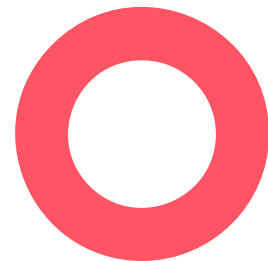


DESENVOLVIMENTO DE SISTEMAS

UC13

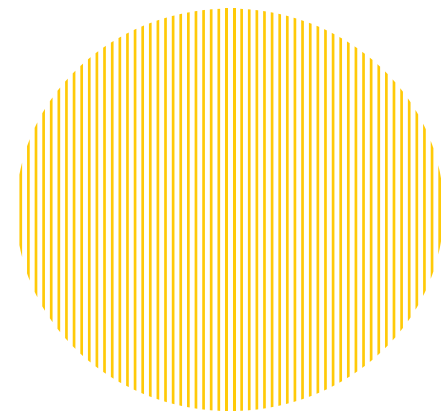
Prof. Viviane de Lima

viviane.lfrancelino@sp.senac.br

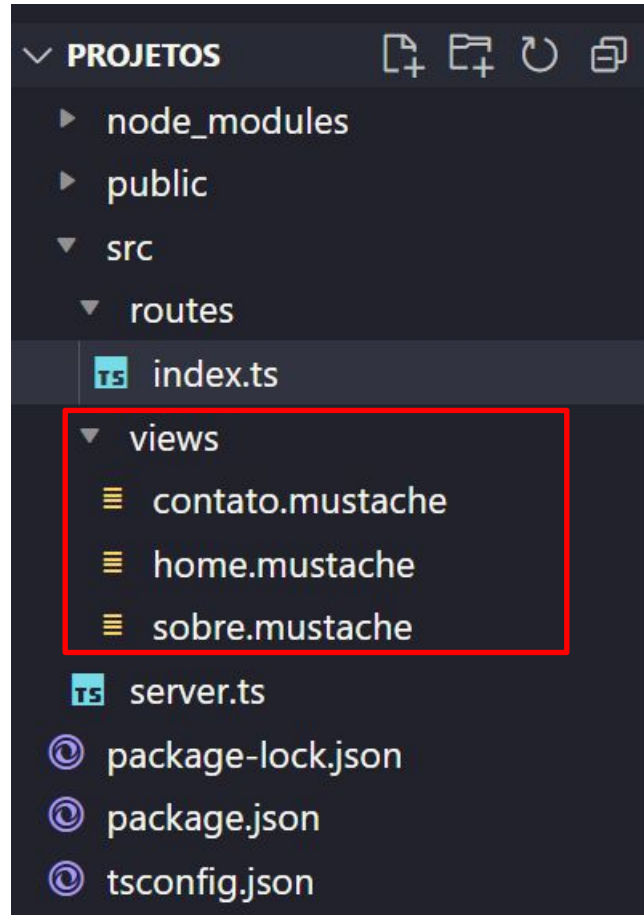


AULA 05

CONTEÚDO PARCIAL



CRIE AS PÁGINAS DE CONTATO E SOBRE



APÓS CRIAR, COLOQUE O CONTEUDO HTML NAS PÁGINAS




```
home.mustache × contato.mustache sobre.mustache ×
src > views > sobre.mustache
1  <html>
2  <head>
3    <title>Página Home</title>
4  </head>
5
6  <body>
7    <h1>Título da Página</h1>
8    <hr/>
9
10
11    <p> Olá, tudo bem?</p>
12
13  </body>
14  </html>
15
```



RENDERIZE AS PÁGINAS CONTATO E SOBRE NO INDEX.TS



```
home.mustache contato.mustache sobre.mustache ts index.ts X
src > routes > ts index.ts > ...
21         {title:'Produto w', price:50 }
22     ],
23     frases:[
24
25     ]
26 })
27 })
28
29 router.get('/contato',(req:Request, res:Response) =>{
30     res.render('contato')
31 })
32
33 router.get('/sobre',(req:Request, res:Response) =>{
34     res.render('sobre')
35 })
36
37 export default router
```



VAMOS SUPOR QUE EU PRECISE COLOCAR O CSS NO SITE OU FAZER ALGUMA ALTERAÇÃO EM TODAS AS PÁGINAS, NOSSO SITE TEM APENAS 3 PÁGINAS, PARA REALIZAR ALTERAÇÕES NOS ARQUIVOS COM 3 PÁGINAS NÃO IRIA DAR MUITO TRABALHO. PORÉM, E SE EU TIVESSE 30 PÁGINAS? SERIA PRODUTIVO FAZER ALTERAÇÃO PÁGINA POR PÁGINA?

