

1 Arquitetura REST

terça-feira, 27 de agosto de 2024 22:06

The screenshot shows a presentation slide with a dark header bar. On the left of the header is a back arrow icon. Next to it, the text 'Estilo arquitetural REST' is displayed in white, bold font. Below this, in a smaller white font, is the subtitle 'Arquitetura de Aplicações Rest com Spring Web'. The main content area of the slide is black.



Arquitetura REST

1. Transferência de Estado Representacional
(Representational State Transfer.)

2. Comunicação entre aplicações

3. Flexibilidade

Arquitetura REST implementa recursos CRUD
REST é arquitetura

RESTFULL é o sistema que utiliza REST
É a utilização da arquitetura

1,1 SPRINGBOOT Endpoint

terça-feira, 27 de agosto de 2024 22:19

Endpoint

URL disponivel para conseguirmos exibir nossa API

API são um conjunto de Endpoints

Podem ser consideradas nossos metodos dentro de Controller

```
ProductController.java

import java.util.List;

@RestController no usages + AndreFeh
@RequestMapping(value = "/produto")
public class ProdutoController {
    @Autowired 4 usages
    private ProdutoService service;

    @GetMapping("/{id}") 1 usage + AndreFeh
    public ResponseEntity<Produto> getOneItem(Long id){
        service.findById(id);
        return ResponseEntity.ok().body(getOneItem(id).getBody());
    }

    @GetMapping("/all") no usages + AndreFeh
    public ResponseEntity<List<Produto>> getAllItem(){
        List<Produto> produtos = service.findAll();
        return ResponseEntity.ok().body(produtos);
    }

    @PostMapping(value = "/save") no usages + AndreFeh
    public ResponseEntity<Produto> saveItem(Produto produto){
        /*Respose para retornar uma resposta HTTP*/
        produto = service.save(produto);
    }
}
```

1,2 SPRINGBOOT Tratamento de Excessoes Exceptions Handlers

sexta-feira, 30 de agosto de 2024 20:57

Quando não queremos que o sistema faça uma determinada ação

Utilizamos do tratamento de excessoes

Caso o usuario não coloque todos os dados que precisam,
Ou coloque um dado incorreto...

Como se fosse um TryCatch

@ExceptionHandler

Aponta que um valor não pode ter valores Null

As tratativas são aconselhadas a serem salvas no Service

Seu tratamento deve ser feito em uma classe propria,
utilizando de uma annotation @CONTROLLERADVISE

EM SERVICE

```
public Produto save(Produto produto) { 2 usages AndreFeh *
    if(produto.getName() == null){
        throw new ProdutoNullException();
    }
    return repository.save(produto);
}

public Produto findById(Long id) {
    return repository.findById(id);
}
```

The screenshot shows an IDE interface with code completion suggestions for the 'throw' statement. The suggestions include:

- Create class 'ProdutoNullException'
- Create record 'ProdutoNullException'
- Create inner class 'ProdutoNullException'
- Create inner record 'ProdutoNullException'
- Create DTO...
- Add Maven dependency...

At the bottom right of the completion list, there is a link: "Remove braces from 'if' statement".

Criando a classe para implementação

```
@ControllerAdvice no usages
/**Tornando uma classe de Avisos*/ public class ProdutoControllerAdvice extends ResponseEntityExceptionHandler { /*Controller de erros*/
// ESSA CLASSE SERVE PARA PEGAR OS ERROS QUE DERM DENTRO DA CLASS SERVICE
// Trata esses erros, e devolve a classe Controller original

//TODO: Ao inves de passar uma URL igual Controller,
// Ele ira passar um caminho caso de algum erro

@ExceptionHandler(ProdutoNullException.class) /*Quando acusar essa falha, entra nessa regra*/ public ResponseEntity<Object> capturaErroNull() {
    Map<String, Object> body = new HashMap<String, Object>();
    body.put("message", "verificar campos obrigatorios");

    return ResponseEntity.status(HttpStatus.BAD_REQUEST).body(body);
}
}
```

Dessa forma, sempre que chamar, ele vai dar um erro e não deixará que salve

```

public Produto save(Produto produto) throws Exception { 2 usages ✎ AndreFeh *
    if((produto.getName() == null) /*|| (produto.getPreco() == null)*){
        /*Criando uma exceção que trate valores Null*/
        throw new ProdutoNullException();
    }
    if (produto.getPreco() < 0) {
        throw new ProdutoPrecoException();
    }
    return repository.save(produto);
}

```

```

@ControllerAdvice no usages
/**Tornando uma classe de Avisos*/
public class ProdutoControllerAdvice extends ResponseEntityExceptionHandler { /*Controller de erros*/
// ESSA CLASSE SERVE PARA PEGAR OS ERROS QUE DEREM DENTRO DA CLASS SERVICE
// Trata esses erros, e devolve a classe Controller original

//TODO: Ao invés de passar uma URL igual Controller,
// Ele ira passar um caminho caso de algum erro

@ExceptionHandler(ProdutoNullException.class) /*Quando acusar essa falha, entra nessa regra*/ no usages
public ResponseEntity<Object> capturaErroNull() {

    Map<String, Object> body = new HashMap<String, Object>();
    body.put("message", "verificar campos obrigatorios");

    return ResponseEntity.status(HttpStatus.BAD_REQUEST).body(body);
}

@ExceptionHandler(ProdutoPrecoException.class) /*Quando acusar essa falha, entra nessa regra*/ no usages
public ResponseEntity<Object> capturaErroPrice() {

    Map<String, Object> body = new HashMap<String, Object>();
    body.put("message", "valor abaixo do permitido");

    return ResponseEntity.status(HttpStatus.BAD_REQUEST).body(body);
}

```

1,3 SPRINGBOOT Metricas de uma Aplicação

sábado, 31 de agosto de 2024 16:24

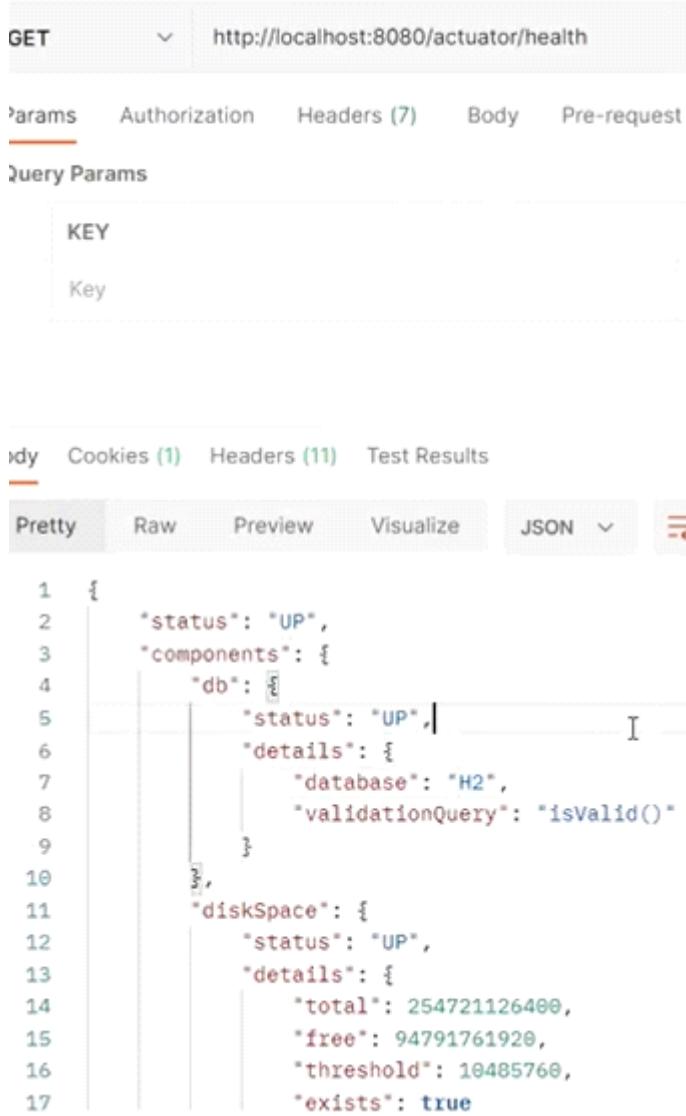
SPRING BOOT ACTUATOR

Verifica os recursos depois dela estar em ambiente de produção

Enquanto a apicação estiver rodando

Ele retornará as informações de disco, processamento, se o banco de dados subiu tal operação

Atraves de EndPoints, por meio de um JSON



```
1  {
2    "status": "UP",
3    "components": {
4      "db": {
5        "status": "UP",
6        "details": {
7          "database": "H2",
8          "validationQuery": "isValid()"
9        }
10       },
11      "diskSpace": {
12        "status": "UP",
13        "details": {
14          "total": 254721126400,
15          "free": 94791761920,
16          "threshold": 10485760,
17          "exists": true
18        }
19      }
20    }
21  }
```

Dessa forma fazemos de forma manual

No entanto tem uma forma de fazer isso de forma automatica

Spring Boot Actuator

É um recurso do Spring boot usado para monitorar, medir e exportar informações operacionais sobre a aplicação em execução.

SPRING BOOT ADMIN

Pesquisando por ele na Web, conseguimos coloca-lo no projeto através do Github

Spring Boot Admin

Metrics

http.server.requests	COUNT	TOTAL_TIME	MAX
(no tags)	313	10.7344323	1.8560513

jvm.buffer.count	VALUE
id:mapped - 'non-volatile memory'	0

```

Deployment Descriptor: REST-spring
JAX-WS Web Services
src/main/java
src/main/resources
src/test/java
JRE System Library [JavaSE-1.8]
Maven Dependencies
Deployed Resources
src
target
HELP.md
mvnw
mvnw.cmd
pom.xml
Spring-boot-admin
Deployment Descriptor: Spring-boot-admin
JAX-WS Web Services
src/main/java
br.com.dio.admin.Spring.boot.admin
ServletInitializer.java
SpringBootAdminApplication.java

```

```

import org.springframework.boot.SpringApplication;
@SpringBootAdminApplication
public class SpringBootAdminApplication {
    public static void main(String[] args) {
        SpringApplication.run(SpringBootAdminApplication.class, args);
    }
}

```

Table of Contents

- What is Spring Boot Admin?
- Getting started
 - Setting up Spring Boot Admin Server
 - Registering Client Applications
 - Spring Boot Admin Client
 - Spring Cloud Discovery
 - Registering Python Applications Using Pycuator
 - Client Applications
 - Show Version in Application List
 - JMX-Bean Management
 - Logfile Viewer
 - Show Tags per Instance
 - Spring Boot Admin Client
 - Spring Boot Admin Server
 - Running Behind a Front-end Proxy Server
 - Configuration Options
 - Spring Cloud Discovery
 - Static Configuration using SimpleDiscoveryClient
 - Other DiscoveryClients
 - Converting ServiceInstances
 - CloudFoundry
 - Clustering
 - Notifications
 - Mail Notifications
 - PagerDuty Notifications
 - OpsGenie Notifications
 - Hipchat Notifications
 - Slack Notifications
 - Let's Chat Notifications
 - Microsoft Teams

Support for Python applications is available using Pycuator.

