero: Masculino</div>
13   </div>
14 </div>
15
16  {{>partials/footer}}
```

Se eu tiver banner, ele irá aparecer

# ATUALIZE A PÁGINA

PetShop Recanto dos Pets

Pesquise por raça

Todos

Cachorros

Gatos

Peixes



Todos os animais

Todos os animais disponíveis para adoção



Feito com ❤️ pela TI

Pastor-alemão

Cor: Amarelo e Preto  
Sexo: Masculino

SE OS TEXTOS E A IMAGEM DE FUNDO ESTIVEREM APARECENDO ENTÃO ESTÁ OK

# AGORA FAÇA O TÍTULO E O BANNER DO CACHORRO, GATO E PEIXE

```
TS pageController.ts M X  ↵ pages.mustache U  ↵ footer.mustache U
src > controllers > TS pageController.ts > [o] home
1 import {Request, Response} from 'express'
2
3
4 export const home = (req:Request, res: Response) =>{
5   res.render("pages/pages",{
6     banner:{
7       title:'Todos os animais',
8       background:'allanimals.jpg'
9     }
10  })
11}
12
13 //coloque o banner do cachorro e o título do cachorro
14 export const dogs = (req:Request, res: Response) =>{
15   res.send("dogs")
16 }
17 //coloque o banner do gato o título do gato
18 export const cats = (req:Request, res: Response) =>{
19   res.send("cats")
20 }
21 //coloque o banner do peixe e o título do peixe
22 export const fishes = (req:Request, res: Response) =>{
23   res.send("fishes")
24 }
```

# RESOLUÇÃO:

```
ts pageController.ts M X ~ pages.mustache U ~ footer.mustache U
src > controllers > ts pageController.ts > [x] fishes
1 import {Request, Response} from 'express'
2
3
4 export const home = (req:Request, res: Response) =>{
5     res.render("pages/pages",{
6         banner:{
7             title:'Todos os animais',
8             background:'allanimals.jpg'
9         }
10    })
11 }
12 export const dogs = (req:Request, res: Response) =>{
13     res.render("pages/pages",{
14         banner:{
15             title:'Todos os cachorros',
16             background:'banner_dog.jpg'
17         }
18    })
19 }
20 export const cats = (req:Request, res: Response) =>{
21     res.render("pages/pages",{
22         banner:{
23             title:'Todos os gatos',
24             background:'banner_cat.jpg'
25         }
26    })
27 }
28 export const fishes = (req:Request, res: Response) =>[
29     res.render("pages/pages",{
30         banner:{
31             title:'Todos os peixes',
32             background:'banner_fish.jpg'
33         }
34     })
35 ]
```