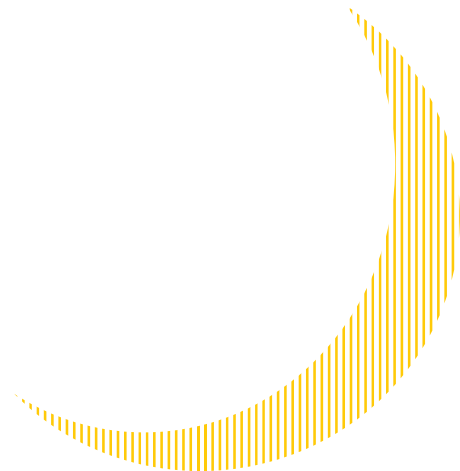
NÃO, POR ISSO EXISTE O CONTEUDO PARCIAL



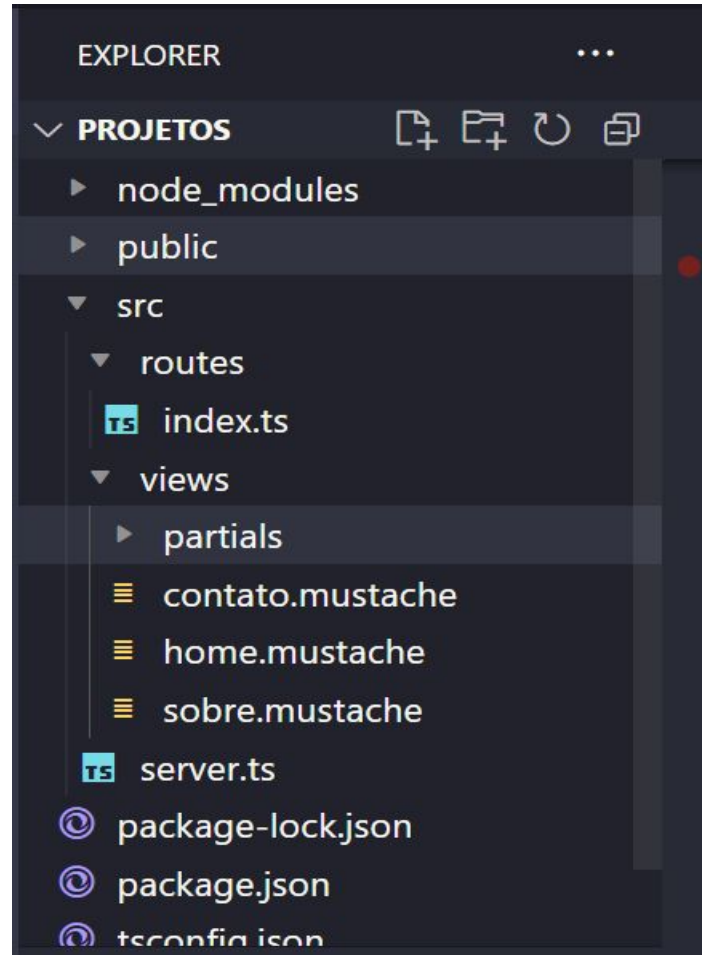
COMO O CONTEÚDO PARCIAL FUNCIONA?



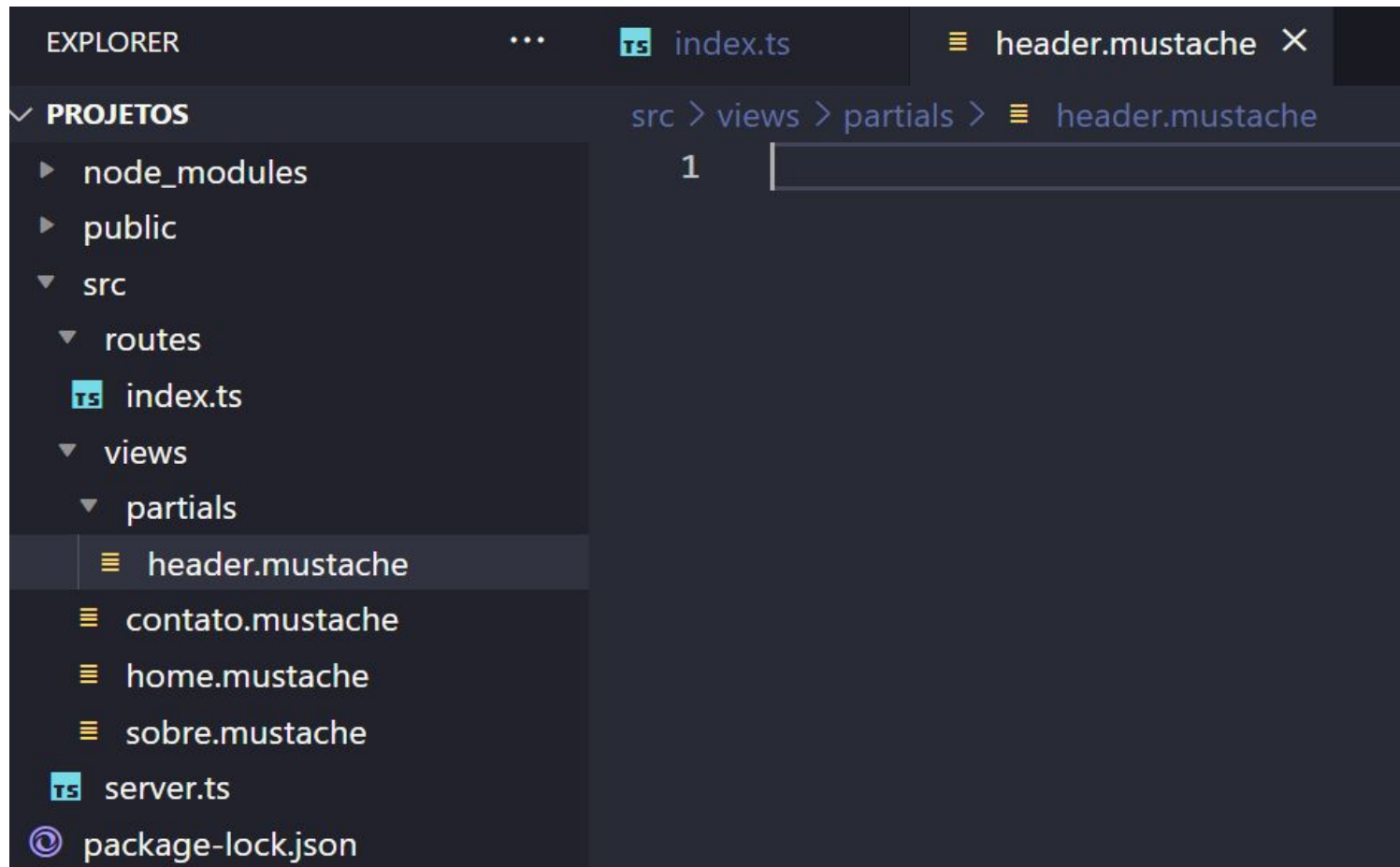
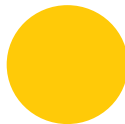
Criamos uma pasta nova dentro da pasta view e usamos essa pasta para colocar todo código que será utilizado em todas as páginas, desse modo realizamos o procedimento apenas uma vez, e não é necessário ficar replicando o código página por página