2. Getting started

2.1. Setting up Spring Boot Admin Server

First, you need to setup your server. To do this just setup a simple boot project (using start.spring.io). As Spring Boot Admin Server is capable of running as servlet or webflux application, you need to decide on this and add the appropriate Spring Boot Starter. In this example we're using the servlet web starter.

- Add Spring Boot Admin Server starter to your dependencies:

```
pom.xml
```

```

<dependency>
    <groupId>de.codecentric</groupId>
    <artifactId>spring-boot-admin-starter-server</artifactId>
    <version>2.4.3</version>
</dependency>
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
</dependency>

```

If you want to use a snapshot version of Spring Boot Admin Server you most likely need to include the spring and sonatype snapshot repositories:

```
pom.xml
```

```

<repositories>
    <repository>
        <id>spring-milestone</id>
        <snapshots>
            <enabled>false</enabled>
        </snapshots>
    </repository>
</repositories>

```

1,4 - SPRINGBOOT Testes Unitarios e EndPoints

sábado, 31 de agosto de 2024 19:56

Teste automatizado, testa ações repetidas

Teste unitario, faz os testes automatizados, ao mesmo tempo, faz de todas as unidades

O pedido da erro se não for inserido algum campo?

Da erro se o valor for menor que zero?

TESTE JUNIT @TESTE

FRAMEWORK JUnit - "Jey Yunit"

Utilizando anotações @Test

É um código que testa códigos

TESTES @SPRINGBOOTTESTE

Quando numa aplicação SPRING

Somente teste unitario não é o suficiente pois precisamos testar chamadas complexas da API

Na lógica do desenvolvimento, sim é unitario, mas envolve toda a API

Por esse motivo que chamamos a annotation @SpringbootTest

Quando tentamos subir uma aplicação com um @Test normal, não da certo quando uma API Web Spring, pois ele testa depois sobe

Quando @Spring...Teste,

Ele primeiro lança a aplicação e depois de iniciada, retorna os testes, pois ele entende que primeiro irá lançar/iniciar a aplicação para depois estar "funcionando" as requisições

Para testarmos todas as requisições, os Save, os Controllers, Métodos, Saídas do repositório, retornos mensagens HTTP



MockMvc

1. MockMvcRequestBuilders

2. MockMvcResultMatchers

1- Controlo a requisição, o que estamos chamando de Body...

2- Para confirmarmos o que retornou, se bateu com o que estávamos esperando do retorno/resposta de tal método

Modo de implementação

@SPRINGBOOTTEST para antes de testar, iniciar a aplicação

@TEST para mostrar que é um teste do método

Quando fez uma injeção de dependência @Autowired MockMvc mockMvc

Ele já tem essa biblioteca dentro do Spring

Feito isso, entenda o outro @

@AUTOCONFIGUREMOCKMVC quando sem ele, se vc chama um SERVICE, ele só executa o que está no service, se um CONTROLLER, só um controller

Quando esse autoconfig esta injetado, ele pega tambem todas as classes necessarias para aquela requisição/método rodar

```
import org.springframework.http.MediaType;
import org.springframework.test.web.servlet.MockMvc;
import org.springframework.test.web.servlet.request.MockMvcRequestBuilders;
import org.springframework.test.web.servlet.result.MockMvcResultMatchers;

@SpringBootTest
@AutoConfigureMockMvc
public class AuthTest {

    @Autowired
    private MockMvc mockMvc;

    @Test
    public void deveRetornarErroComCredenciaisInvalidas() throws Exception {
        URI uri = new URI("/auth/signin");

        String content = "{ \"username\" : \"karantes\" , \"senha\" : \"123456\"}";

        mockMvc.perform(MockMvcRequestBuilders
                .post(uri)
                .content(content)
                .contentType(MediaType.APPLICATION_JSON))
                .andExpect(MockMvcResultMatchers
                        .status()
                        .is(200));
    }
}
```

```
@SpringBootTest
public class AuthTest {

    @Autowired
    private MockMvc mockMvc;

    @Test
    public void deveRetornarErroComCredenciaisInvalidas() throws URISyntaxException {
        URI uri = new URI("/auth/signin");

        String content = "{ \"username\" : \"karantes\" , \"senha\" : \"123456\"}";

        mockMvc.perform(MockMvcRequestBuilders
                .post(uri)
                .content(content)
                .contentType(MediaType.APPLICATION_JSON))
    }
}
```

10 - O QUE UTILIZAR PARA PROGRAMAR EM FRONT

domingo, 1 de setembro de 2024 21:29

HTML - Linguagem de Marcação

CSS - Linguagem de

ANGULAR - Framework JS

JAVASCRIPT - Linguagem de Interpretação

NODE JS - Interpretador (Runtime para Desktop)

JS - pode ser utilizado para Front mas tambem Back

NODE JS - Junto ao NODE é instalado junto no pacote o NPM, para instalação de pacotes a parte e rodar o node

10,1 Para que utilizar WebComponentes

terça-feira, 3 de setembro de 2024 21:22

Como entendemos de DIVs e HTML

Em um site da MEDIUM, vemos varios valores se repetindo,

Se fossem na mao, ficariam dessa forma



Agora pense fazer isso na mao, um por um e sempre que tiver uma manutenção, fazermos mais uma div, ou sempre atualizarmos manual

A screenshot of the Medium homepage displaying several articles. On the left, there are cards for "The Standards Innovation Paradox" by Frank Islam & Ed Osgo, "The Tale of Three Americas: Revolution. Evolution. Devolution." by Frank Islam & Ed Osgo, "Beyond Fake Toughness" by Brad Stulberg, and "I Asked an AI Robot to Generate 200-Year-Old Woodcuts of Mount Diablo" by Thomas Smith. On the right, the DOM structure for the "Beyond Fake Toughness" article is shown, highlighting the repetitive nature of the code for each article component like the header, image, and description.

Agora, se fizermos uma TAG chamada News

Essa tag seria como um "Publicar Anuncio",

Onde vc tem um escopo pronto, utilizando se de reaproveitamento de codigo e só seria necessário algumas minimas informações

11 - HTML

quarta-feira, 4 de setembro de 2024 17:49

Agora para um desafio, vamos fazer uma pagina com o Darth Vader

Colocando as marcações, estrutura com HTML

The screenshot shows a browser window with the title "PORTAL DE NOTICIAS GALACTICA". The address bar shows the URL "http://127.0.0.1:3000/index.html". The page content is titled "Darth Vader Contrata Devs" and includes the text "Darth Vader Procura Devs Que Sabem Trabalhar Com Componentes". Below the text is a blue rectangular image. To the left of the browser is a code editor window titled "index.html > html". The code is as follows:

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>PORTAL DE NOTICIAS GALACTICA</title>
7  </head>
8  <body>
9      <div class="container">
10         <div class="card">
11             <div class="card_left">
12                 <span>Andre Felipe</span>
13                 <h1>Darth Vader Contrata Devs</h1>
14                 <p>Darth Vader Procura Devs Que Sabem Trabalhar Com Componentes</p>
15             </div>
16             <div class="card_right">
17                 
18             </div>
19         </div>
20     </div>
21 </body>
22 </html>
```

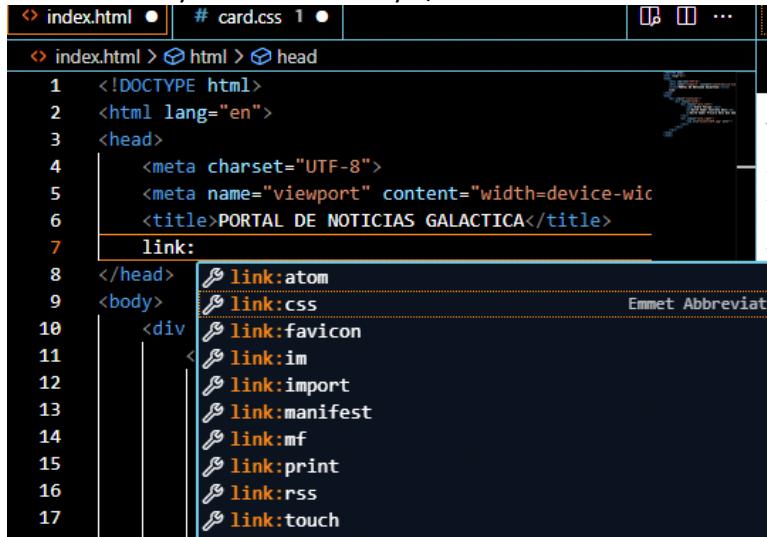
12 - CSS

quarta-feira, 4 de setembro de 2024 17:49

Para atrelar definições CSS nele

link:css

```
<link rel="stylesheet" href="style/card.css">
```



A screenshot of a code editor interface. At the top, there are tabs for 'index.html' and '# card.css'. The code editor shows the following HTML structure:

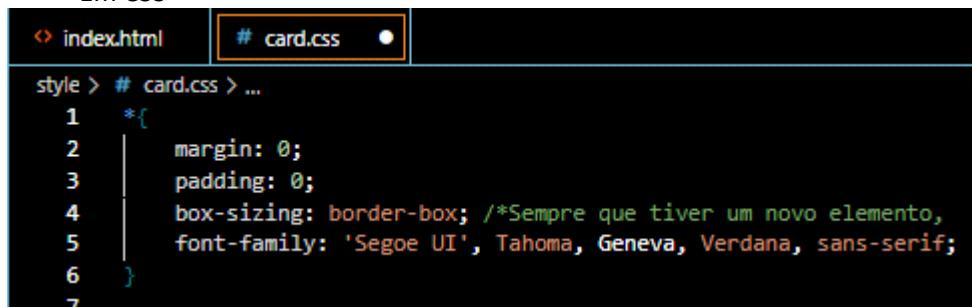
```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>PORTAL DE NOTICIAS GALACTICA</title>
7   link:
8 </head>
9 <body>
10  <div>
11    <link:atom>
12    <link:css> Emmet Abbreviations
13    <link:favicon>
14    <link:im>
15    <link:import>
16    <link:manifest>
17    <link:mf>
18    <link:print>
19    <link:rss>
20    <link:touch>
```

The cursor is positioned on the 'link:' tag at line 7. A dropdown menu titled 'Emmet Abbreviations' is open, listing various link types: atom, css, favicon, im, import, manifest, mf, print, rss, and touch.

Por default o navegador acaba adicionando margens, borda, tamanho

VAMOS RESETAR ISSO UTILIZANDO CSS

Em CSS



A screenshot of a code editor interface. At the top, there are tabs for 'index.html' and '# card.css'. The code editor shows the following CSS rule:

```
style > # card.css > ...
1 *{
2   margin: 0;
3   padding: 0;
4   box-sizing: border-box; /*Sempre que tiver um novo elemento,
5   font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
6 }
```

Como já visto em aulas de Flexbox, o flex container, que contém elementos flex, pode alinhar tudo conforme necessite, em blocos e direções

```
display: flex;
```

The screenshot shows a code editor with two tabs: 'index.html' and '# card.css'. The '# card.css' tab contains the following CSS:

```

1  *{
2   margin: 0;
3   padding: 0;
4   box-sizing: border-box; /*Sempre que tiver um novo elemento,
5   font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
6 }

7
8 .card{
9   width: 720px;
10  border: 1px solid #gray;
11  display: flex;
12  flex-direction: row; /*Separa as coisas em linhas, nao coluna
13 }

14
15 .card_left, .card_right{
16  border: 1px solid #blue;
17  display: flex;
18  flex-direction: column;
19  justify-content: center;
20 }

21
22 .card_left{
23  /*AQUI ADICIONAMOS O ADICIONAL SOMENTE DE card_left!*/
24  padding-left: 10px;
25
26 }

27
28 .card_left > span{
29  font-weight: bold;
30 }
31
32 .card_left > h1 {
33  margin-top: 10px;
34  font-size: 25px;
35 }
36
37 .card_left > p {
38  color: #rgb(70, 70, 70);
39 }

```

To the right, a browser window displays the rendered card. The card has a blue header bar at the top. The left section contains a bolded name and a large heading, while the right section features a large image of Darth Vader.

Podemos alterar ate o tamanho de uma imagem com CSS

The screenshot shows a code editor with the same '# card.css' tab. A new rule is added at the bottom:

```

img_darth_vader{
...height:200px;
...width:200px;
}

```

To the right, the browser preview shows the image of Darth Vader has been scaled down to fit within a 200x200 pixel area.

Para alguns efeitos especiais, temos algumas ferramentas a nosso dispor
CSS GENERATOR, atraves dele, temos como manipular a forma que caixas são exibidas
por exemplo

[CSS Generator Tool - CSS Demonstration and Generation](#)

13 DOM - Refatorando Objetos HTML/CSS

quarta-feira, 4 de setembro de 2024 20:12

ENTENDAAAAA!!!!

The screenshot shows a code editor with two files open: `index.html` and `card.css`, and a browser window displaying the result.

index.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>PORTAL DE NOTÍCIAS GALÁCTICA</title>
    <link rel="stylesheet" href="style/card.css">
</head>
<body>
    <div class="container">
        <div class="card">
            <div class="card_left">
                <span>Andre Felipe</span>
                <a href="#">Darth Vader Contrata Devs</a>
                <p>Darth Vader Procura Devs Que Sabem T...</p>
            </div>
            <div class="card_right">
                
            </div>
        </div>
    </div>
</body>
</html>
```

card.css:

```
style > # card.css > .card_left > a
7   .card{
8     width: 80%; /*720px valor real*//*100% porcento*/
9     display: flex;
10    flex-direction: row; /*Separa as coisas em linhas*/
11   justify-content: space-between;
12   box-shadow: 10px 8px 4px 6px rgba(0,0,0,0.28);
13   -webkit-box-shadow: 10px 8px 4px 6px rgba(0,0,0,0.28);
14   -moz-box-shadow: 10px 8px 4px 6px rgba(0,0,0,0.28);
15 }
16 }

17 .card_left, .card_right{
18   display: flex;
19   flex-direction: column;
20   justify-content: center;
21 }

22 }

23 }

24 .card_left{
25   /*AQUI ADICIONAMOS O ADICIONAL SOMENTE DE card_*/
26   padding-left: 10px;
27 }

28 }

29 }

30 .card_left > span{
31   font-weight: 400;
32 }

33 }

34 .card_left > a{
35   margin-top: 10px;
36   font-size: 25px;
37   text-decoration: none;
38   font-weight: bold;
39   color: black;
40 }

41 }

42 .card_left > p{
43   color: #rgb(70, 70, 70);
44 }

45 }

46 .card_right{
47   flex-direction: column;
48 }

49 }

50 .img_darth_vader{
51   height: 200px;
52   width: 200px;
53 }
```

The browser preview shows a card component with a left section containing a name and a link, and a right section featuring an image of Darth Vader.

Para fazer isso manual, foi preciso algum tempo gasto, varias linhas e se for duplicar esses elementos em mais cards, seria necessario muito codigo e muita poluição de codigos,

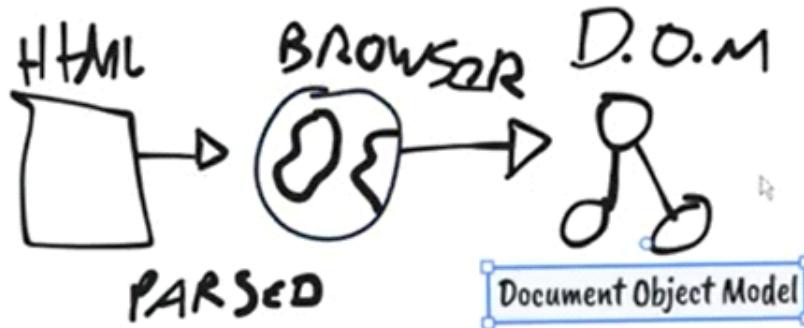
Em encapsulamento poderemos definir tudo isso em uma "CLASSE"

Para isso, precisamos entender,

Quando um Código HTML/CSS/JS... É passado para um navegador,

Essa passagem se chama PARSED

O navegador entende esse projeto em formato D.O.M



Onde na verdade ele entende como se fosse uma árvore,

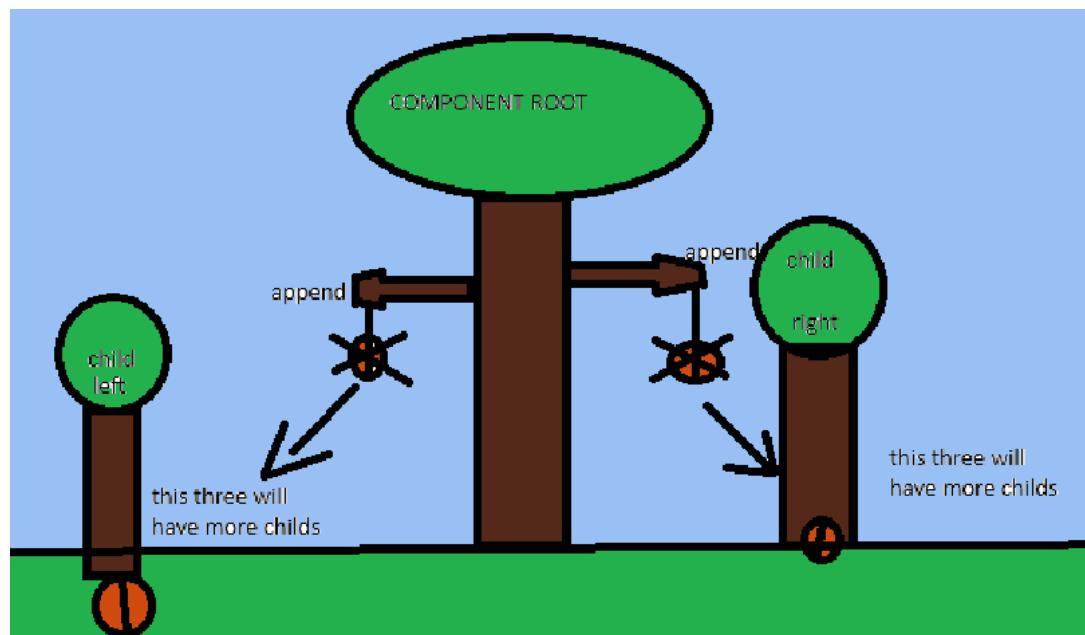
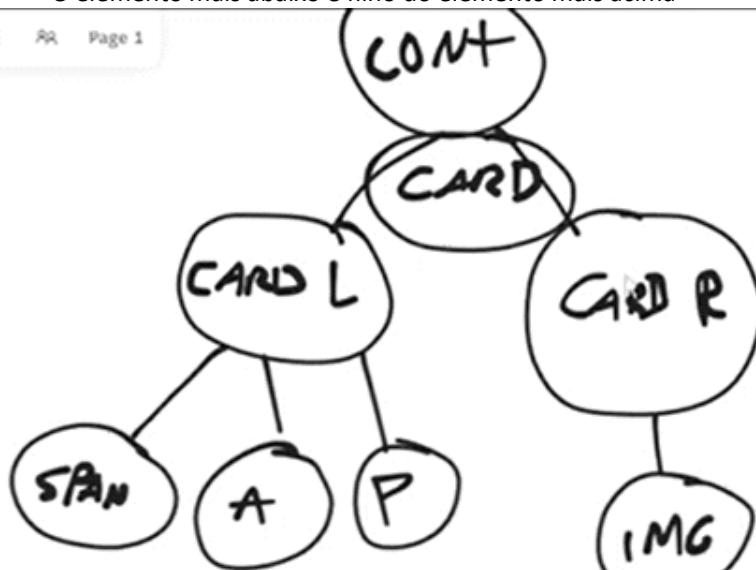
TEM O HTML