CRIE UMA PASTA CHAMADA PARTIALS DENTRO DA VIEW



CRIE UM ARQUIVO CHAMADO HEADER.MUSTACHE, VAMOS CRIAR UM CONTEUDO PARCIAL PARA O NOSSO CABEÇALHO



The image shows a screenshot of a code editor (VS Code) with a dark theme. The Explorer sidebar on the left shows a project structure under 'PROJETOS'. The file 'header.mustache' is selected and highlighted. The breadcrumb navigation at the top of the editor area shows the path 'src > views > partials > header.mustache'. The editor area itself is mostly empty, with a line number '1' visible on the left side of the text area.

```
EXPLORER
```

... index.ts header.mustache X

✓ PROJETOS

- node_modules
- public
- ▼ src
 - ▼ routes
 - index.ts
 - ▼ views
 - ▼ partials
 - header.mustache
 - contato.mustache
 - home.mustache
 - sobre.mustache
 - server.ts
 - package-lock.json

src > views > partials > header.mustache

1

DENTRO DE HEADER.MUSTACHE COLOQUE TODA A INFORMAÇÃO QUE SE REPETE EM CONTATO, HOME E SOBRE

header.mustache X

src > views > partials > header.mustache

```
1  <html>
2  <head>
3    <title>Página Home</title>
4  </head>
5
6    <h1>Título da Página</h1>
7    <hr/>
8
9  <body>
```

10

EM SEGUIDA COLOQUE A PARTIALS HEADER NAS PÁGINAS HOME,CONTATO E SOBRE

≡ contato.mustache X

≡ home.mustache

≡ sobre.mustache X

src > views > ≡ sobre.mustache

```
1  {{>partials/header}}
```

```
2
```

```
3  <html>
```

```
4  <head>
```

```
5    <title>Página Home</title>
```

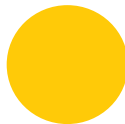
```
6  </head>
```

```
7
```

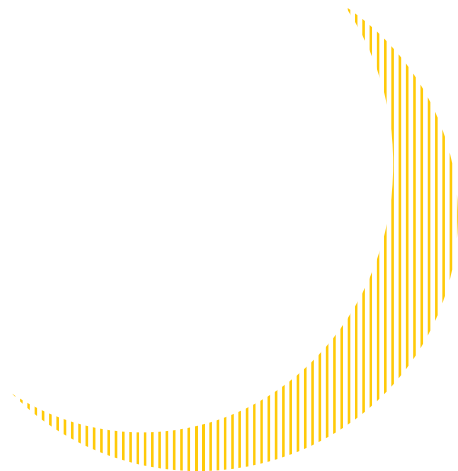
```
8  <body>
```

```
9    <h1>Título da Página</h1>
```