DEPOIS O CSS, COM CLASSES, CLASSES FILHAS E PRÓXIMAS CLASSES

Sempre gerando galhos ou raízes mais abaixo

O elemento mais abaixo é filho do elemento mais acima

≡ PR Page 1



10,3 EXTEENOES E VS CODE PARA FRONT

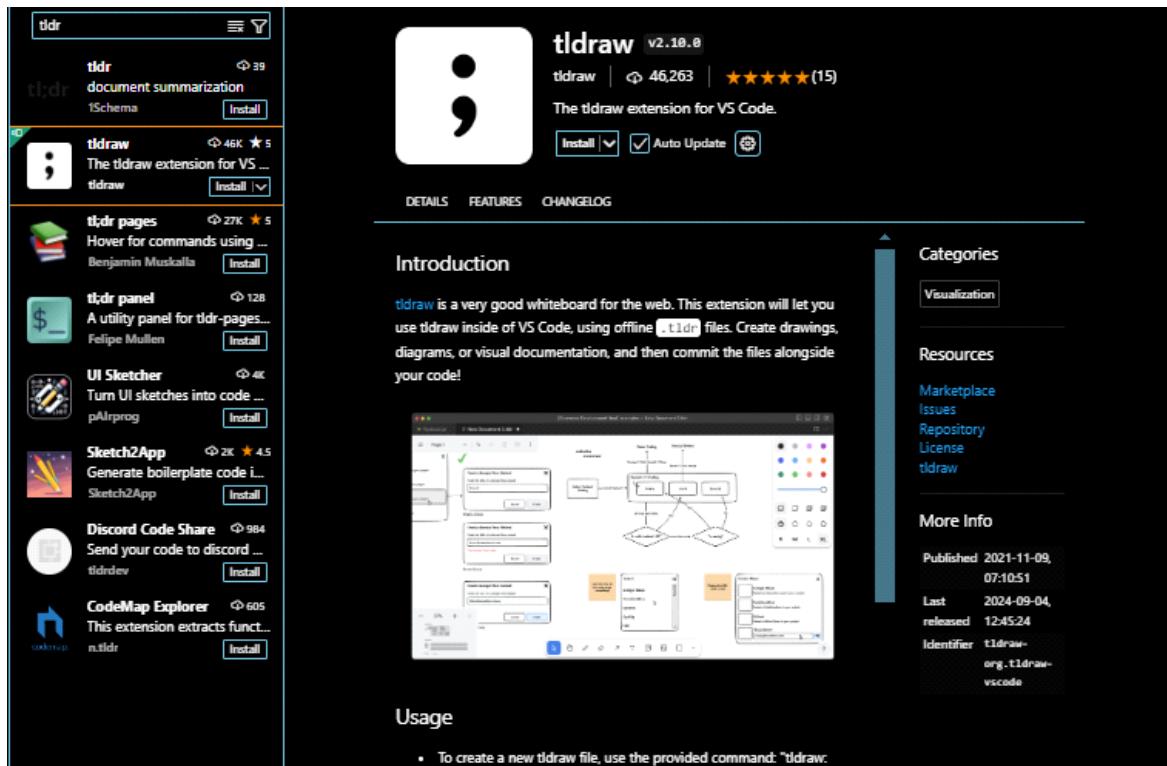
quarta-feira, 4 de setembro de 2024 20:16

EXTEENOES UTEIS

TLDRAW

Live Preview - (Atualiza em tempo real na IDE)

Live Serve - (Atualiza no Navegador na porta configurada)



10,4 Quebrando logica DOM com Shadow DOM

quarta-feira, 4 de setembro de 2024 20:43

Quando trabalhamos com Front

Todos os codigos são manipulados pelo navegador

No entanto,

Podemos burlar essa manipulação, quando dentro de uma DOM há uma DOM Oculta

Chamamos de DOM Fantasma

Ou Shadow DOM

Dessa forma podemos manipular o rendimento e performance da aplicação

ELE TEM COMPATIBILIDADE COM 90% DOS BROWSERS, EXCETO OPERA MINI (OPERA FUNCIONA) E INTERNET EXPLORER

13,5 Shadow DOM - _Component_ para Encapsular Elementos

quarta-feira, 4 de setembro de 2024 21:01

COMPONENTS são simplesmente Metodos/Funções JS

JS utiliza assim como Java, algumas extensoes

Ou seja, para tornar alguma classe como SHADOW DOM

13,6 JS CRIANDO COMPONENTS COM JS

quarta-feira, 4 de setembro de 2024 22:42

CRIANDO PROJETO JS

JAVASCRIPT TEM ALGUMAS SIMILARIDADES DO JAVA

COMO EXTENDS... CLASS... CONSTRUCTOR()... SUPER()... THIS.X;

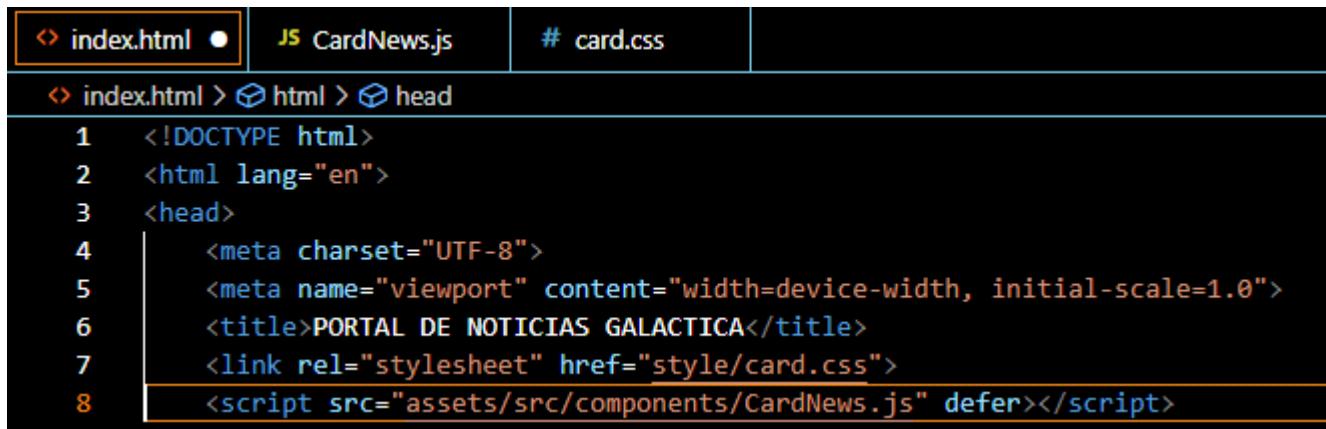
Isso ajuda a entender e aperfeiçoar o conhecimento já adquirido

```
class CardNews extends HTMLElement {  
    constructor(){  
        super();  
    }  
}
```

O **HTMLElement** serve para puxar elementos HTML para dentro do JS

Constructor pois tudo dentro de um JS primeiro inicia com Construtor

ANTES DE COMEÇAR, LEMBRE-SE DE ADICIONAR A REFERENCIA NO HTML



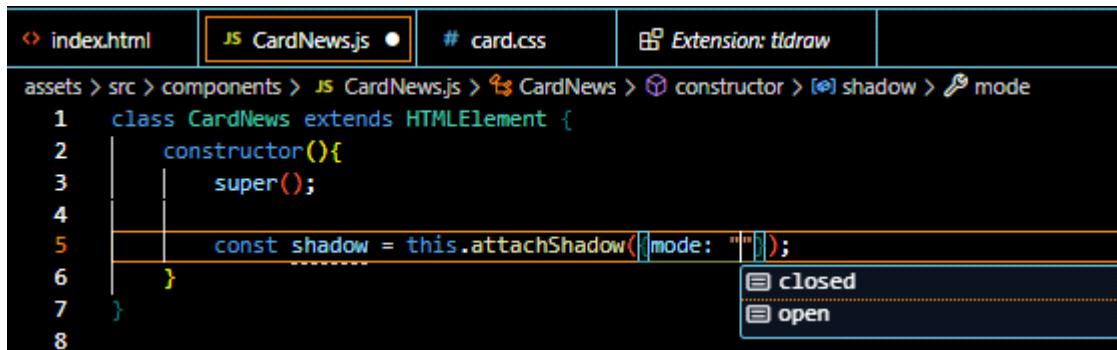
The screenshot shows a code editor with four tabs: index.html, CardNews.js, card.css, and Extension: tldraw. The index.html tab contains the following code:

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
    <meta charset="UTF-8">  
    <meta name="viewport" content="width=device-width, initial-scale=1.0">  
    <title>PORTAL DE NOTICIAS GALACTICA</title>  
    <link rel="stylesheet" href="style/card.css">  
    <script src="assets/src/components/CardNews.js" defer></script>
```

Esse DEFER depois do link, diz que primeiro crie toda sua estrutura, para entao chamar os metodos

Super, pois esta atribuindo funções da classe HTMLElement dentro desse construtor

OU seja, estamos implementando o Construtor da nossa classe, junto com o Construtor da classe HTML Element



The screenshot shows a code editor with four tabs: index.html, CardNews.js, card.css, and Extension: tldraw. The CardNews.js tab contains the following code:

```
class CardNews extends HTMLElement {  
    constructor(){  
        super();  
        const shadow = this.attachShadow({mode: ""});  
    }  
}
```

A tooltip for the 'mode' parameter in the attachShadow method shows two options: 'closed' and 'open'.

Esse comando atribui Shadow a classe

This pois é essa classe + chamada de elemento da classe extendida

Essa classe, vai criar uma sombra

ANEXAR UMA SOMBRA ABERTA

Mode closed or open

Pois closed, para dizer que nenhuma ação externa pode mudar a forma que esse elemento é executado

Open, para dizer que ações externas podem mudar a forma que esse elemento é executado, seja alguma tecla ou interação do usuário

Tecle Sim para...

Não para...

Ou até para outro JS influenciar-lo

```
shadow.innerHTML = "<h1> HELLO WORLD </h1>";
```

Esse método para imprimir alguma frase!

DENTRO DESSA SOMBRA, IMPRIMA UM TÍTULO HELLO WORLD

Elementos Customizados

```
customElements
  [e] customElement... var customElements: CustomElementRegi...
  [e] CustomElementRegistry

customElements.define('cardNews', CardNews);
```

PRONTO !!!

Criado um Shadow DOM

Através de uma classe

Através de um construtor chamado com a classe pai

Criando uma const chamada shadow, anexando o modo que ela opera

Criando um Elemento customizado definindo essa Classe CardNews como TAG

```
index.html JS CardNews.js # card.css
assets > src > components > JS CardNews.js > ...
1  class CardNews extends HTMLElement {
2    constructor(){
3      super();
4
5      const shadow = this.attachShadow({mode: "open"});
6      shadow.innerHTML = "<h1> HELLO WORLD </h1>";
7    }
8
9
10 customElements.define ("card-news", CardNews);
11
```

CONSEGUIMOS UTILIZAR COMO TAG HTML ESSE COMANDO

<card-news></card-news>

```
› index.html X JS CardNews.js # card.css
index.html > html > body > card-news
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>PORTAL DE NOTICIAS GALACTICA</title>
7      <link rel="stylesheet" href="style/card.css">
8      <script src="assets/src/components/CardNews.js" defer></script>
9  </head>
10 <body>
11 >     <div class="container">...
12     </div>
13     <card-news></card-news>
14 </body>
15 </html>
```

13,7 JS Entendendo sobre Componentes

quinta-feira, 5 de setembro de 2024 21:14

Component

COMPONENTE é um JavaScript que cria HTML, CSS e JS

Podemos entender como Métodos

Tem o HTML composto por CSS

Junto ao HTML podemos implementar components

Esses components podem conter tags html