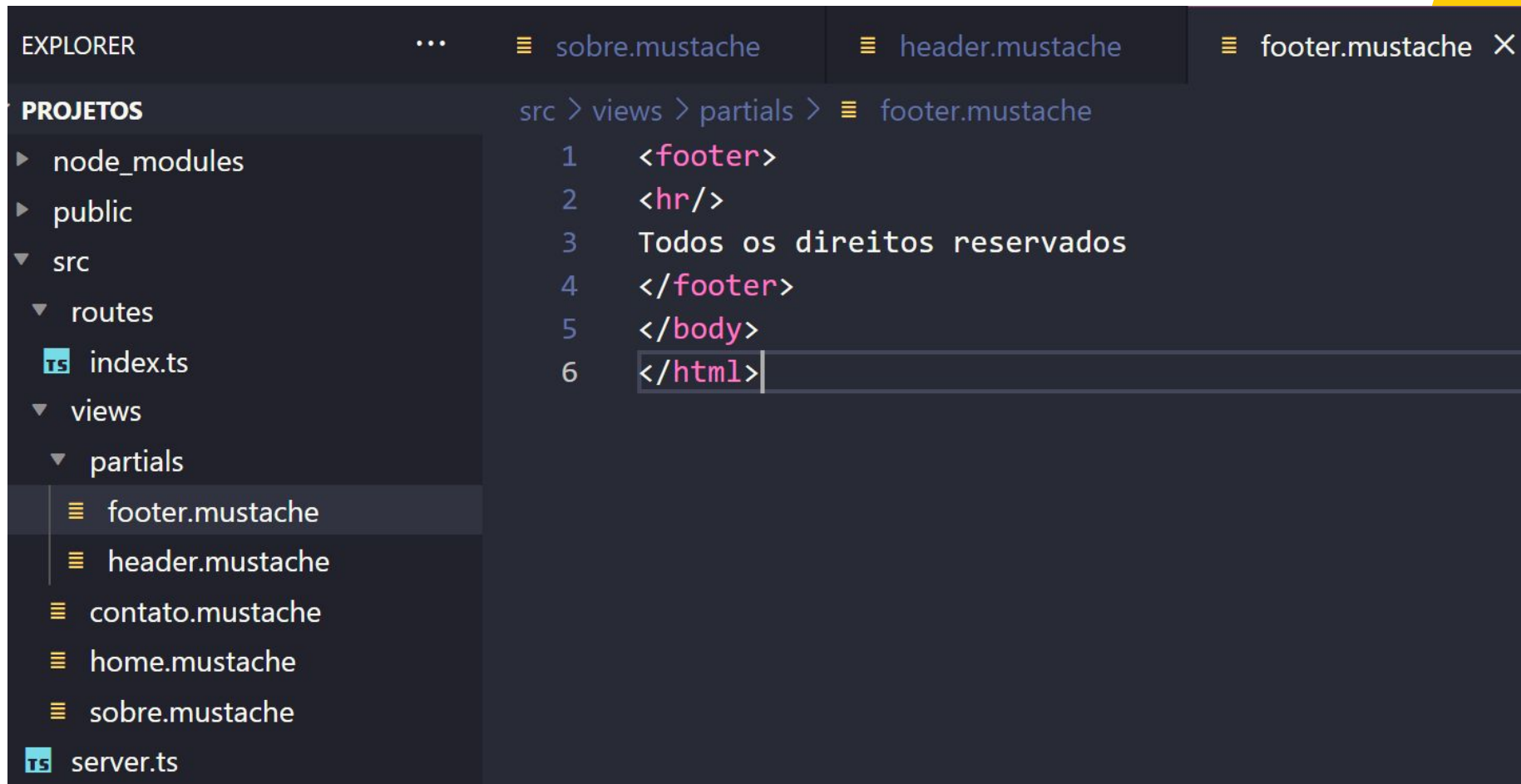
```
10   <hr/>
```



**CRIE UM PARTIAL PARA O RODAPÉ DA SUA
PÁGINA**



CRIANDO O FOOTER.MUSTACHE



The image shows a Visual Studio Code editor window with a dark theme. The Explorer sidebar on the left displays a project structure under 'PROJETOS'. The file 'footer.mustache' is selected in the 'src > views > partials' directory. The main editor area shows the content of 'footer.mustache' with line numbers 1 through 6. The breadcrumb at the top of the editor indicates the current file path: 'src > views > partials > footer.mustache'.

EXPLORER ... **sobre.mustache** **header.mustache** **footer.mustache** ✕

PROJETOS

- node_modules
- public
- src
 - routes
 - index.ts
 - views
 - partials
 - footer.mustache**
 - header.mustache
 - contato.mustache
 - home.mustache
 - sobre.mustache
 - server.ts

src > views > partials > **footer.mustache**

```
1 <footer>
2 <hr/>
3 Todos os direitos reservados
4 </footer>
5 </body>
6 </html>
```

COLOCANDO A PARTIALS NAS PAGINAS CONTATO,HOME E SOBRE



```
sobre.mustache X contato.mustache home.mustache f
src > views > sobre.mustache
1  {{>partials/header}}
2
3      <h1>Título da Página</h1>
4      <hr/>
5
6
7      <p> Olá, tudo bem?</p>
8
9  {{>partials/footer}}
10
```

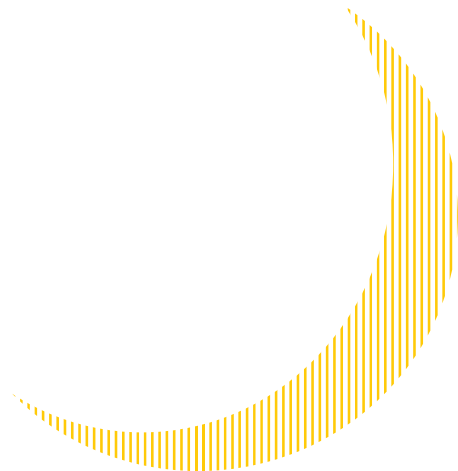


localhost/contato

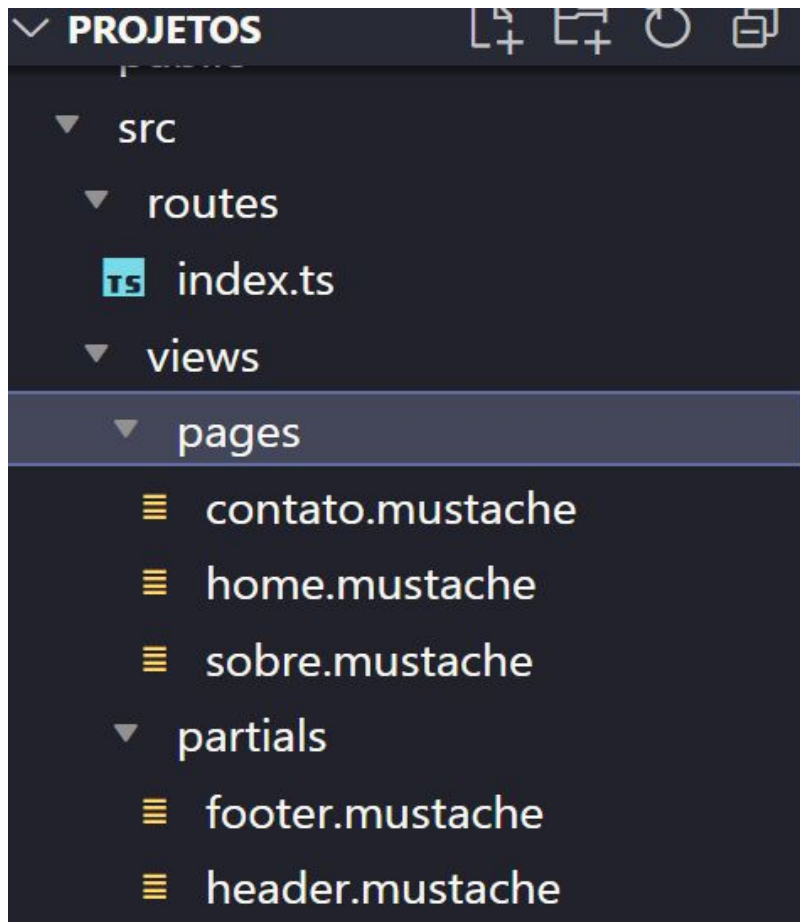
Título da Página

Olá, tudo bem?