```
<div>  
<p>  
<a>  
<...>
```

Component é um conjunto de elementos de forma encapsulada

Podendo ate ter CSS dentro de um component

Lembrando que cada tag tem sua diferenciação

ASSIM COMO IMAGEM

```
<div class="card__right">  
|     
</div>
```

Para definirmos um componente, primeiro estruturamos dessa forma

The screenshot shows a code editor with the following structure:

- File tabs: Welcome, index.html, card.css, TituloDinamico.js (highlighted).
- Project tree: src > Components > TituloDinamico.js.
- Code content (TitoloDinamico.js):

```
1 // TITULO DINAMICO (JS que retorna JS, HTML, CSS)  
2 class TituloDinamico extends HTMLElement{  
3     constructor(){  
4         super();  
5         /*CRIANDO SHADOW*/  
6         const shadow = this.attachShadow({mode: "open"});  
7     }  
8 }  
9 }
```

O QUE O COMPONENTE PRECISA TER

Base do Componente

```
//base do component <h1>Felipe</h1>  
const componentRoot = document.createElement("h1");  
componentRoot.textContent = 'Felipe'
```

Estilizar o Componente // Dar estilo ao componente

Enviar para a Shadow

- customElement.define("nome-algo", construtor);
 - o É esse o nome do metodo quando ele for chamado no HTML

Terminado o metodo, ele ficará dessa forma

IMPLEMENTADO NO JS

The screenshot shows a code editor with two tabs: 'index.html' and 'JS TituloDinamico.js'. The 'JS TituloDinamico.js' tab is active, displaying the following code:

```
1 // TITULO DINAMICO (JS que retorna JS, HTML, CSS)
2 class TituloDinamico extends HTMLElement{
3     constructor(){
4         super();
5
6         /*CRIANDO SHADOW*/
7         const shadowComp = this.attachShadow({mode: "open"});
8
9         // Precisamos ter a BASE DO COMPONENTE, ESTILIZA O COMPONENETE E ENVIA PARA A SHADOW
10        // Base do Componente
11        const componentRoot = document.createElement("h1");
12        componentRoot.textContent = 'André'; // Primeiro vamos fazer constante, mas futuramente dinamico
13        // Estilizar o Componente >>> Dar estilo ao componente
14        const style = document.createElement("style");
15        style.textContent =
16            `h1{
17                color: red;
18            }
19        `;
20
21        // Enviar para a Shadow
22        shadowComp.appendChild(componentRoot);
23        shadowComp.appendChild(style);
24    }
25 }
26
27 customElements.define("titulo-dinamico", TituloDinamico);
```

CHAMANDO ELE NO HTML

Lembre que no JS definiu a chamada dele como tag `titulo-dinamico`

The screenshot shows a code editor with two tabs: 'index.html' and 'JS TituloDinamico.js'. The 'index.html' tab is active, displaying the following code:

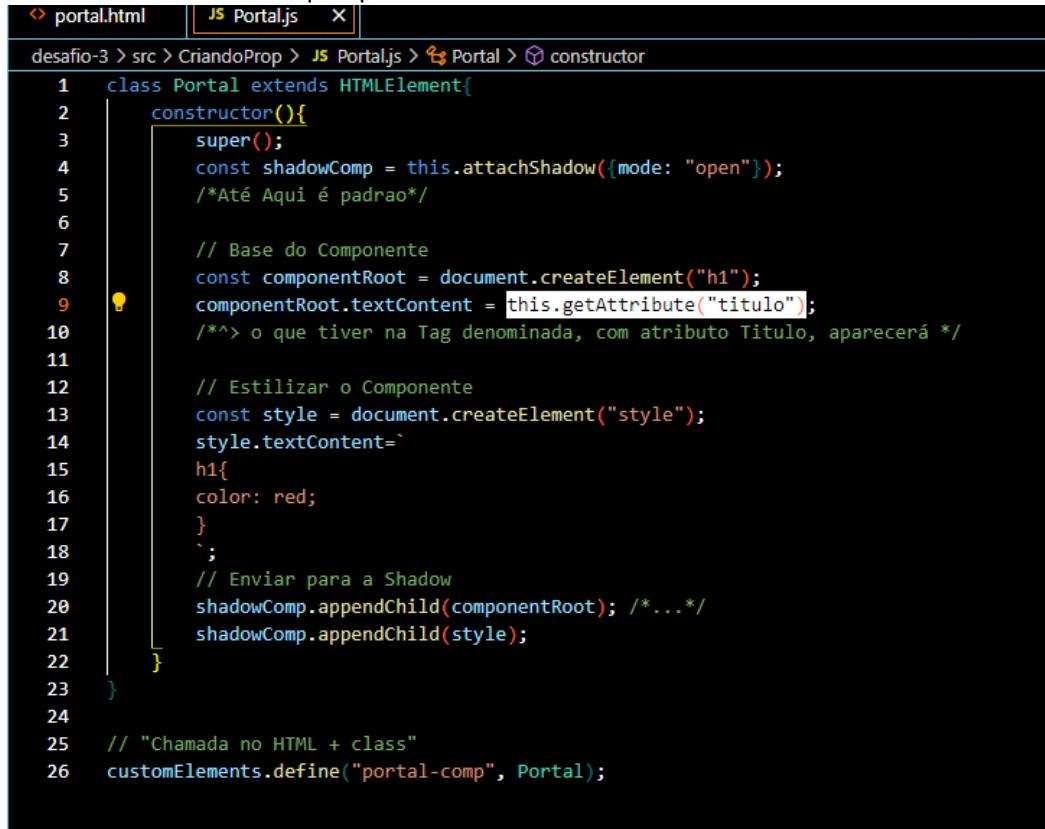
```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>Titulo Dinamico</title>
7     <link rel="stylesheet" href="styles/card.css">
8     <script src="src/Components/TituloDinamico.js"></script>
9 </head>
10 <body>
11     <div><h1>Hello World</h1></div>
12     <!--<titulo-dinamico></titulo-dinamico>-->
13 </body>
14 </html>
```

13,8 Criando PROPS

sábado, 14 de setembro de 2024 22:43

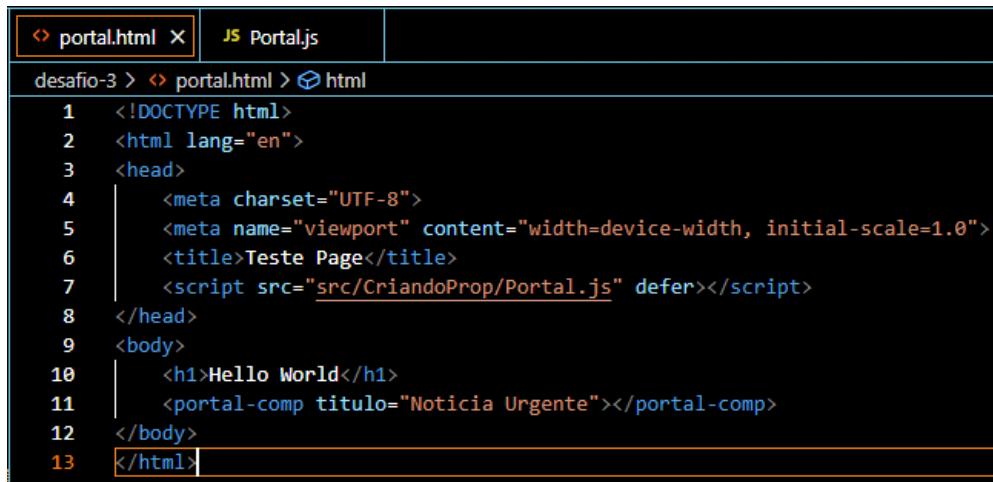
Uma prop é quando reaproveita um componente,
Deixando como se fosse um metodo e sua chamada é global, mudando somente parametros

No JS ficando dessa forma, com get attribute
e o atributo que quer definir



```
portal.html JS Portal.js X
desafio-3 > src > CriandoProp > JS Portal.js > Portal > constructor
1 class Portal extends HTMLElement{
2     constructor(){
3         super();
4         const shadowComp = this.attachShadow({mode: "open"});
5         /*Até Aqui é padrao/
6
7             // Base do Componente
8             const componentRoot = document.createElement("h1");
9             componentRoot.textContent = this.getAttribute("titulo");
10            /*^> o que tiver na Tag denominada, com atributo Titulo, aparecerá */
11
12            // Estilizar o Componente
13            const style = document.createElement("style");
14            style.textContent=`
15                h1{
16                    color: red;
17                }
18            `;
19            // Enviar para a Shadow
20            shadowComp.appendChild(componentRoot); /*...*/
21            shadowComp.appendChild(style);
22        }
23    }
24
25    // "Chamada no HTML + class"
26    customElements.define("portal-comp", Portal);
```

No HTML fica dessa forma



```
portal.html X JS Portal.js
desafio-3 > portal.html > html
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>Teste Page</title>
7     <script src="src/CriandoProp/Portal.js" defer></script>
8 </head>
9 <body>
10    <h1>Hello World</h1>
11    <portal-comp titulo="Noticia Urgente"></portal-comp>
12 </body>
13 </html>
```

ENTAO AGORA COM UMA NOVA CLASSE

13,10 Trabalhando com PROPS de Maneira Profissional

domingo, 15 de setembro de 2024 01:00

Para aplicarmos uma Shadow, de maneira que não fique uma bagunça e ser difícil a leitura

Necessario entender da seguinte forma

```
desafio-4 > src > JS CardNews.js > CardNews
1  v class CardNews extends HTMLElement{
2  v   constructor(){
3    |     super();
4    |     const shadow = this.attachShadow({mode: open});
5    |     /*Até Aqui é padrao para "COMP DENTRO DE COMP"*/
6    |
7
8    |     .....build(){}
9    |     .....styles(){}
10   }
11
12  customElements.define("card-news", CardNews);
```

build(){}

Para definir os escopos, metodos e atributos dentro da Prop

styles(){}

Para definir os estilos e layouts dentro da Prop

```
News.html • JS CardNews.js X JS Portal.js
-4 > src > JS CardNews.js > CardNews > build
class CardNews extends HTMLElement{
  constructor(){
    super();
    const shadow = this.attachShadow({mode: open});
    /*Até Aqui é padrao para "COMP DENTRO DE COMP"*/
    shadow.appendChild(this.build());
    shadow.appendChild(this.styles());
  }

  build()/*      Para definir os escopos, metodos e atributos dentro da Prop*/
  {
    const componentRoot = document.createElement("div"); /* pois Root é o component mais alto da hierarquia */
    /*2º*/componentRoot.setAttribute("class", "card");/*Setando um novo Atributo, para ligá-lo aos outros ramos */