Todos os direitos reservados



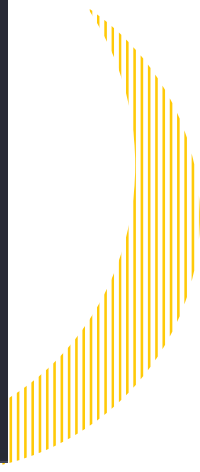
ORGANIZE SUAS PARTIALS NA PASTA PARTIALS E CRIE UMA NOVA PASTA CHAMADA PAGES E COLOQUE SUAS PÁGINAS



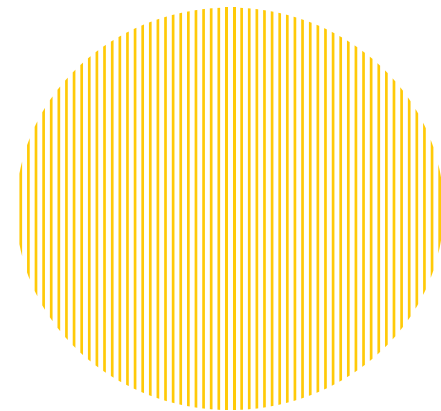
NÃO ESQUEÇA DE ARRUMAR AS ROTAS EM INDEX.TS



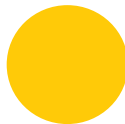
```
index.ts  X
src > routes > index.ts > ...
10     if(age > 18){
11         showOld = true
12     }
13
14     res.render('pages/home',{
15         name:'Viviane',
16         lastName: 'de Lima',
17         showOld,
18         products:[
19             {title:'Produto x', price:10 },
20             {title:'Produto y', price:30 },
21             {title:'Produto w', price:50 }
22         ],
23         frases:[]
24     })
25 })
26 router.get('/contato',(req:Request, res:Response) =>{
27     res.render('pages/contato')
28 })
29
30 router.get('/sobre',(req:Request, res:Response) =>{
31     res.render('pages/sobre')
32 })
33 export default router
```



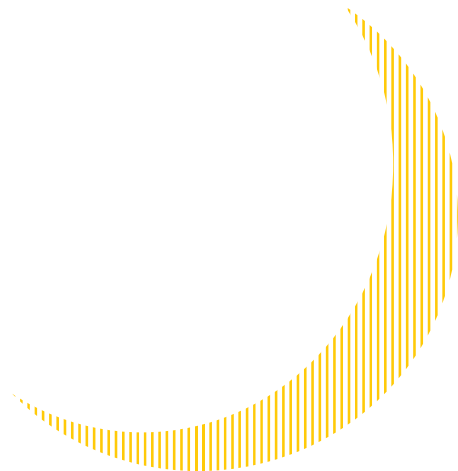
RECEBENDO DADOS DA URL



RECEBENDO DADOS PELA URL



Agora vamos aprender a receber dados da URL através de um formulário, o usuário vai digitar uma informação e eu vou receber esse dado. Para isso, crie uma nova rota (pode ser uma rota chamada **/nome**) e em seguida crie a página **nome.mustache**, renderize a rota e veja se aparece no seu servidor.



CRIANDO O PÁGINA NOME.MUSTACHE E A ROTA NOME



```
18         products:[
19             {title:'Produto x', price:10 },
20             {title:'Produto y', price:30 },
21             {title:'Produto w', price:50 }
22         ],
23         frases:[]
24     })
25 })
26 router.get('/contato',(req:Request, res:Response) =>{
27     res.render('pages/contato')
28 })
29
30 router.get('/sobre',(req:Request, res:Response) =>{
31     res.render('pages/sobre')
32 })
33
34 router.get('/nome',(req:Request, res:Response) =>{
35     res.render('pages/nome')
36 })
37 export default router
```

src

- routes
- index.ts
- views
 - pages
 - contato.mustache
 - home.mustache
 - nome.mustache
 - sobre.mustache
 - partials
 - footer.mustache
 - header.mustache
- server.ts
- package-lock.json
- package.json
- tsconfig.json

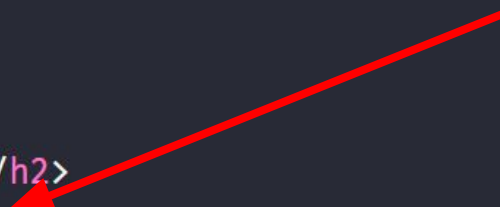
IMPORTE AS PARTIALS HEADER E FOOTER E CRIE UM FORMULÁRIO

nome.mustache X

src > views > pages > nome.mustache

```
1  {{>partials/header}}
2
3
4  <h2>Qual seu nome?</h2>
5  <form method="GET">
6      <input type="text" name="nome" placeholder="Digite o seu nome"/><br/>
7      <input type="submit" value="Enviar" />
8  </form>
9
10 {{>partials/footer}}
```

Observe que os dados estão sendo enviados através de GET



DIGITE UM NOME, CLIQUE EM ENVIAR E VEJA O NOME DIGITADO NA URL

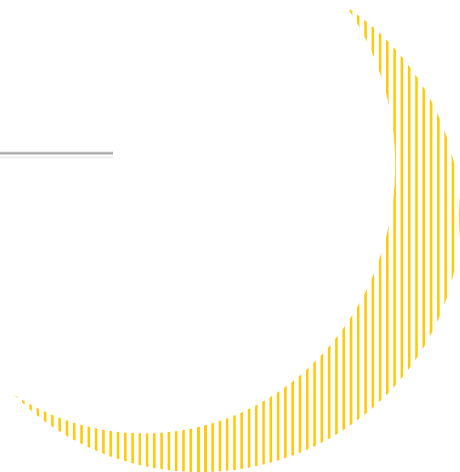


← → ↻ ⓘ localhost/nome?nome=Viviane

Título da Página

Qual seu nome?

Todos os direitos reservados



VAMOS FAZER O NOME DIGITADO APARECER NA PÁGINA, VÁ EM INDEX.TS



```
nome.mustache  index.ts  X
src > routes > index.ts > ...
29
30 router.get('/sobre',(req:Request, res:Response) =>{
31     res.render('pages/sobre')
32 })
33
34 router.get('/nome',(req:Request, res:Response) =>{
35
36     /* lembra do nosso req? (requisição)
37     chegou a hora de usar o Request, pois iremos
38     fazer uma requisição para o usuário */
39     let nome: string = req.query.nome as string
40     /* 'as string' para ajudar o typescript a especificar o
41     tipo desse dado que vou receber */
42
43     res.render('pages/nome',{
44         nome
45     })
46 })
47 export default router
```



VÁ EM NOME.MUSTACHE E COLOQUE A VARIÁVEL QUE SERÁ DIGITADA PELO USUÁRIO

```
nome.mustache x index.ts
src > views > pages > nome.mustache
1 {{>partials/header}}
2
3
4 <h2>Qual seu nome?</h2>
5 <form method="GET">
6   <input type="text" name="nome" placeholder="Digite o seu nome"/><br/>
7   <input type="submit" value="Enviar" />
8 </form>
9
10 <!-- usando a variável nome criada em index1.ts -->
11 {{#nome}}
12   Opa {{nome}}, tudo bem?
13 {{/nome}}
14
15
16 {{>partials/footer}}
```

Coloque o mesmo nome da variável que você criou

TESTE NO NAVEGADOR



Título da Página

Qual seu nome?

Opa Viviane, tudo bem?

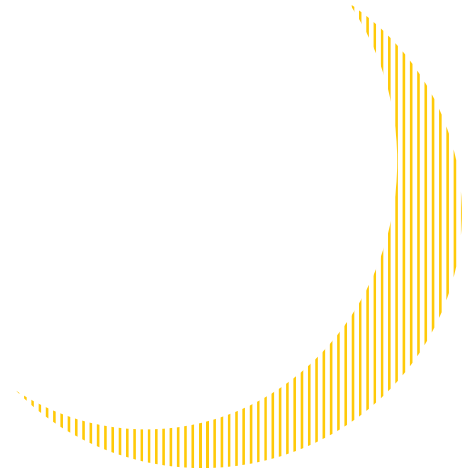
Todos os direitos reservados

Caso queira inserir um novo dado, delete o nome da url

VAMOS PRATICAR



Já aprendemos a receber dados via URL, sua missão é criar uma página nova que contém um formulário e solicite o nome, idade, telefone e endereço do usuário. Essas informações devem aparecer na página



VAMOS PRATICAR

Crie uma página nova que contém um formulário perguntando em que ano o usuário nasceu, em seguida mostrar quantos anos o usuário tem.

Não esqueça:

- Crie a página no Mustache
- Crie a rota dessa página
- A lógica do código deve estar na rota da página criada

Cabeçalho da Página

Qual ano você nasceu?

Todos os direitos reservados



RESOLUÇÃO : INDEX.TS

idade.mustache

index.ts



server.ts

src > routes > index.ts > ...

```
41  })
42  router.get('/idade',(req:Request, res:Response) =>{
43      //inicie a variável idade como zero para não dar erro
44      let mostrarIdade: boolean = false
45      let idade: number = 0
46      /*aqui vamos usar um if então SE o usuário enviou o ano,
47      vamos receber esse valor*/
48      if(req.query.anoNascimento){
49          //vamos pegar o ano de nascimento
50          let anoNascimento: number = parseInt(req.query.anoNascimento as string)
51          //em seguida vamos pegar o ano atual
52          let anoAtual: number = new Date().getFullYear()
53          //vamos fazer a conta para descobrir a idade
54          idade = anoAtual - anoNascimento
55          mostrarIdade = true
56      }
57      //recebendo a idade no render
58      res.render('pages/idade',{
59          idade,
60          mostrarIdade
61      })
62  })
63  export default router
```

RESOLUÇÃO: IDADE.MUSTACHE

TS index.ts × idade.mustache ×

src > views > pages > idade.mustache

```
1  {{>partials/header}}
2
3  <h2>Quantos anos você tem ?</h2>
4
5  <form method="GET">
6    <input type="text" name="anoNascimento" placeholder="digite seu ano de nascimento"/>
7    <input type="submit" value="Enviar"/>
8  </form>
9
10
11  {{#mostrarIdade}}
12    Você tem {{idade}} anos
13  {{/mostrarIdade}}
14
15
16  {{>partials/footer}}
17
```

VEJA NO NAVEGADOR



localhost/idade?anoNascimento=1995

Título da Página

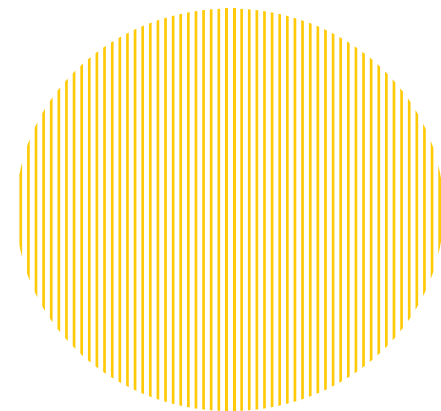
Qual ano você nasceu?

você tem 27 anos

Todos os direitos reservados



RECEBENDO DADOS VIA POST



VAMOS REUTILIZAR ESSA TELA



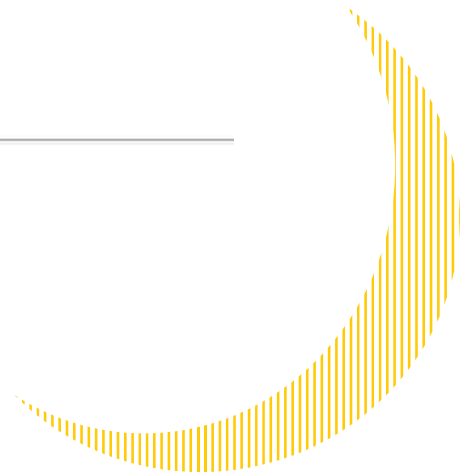
localhost/idade?anoNascimento=1995

Título da Página

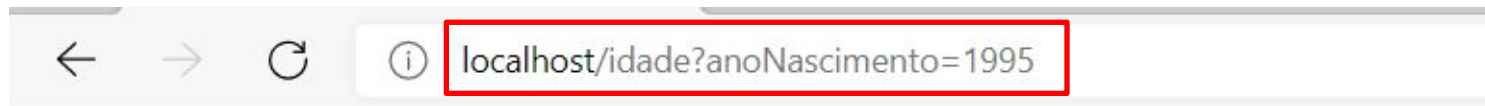
Qual ano você nasceu?