    /*1º*/
    const cardLeft = document.createElement("div");
    |   cardLeft.setAttribute("class", "card_left"); /*definindo as classes*/
    const cardRight = document.createElement("div");
    cardRight.setAttribute("class", "card_right"); /*definindo as classes*/

    /*3º// Atrelando CardLeft e ...Right como filhas de Root*/
    componentRoot.appendChild(cardLeft);
    componentRoot.appendChild(cardRight);

    // como se trata de "metodo" deve ter um return, pois nao é VOID
    return componentRoot;
  }

  styles()/*      Para definir os estilos e layouts dentro da Prop*/
}

customElements.define("card-news", CardNews);
```

Entenda

Nesse código dentro de Build, como já descrito

Temos a classe pai que seria o Root ali descrito,
onde mais a frente ele teria um estilo, mas primeiro um atributo classe card
e seus "filhos" left e right
Que por sua vez também terão estilos por classe card...

Depois de finalizar esse método/prop/sombra por completo, deve dar um retorno para
que de certo na hora da chamada do método

14,1 Aplicando Elementos em Props

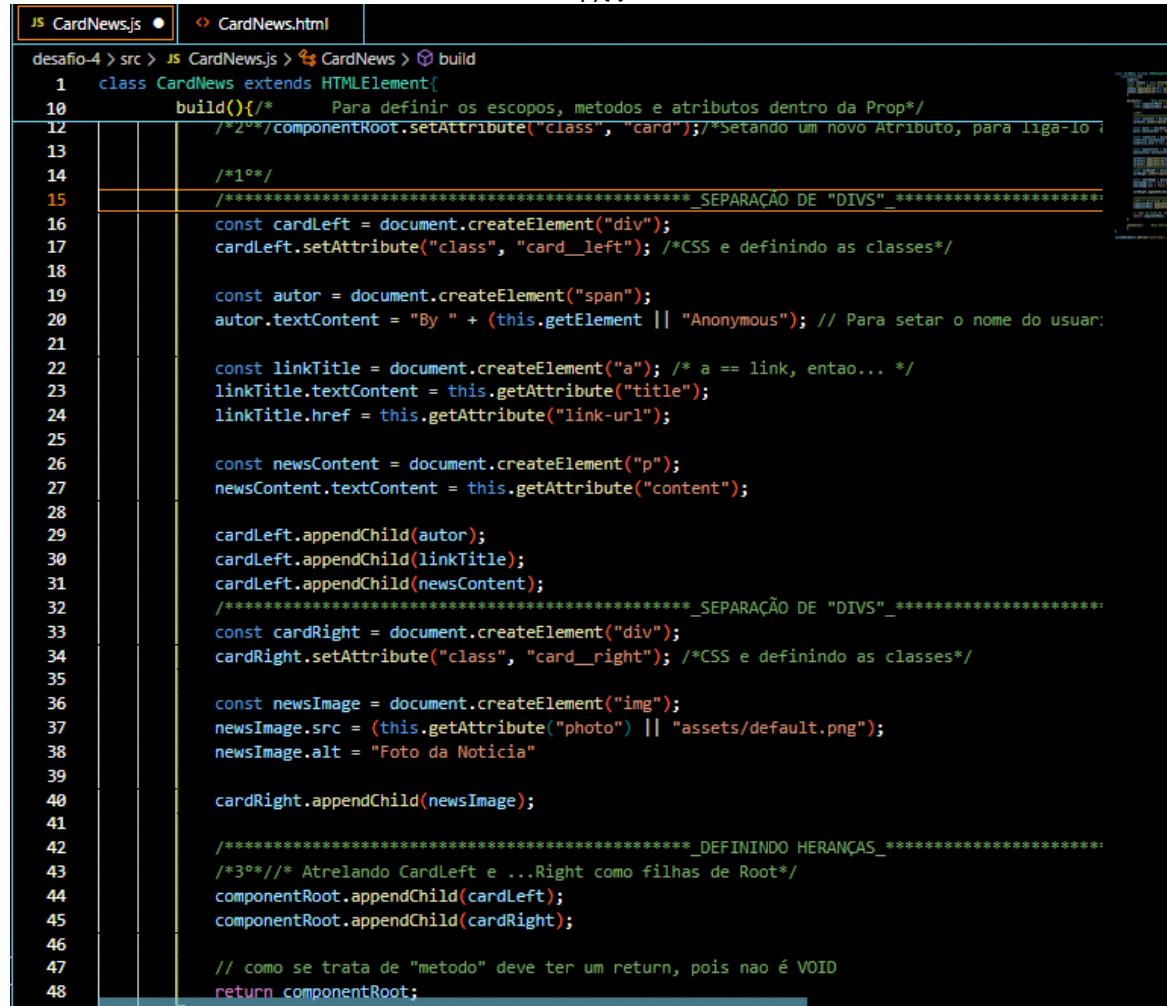
domingo, 15 de setembro de 2024 17:13

Primeiro precisamos revisar os documentos anteriores.

Agora entender que aquele HTML é tudo feito como se fosse um C++ onde tudo é estruturado

Com JS, podemos encapsular e criar um tipo de Herança onde podemos utilizar um conjunto de código denominado como prop e ainda aplicar estilo a eles

APLICANDO OS ELEMENTOS ATRAVES DE BUILD(){ }



```
JS CardNews.js   CardNews.html
desafio-4 > src > js CardNews.js > CardNews > build
1  class CardNews extends HTMLElement{
10    build(){/*      Para definir os escopos, métodos e atributos dentro da Prop*/
12      /*2**/componentRoot.setAttribute("class", "card");/*Setando um novo Atributo, para ligá-lo*/
13
14      /*1*/
15      /*****_SEPARAÇÃO DE "DIVS" *****/
16      const cardLeft = document.createElement("div");
17      cardLeft.setAttribute("class", "card_left"); /*CSS e definindo as classes*/
18
19      const autor = document.createElement("span");
20      autor.textContent = "By " + (this.getElement || "Anonymous"); // Para setar o nome do usuário
21
22      const linkTitle = document.createElement("a"); /* a == link, então... */
23      linkTitle.textContent = this.getAttribute("title");
24      linkTitle.href = this.getAttribute("link-url");
25
26      const newsContent = document.createElement("p");
27      newsContent.textContent = this.getAttribute("content");
28
29      cardLeft.appendChild(autor);
30      cardLeft.appendChild(linkTitle);
31      cardLeft.appendChild(newsContent);
32      /*****_SEPARAÇÃO DE "DIVS" *****/
33      const cardRight = document.createElement("div");
34      cardRight.setAttribute("class", "card_right"); /*CSS e definindo as classes*/
35
36      const newsImage = document.createElement("img");
37      newsImage.src = (this.getAttribute("photo") || "assets/default.png");
38      newsImage.alt = "Foto da Notícia"
39
40      cardRight.appendChild(newsImage);
41
42      /*****_DEFININDO HERANÇAS_ *****/
43      /*3**/* Atrelando CardLeft e ...Right como filhas de Root*/
44      componentRoot.appendChild(cardLeft);
45      componentRoot.appendChild(cardRight);
46
47      // como se trata de "método" deve ter um return, pois não é VOID
48      return componentRoot;
```

Para definir elementos de forma a serem setados,

THIS.SETATTRIBUTE(e a forma que o atributo será chamado na tag)

14,2 Aplicando Estilo em Props

domingo, 15 de setembro de 2024 20:58

COM A TAG STYLES(){}

Lembrando que com componentes

Cada parte do código terá seu próprio CSS, dessa forma não ficará tão poluído o CSS.

Conseguira ver tudo de maneira Clean

Entao para Style...

```
styles()/*      Para definir os estilos e layouts dentro da Prop*/
  const style = document.createElement("style");
  style.textContent =
    DENTRO DESSE CONJUNTO DE CODIGOS, PODEMOS IMPLEMENTAR COMO SE FOSSE EM
    UM CSS NORMAL
  ;
  return style;
}
```

Fica desse jeito