you have 27 years

All rights reserved



**NÃO VAMOS MAIS ENVIAR A INFORMAÇÃO DO ANO ATRAVÉS DA URL,
VAMOS ENVIAR INTERNAMENTE**



Título da Página

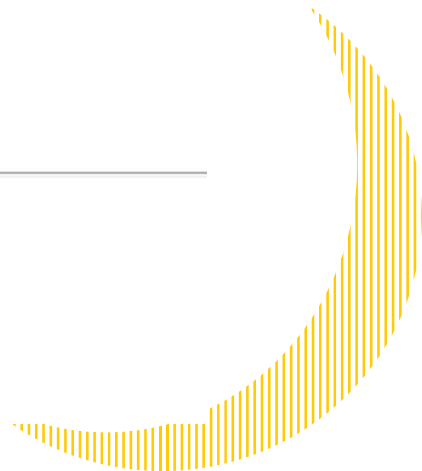
Qual ano você nasceu?

Digite sua idade

Enviar

you tem 27 anos

Todos os direitos reservados



VAMOS ENCERRAR O TERMINAL PRIMEIRAMENTE, APERTE CTRL C E EM SEGUIDA S



```
PS C:\Users\Viviane\Downloads\node\projetos> npm run start-dev
```

```
> projetos@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`
```

```
Deseja finalizar o arquivo em lotes (S/N)? S
```

VÁ EM SERVER.TS E USE URLENCODED

```
idade.mustache X server.ts X index.ts
src > server.ts > ...
1  import express,{Request, Response} from 'express'
2  import mustache from 'mustache-express'
3  import path from 'path'
4  import mainRoutes from './routes/index'
5
6  const server = express()
7
8  server.set('view engine','mustache')
9  server.set('views',path.join(__dirname,'views'))
10 server.engine('mustache',mustache())
11
12 server.use(express.static(path.join(__dirname,'../public')))
13
14 /*esse comando está me habilitando para
15 que eu pegue dados via POST */
16 server.use(express.urlencoded({extended:true}))
17
18 server.use(mainRoutes)
19
20 server.use((req: Request, res:Response) =>{
21
22     res.status(404).send('Página não encontrada!')
23 })
```

True significa que ele vai me dar os dados detalhadamente, ou seja, nome da variável, valor etc

VÁ EM IDADE.MUSTACHE E TROQUE O MÉTODO DO FORMULÁRIO PARA POST

```
idade.mustache X server.ts index.ts
src > views > pages > idade.mustache
1  {{>partials/header}}
2
3
4  <h2>Qual ano você nasceu?</h2>
5
6  <form method="POST">
7      <input type="text" name="anoNascimento" placeholder="Digite sua idade"/><br/>
8      <input type="submit" value="Enviar" />
9  </form>
10
11  você tem {{idade}} anos
12
13  {{>partials/footer}}
```

EXECUTE O TERMINAL NOVAMENTE



Experimente a nova plataforma cruzada PowerShell <https://aka.ms/pscore6>

```
PS C:\Users\Viviane\Downloads\node\projetos> npm run start-dev
```

```
> projetos@1.0.0 start-dev
```

```
> nodemon -e ts,json,mustache src/server.ts
```

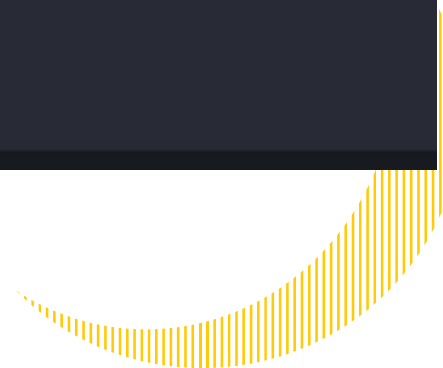
```
[nodemon] 2.0.15
```

```
[nodemon] to restart at any time, enter `rs`
```

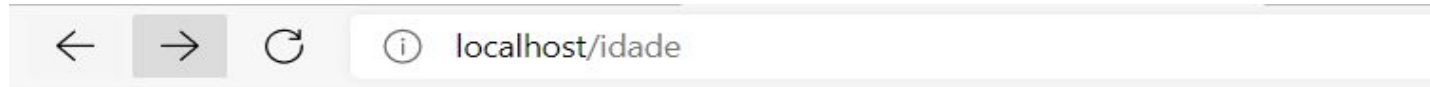
```
[nodemon] watching path(s): *.*
```

```
[nodemon] watching extensions: ts,json,mustache
```

```
[nodemon] starting `ts-node src/server.ts`
```



TESTE NO NAVEGADOR E VEJA O QUE ACONTECE

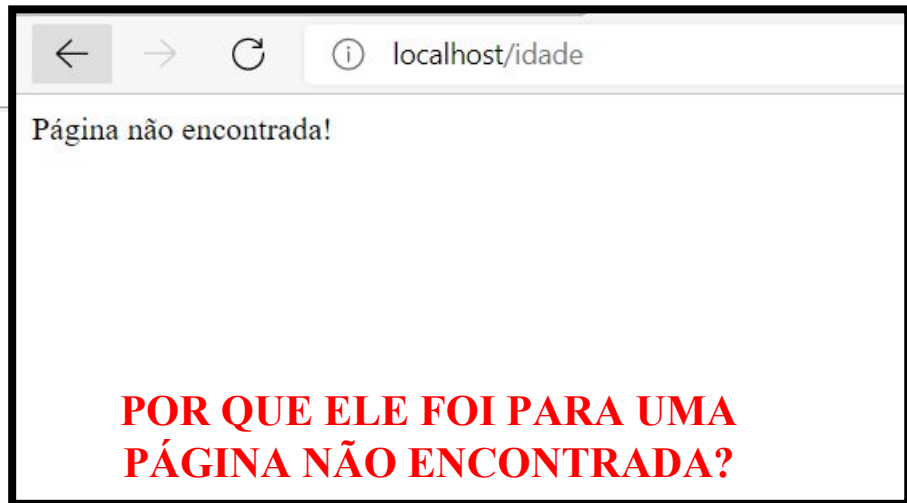


Título da Página

Qual ano você nasceu?