```
desafio-4 > src > CardNews.js > Cardnews > build
  1  class Cardnews extends HTMLElement{
  51    styles(){/*      Para definir os estilos e layouts dentro da Prop*/
  52
  53    .card{
  54      width: 80%; /*720px valor real*//*100% porcentagem independente tamanho da tela*/
  55      display: flex;
  56      flex-direction: row; /*Separa as coisas em linhas, nao colunas*/
  57      justify-content: space-between;
  58      box-shadow: 10px 8px 4px 6px rgba(0,0,0,0.28);
  59      -webkit-box-shadow: 10px 8px 4px 6px rgba(0,0,0,0.28);
  60      -moz-box-shadow: 10px 8px 4px 6px rgba(0,0,0,0.28);
  61    }
  62  }
  63
  64  .card_left, .card_right{
  65    display: flex;
  66    flex-direction: column;
  67    justify-content: center;
  68  }
  69
  70  .card_left{
  71    /*AQUI ADICIONAMOS O ADICIONAL SOMENTE DE card_left!*/
  72    padding-left: 10px;
  73  }
  74
  75
  76  .card_left > span{
  77    font-weight: 400;
  78  }
  79
  80  .card_left > a {
  81    margin-top: 10px;
  82    font-size: 25px;
  83    text-decoration: none;
  84    font-weight: bold;
  85    color: black;
  86  }
  87
  88  .card_left > p {
  89    color: rgb(70, 70, 70);
  90  }
```

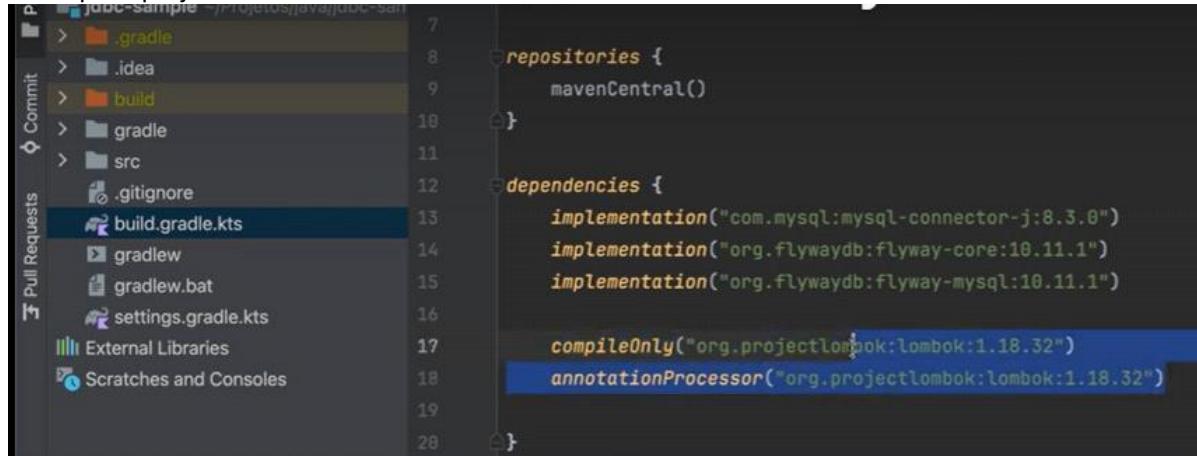
2 JDBC: Explorando Persistência de Dados com Java

sexta-feira, 20 de setembro de 2024 20:56

Para criar persistencia dentro do IntelliJ cm Java, necessario algumas informações

Como

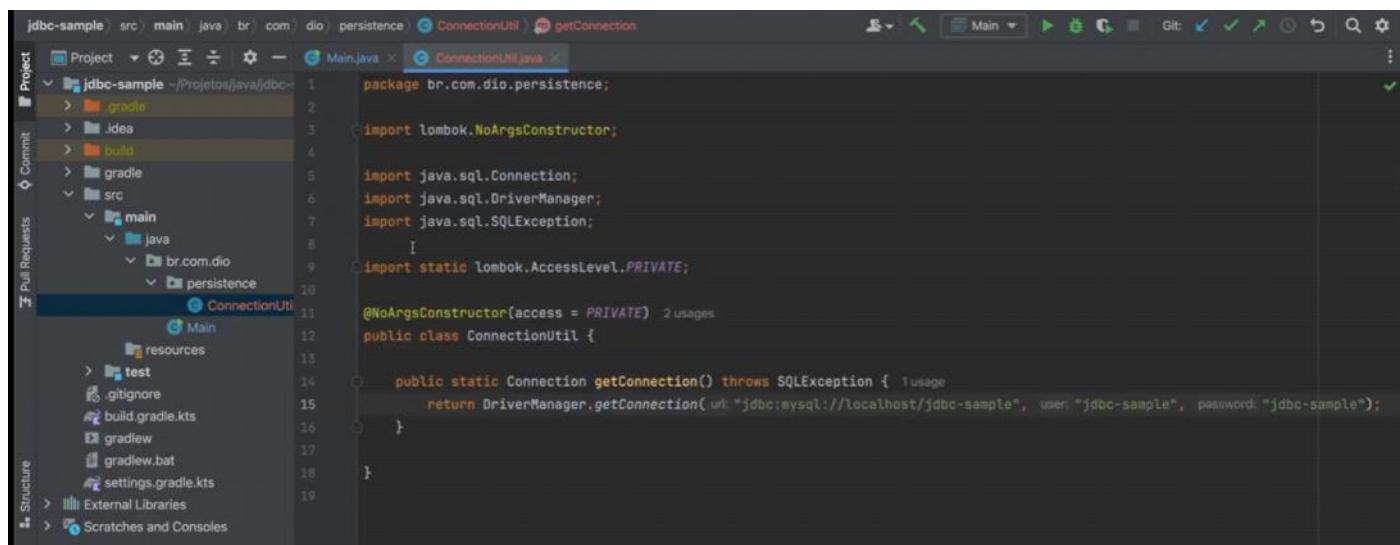
Note que o projeto em Graddle



```
repositories {
    mavenCentral()
}

dependencies {
    implementation("com.mysql:mysql-connector-j:8.3.0")
    implementation("org.flywaydb:flyway-core:10.11.1")
    implementation("org.flywaydb:flyway-mysql:10.11.1")

    compileOnly("org.projectlombok:lombok:1.18.32")
    annotationProcessor("org.projectlombok:lombok:1.18.32")
}
```



```
package br.com.dio.persistence;

import lombok.NoArgsConstructor;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

import static lombok.AccessLevel.PRIVATE;

@NoArgsConstructor(access = PRIVATE) 2 usages
public class ConnectionUtil {

    public static Connection getConnection() throws SQLException { 1 usage
        return DriverManager.getConnection("jdbc:mysql://localhost/jdbc-sample", "jdbc-sample", "jdbc-sample");
    }
}
```

2,1 Flyway

sábado, 21 de setembro de 2024 15:30

Trabalha com migrations

FlyWay é mais simples para montar configuração

Tem scripts para montar a migração DB de forma ordenada

Em vez de sempre montar o SQL, o flyway é uma das opções de migração

FLYWAY - simples em configuração de estrutura, quando não em uma configuração padrão

LINKBASE - mais complexo e necessita de configurações default prontas

2,2 JDBC

sábado, 21 de setembro de 2024 17:00

Java Database Vconnectivity

Java Database Connectivity ou JDBC é um conjunto de classes e interfaces (API) escritas em Java que fazem o envio de instruções SQL para qualquer banco de dados relacional

Dentro dos JDKs já temos os kits do JDBC

Então tudo que está nomeado como .SQL dentro do Java. Está dentro do JDBC

Dentro dele

Driver Manager - cria conexão

Connection - faz a criação do State Initiation

Sql Injection - Injeta a consulta de forma segura

Statement

Result Set - injeta a resposta com

Esses são interfaces

Exceto o Driver Manager que é uma classe dentro do jdbc .sql

No entanto o JDBC em si é uma interface, ou seja

Ela só existe, mas sem a devida chamada e implementação, de nada adianta

Essa ligação é um driver, que implementa um contrato entre o Banco de Dados (Postres MySql) e o back (Java)

Para essa ligação, podemos implementar uma dependência, se projeto Maven e não Graddle

<!-- <https://mvnrepository.com/artifact/mysql/mysql-connector-java> -->

<dependency>

 <groupId>mysql</groupId>

 <artifactId>mysql-connector-java</artifactId>

 <version>8.0.33</version>

</dependency>

Esse JDBC conversa com o driver e esse driver conversa com o MySql

2,3 ORM

terça-feira, 24 de setembro de 2024

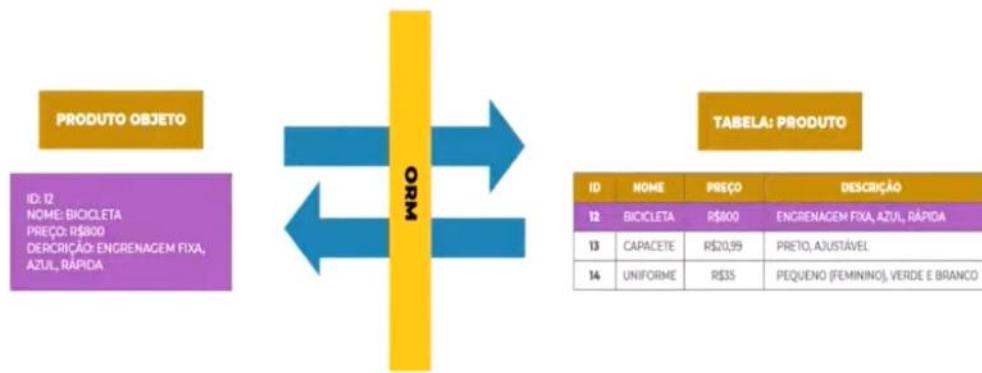
Em vez de ficar rodando While toda vez para ficar persistindo no banco e rodando a aplicação

Utilizamos do ORM pois é manipulando a classe como se fosse uma tabela

Object Relational Mapping



ORM



ORM é um Conceito Geral, em PHP == Laravel, Node.JS tem tb entre outras

2,4 JPA Persistence

terça-feira, 24 de setembro de 2024 22:34

Java Persistence API

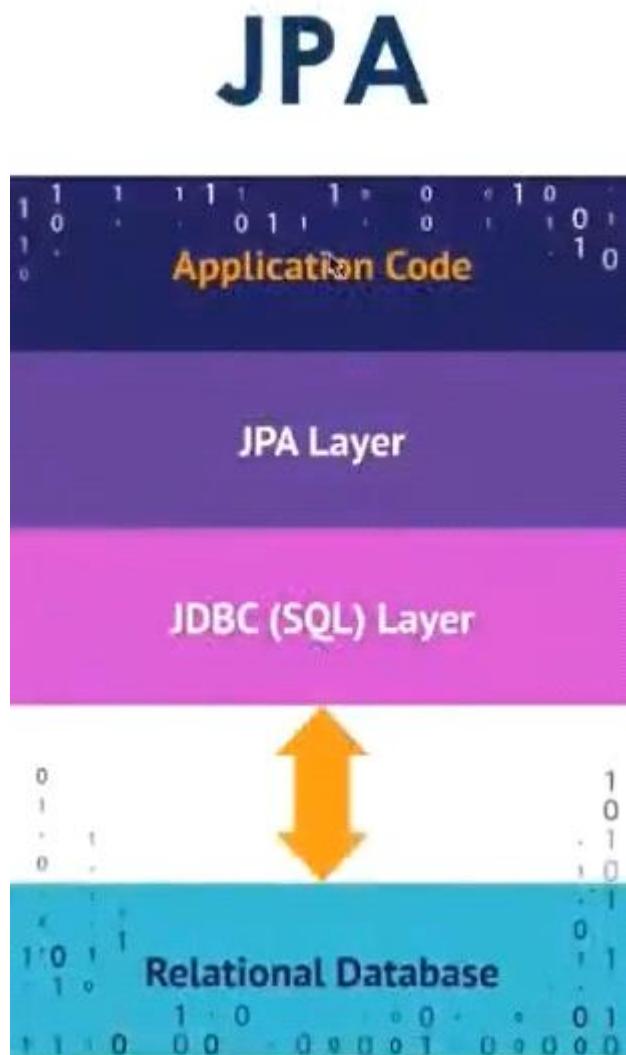
Descreve interface para persistencia em Banco de Dados

O JPA é um ORM

Define um meio de mapeamento objeto-relacional para objetos Java

Hibernate e JPA são bem alinhados,

o JPA é um padrao de conexao para BD com base no Hibernate



Aplicação Java

Camada JPA

JPA Traduzida com o JDBC

Conversacao entre JDBC e o Driver

Relação entre o Driver e o Banco de Dados

Para trabalhar com JPA,

Criar projeto Maven

DENTRO DE POM.XML

Inserir Dependencias HIBERNATE, BANCO DE DADOS (MY SQL, POSTGRESQL OU OUTRO)

Dentro de RESURCES, CRIAR UMA PASTA

Criar o META-INF

Dentro dele, o arquivo persistence.xml

The screenshot shows the IntelliJ IDEA interface with the project 'artigo-tutorial-jpa-master' open. The 'persistence.xml' file is selected in the editor. The code in the file is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<persistence xmlns="http://xmlns.jcp.org/xml/ns/persistence"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/persistence http://xmlns.jcp.org/xml/ns/persistence_2_2.xsd"
    version="2.2">
    <persistence-unit name="Clientes-PU">
        <properties>
            <!-- Url de conexão do banco de dados -->
            <property name="javax.persistence.jdbc.url" value="jdbc:mysql://localhost/db_cadastro_cliente" />
            <!-- Usuário de conexão do banco de dados-->
            <property name="javax.persistence.jdbc.user" value="root" />
            <!-- Senha do usuário de conexão do banco de dados-->
            <property name="javax.persistence.jdbc.password" value="root" />
            <!-- Driver do SGBD para conectar ao banco-->
            <property name="javax.persistence.jdbc.driver" value="com.mysql.cj.jdbc.Driver" />
            <!-- Dialeto é responsável por traduzir o mapeamento do hibernate para a linguagem SQL nativa do SGBD-->
            <property name="hibernate.dialect" value="org.hibernate.dialect.MySQL8Dialect" />
            <!-- Cria o banco de dados e encerra no final da sessão-->
            <property name="hibernate.hbm2ddl.auto" value="update" />
            <!-- Imprimir as consultas no log-->
            <property name="hibernate.show_sql" value="true" />
            <!-- Formatar a impressão das consultas no Log-->
            <property name="hibernate.format_sql" value="true" />
            <property name="hibernate.connection.autocommit" value="true" />
        </properties>
    </persistence-unit>

```

The bottom right corner of the screen shows the status bar with the text '6/214 char(s) 1F 1M 1C 1B 1S 1P master 2s'.