você tem 0 anos

Todos os direitos reservados



AGORA VAMOS FAZER A ROTA DO POST DENTRO DA ROTA POST COLOQUE O CÓDIGO QUE VOCÊ FEZ NO GET



```
src > routes > index.ts > ...  
38     res.render('pages/nome',{  
39       nome  
40     })  
41   })  
42  
43   router.get('/idade',(req:Request, res:Response) =>{  
44     res.render('pages/idade')  
45   })  
46  
47   router.post('/idade',(req:Request, res:Response) =>{  
48     let mostrarIdade: boolean = false  
49     let idade: number = 0  
50     if(req.query.anoNascimento){  
51       let anoNascimento: number = parseInt(req.query.anoNascimento as string)  
52       let anoAtual: number = new Date().getFullYear()  
53       idade = anoAtual - anoNascimento  
54       mostrarIdade = true  
55     }  
56     res.render('pages/idade',{  
57       idade,  
58       mostrarIdade  
59     })  
60   })  
61   export default router
```

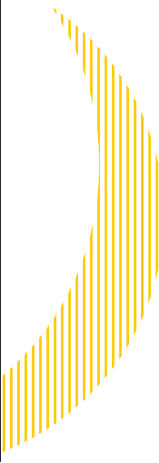
A partir de agora o GET ficará responsável apenas por exibir a página

O POST ficará responsável por processar as informações

ATENÇÃO: O DADO NÃO ESTÁ MAIS EM **REQ.QUERY** POR EU NÃO ENVIEI O DADO POR STRING, AGORA QUE ESTAMOS USANDO O POST NOS ENVIAMOS O DADO NO CORPO DA REQUISIÇÃO, ENTÃO AGORA NOSSO DADO ESTÁ EM **REQ.BODY**



```
idade.mustache  index.ts  X
src > routes > index.ts > ...
38     res.render('pages/nome',{
39         nome
40     })
41 })
42
43 router.get('/idade',(req:Request, res:Response) =>{
44     res.render('pages/idade')
45 })
46
47 router.post('/idade',(req:Request, res:Response) =>{
48     let mostrarIdade: boolean = false
49     let idade: number = 0
50     if(req.body.anoNascimento){
51         let anoNascimento: number = parseInt(req.body.anoNascimento as string)
52         let anoAtual: number = new Date().getFullYear()
53         idade = anoAtual - anoNascimento
54         mostrarIdade = true
55     }
56     res.render('pages/idade',{
57         idade,
58         mostrarIdade
59     })
60 })
61 export default router
```



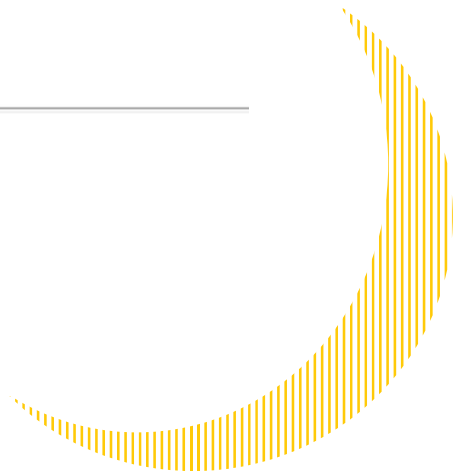


Título da Página

Qual ano você nasceu?

você tem 27 anos

Todos os direitos reservados



SE VOCÊ QUISER PODEMOS ENVIAR A INFORMAÇÃO DA IDADE PARA OUTRA PÁGINA

TS index.ts

idade.mustache X

src > views > pages > idade.mustache

```
1  {{>partials/header}}
2
3
4  <h2>Qual ano você nasceu?</h2>
5
6  <form method="POST" action="idade-resultado">
7    <input type="text" name="anoNascimento" placeholder="Digite sua idade"/><br/>
8    <input type="submit" value="Enviar" />
9  </form>
10
11  {{#mostrarIdade}}
12    você tem {{idade}} anos
13  {{/mostrarIdade}}
14
15  {{>partials/footer}}
```

TROQUE A ROTA DO POST PARA /IDADE-RESULTADO

```
TS index.ts  X
src > routes > TS index.ts > ...
17  HOME
18  })
19  })
20
21  router.get('/idade', (req: Request, res: Response) => {
22    res.render('pages/idade')
23  })
24
25  router.post('/idade-resultado', (req: Request, res: Response) => {
26    let mostrarIdade: boolean = false
27    let idade: number = 0
28    if (req.body.anoNascimento) {
29      let anoNascimento: number = parseInt(req.body.anoNascimento as string)
30      let anoAtual: number = new Date().getFullYear()
31      idade = anoAtual - anoNascimento
32      mostrarIdade = true
33    }
34    res.render('pages/idade', {
35      idade,
36      mostrarIdade
37    })
38  })
39  export default router
```



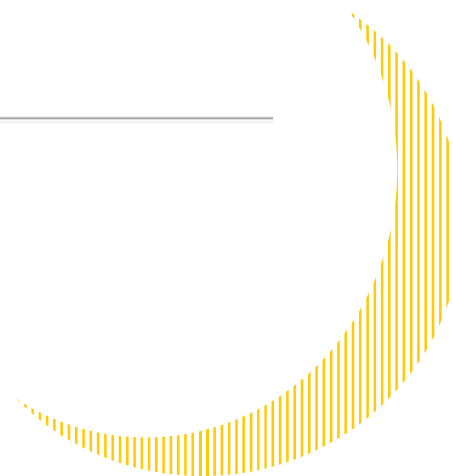
localhost/idade-resultado

Título da Página

Qual ano você nasceu?

você tem 27 anos

Todos os direitos reservados



EXERCÍCIO



Crie uma página de login que receba os dados de e-mail e senha via POST

DICA:

Criar a rota get só pra visualizar a página

Criar a post para os dados