Lembrando que é só procurar como fazer uma persistencia já aparece como criar

Para cada banco de dados diferente, necessário criar um persistence.xml

The screenshot shows the IntelliJ IDEA interface with the project 'artigo-tutorial-jpa' open. The 'persistence.xml' file is selected in the editor. The code in the file is as follows, with specific database connection properties highlighted in red:

```
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/persistence http://xmlns.jcp.org/xml/ns/persistence/persistence_2_2.xsd"
version="2.2">
<persistence-unit name="Clientes-PU">
    <properties>
        <!-- Url de conexão do banco de dados -->
        <property name="javax.persistence.jdbc.url" value="jdbc:mysql://localhost/db_cadastro_cliente" />
        <!-- Usuário de conexão do banco de dados-->
        <property name="javax.persistence.jdbc.user" value="root" />
        <!-- Senha do usuário de conexão do banco de dados-->
        <property name="javax.persistence.jdbc.password" value="root" />
        <!-- Driver do SGBD para conectar ao banco-->
        <property name="javax.persistence.jdbc.driver" value="com.mysql.cj.jdbc.Driver" />
        <!-- Dialeto é responsável por traduzir o mapeamento do hibernate para a linguagem SQL nativa do SGBD-->
        <property name="hibernate.dialect" value="org.hibernate.dialect.MySQL8Dialect" />
        <!-- Cria o banco de dados e encerra no final da sessão-->
        <property name="hibernate.hbm2ddl.auto" value="update" />
        <!-- Imprimir as consultas no Log-->
        <property name="hibernate.show_sql" value="true" />
        <!-- Formatar a impressão das consultas no Log-->
        <property name="hibernate.format_sql" value="true" />
        <property name="hibernate.connection.autocommit" value="true" />
    </properties>
</persistence-unit>
```

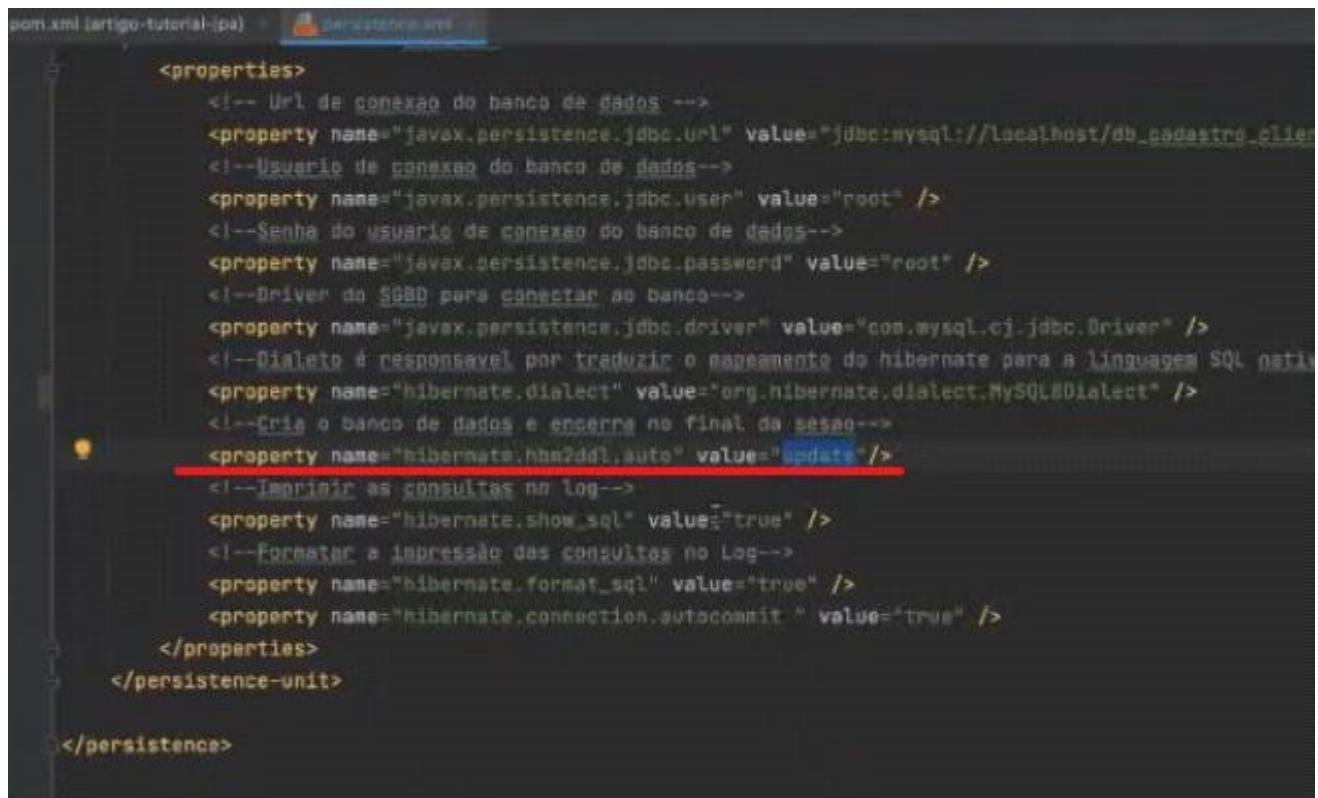
Nessa parte de implementação

O link do Banco de dados se ele já estiver no Servidor

O Usuario e senha, um em seguida do outro, não mais como inserido na aula JDBC

O driver MySQL ou se utilizar outro, o *driver dele*

Entre outros



```
pom.xml (artigo-tutorial-jpa)  PersistenceUnit
```

```
<properties>
    <!-- Url de conexão do banco de dados -->
    <property name="javax.persistence.jdbc.url" value="jdbc:mysql://localhost/db_cadastro_clientes" />
    <!-- Usuário de conexão do banco de dados -->
    <property name="javax.persistence.jdbc.user" value="root" />
    <!-- Senha do usuário de conexão do banco de dados -->
    <property name="javax.persistence.jdbc.password" value="root" />
    <!-- Driver do SGBD para conectar ao banco -->
    <property name="javax.persistence.jdbc.driver" value="com.mysql.cj.jdbc.Driver" />
    <!-- Dialeto é responsável por traduzir o mapeamento do hibernate para a linguagem SQL nativa -->
    <property name="hibernate.dialect" value="org.hibernate.dialect.MySQL8Dialect" />
    <!-- Cria o banco de dados e encerra no final da sessão -->
    <property name="hibernate.hbm2ddl.auto" value="update" />
    <!-- Imprimir as consultas no log -->
    <property name="hibernate.show_sql" value="true" />
    <!-- Formatar a impressão das consultas no Log -->
    <property name="hibernate.format_sql" value="true" />
    <property name="hibernate.connection.autocommit" value="true" />
</properties>
</persistence-unit>

</persistence>
```

ESSA PARTE É IMPORTANTE

SE DEFINIR value="create"

Ele só vai criar a tabela

Se inserir mais dados na tabela futuramente, não irá atualizar pois esta como Create somente

QUANDO DEFINIDO "update", ele define como...

Se ele já tem uma tabela no banco de dados, ele atualiza, se ele não tem ele cria, mas sempre nessa verificação

Se não existir ele cria

FORMAT SQL é para mostrar as tabelas criadas de forma organizada no Terminal da IDE em forma SQL

Autocommit true - quando abre uma atualização com o banco ele faz esse commit automático

3 Padrao de Projeto - Design Patterns

quinta-feira, 26 de setembro de 2024 18:24

Um padrão de projeto é uma proposta de solução para um problema existente, diferente de uma função, biblioteca ou framework, não é algo pronto, cabe a quem pretende aplicar no projeto estudar e entender como o padrão funciona.

Necessário sempre entender sobre cada ferramenta

As vezes refatorar um projeto em um outro framework ou em objetos, mas o ganho é pequeno em comparação ao tempo gasto

Design Patterns

Estruturais - Estrutura o projeto de forma que o projeto possa crescer sem muita complexibilidade

Crescendo de forma saudável para uma manutenção no futuro

Comportamentais - como um objeto vai se comportar no código, enxergar o comportamento dentro do código

<HTTPS://refactoring.guru/pt-br/design-patterns>

3,1 Conhecendo Solid

quinta-feira, 26 de setembro de 2024 18:35

Acronomo que cita 5 -principios

Serve para deixar o codigo de maneira mais robusta e de facil manutenção

Quando o codigo necessita um pouco mais de qualidade

Substituição de dados / components / interfaces / abstract ...

Substituição de Likov - independente da linguagem,
Ele é visto quando necessário a mudança de algum elemento no código

LSP – Likov Substitution Principle

O Princípio da substituição de Likov diz que objetos de um tipo base devem ser substituíveis por instâncias de um subtipo

ISP Interface Segregation Principle - Quando vc tem um método, como em um crud "FindByName" "...ById" "DeleteAll"

ISP – Interface Segregation Principle

O Princípio da segregação de interface diz que interfaces com muitas funcionalidades não coesas devem ser divididas em interfaces menores

DIP Dependency Inversion Principle

Diz que o sistema deve depender de interface e não de implementação específica

Baixo acoplamento

Mínimas Dependências

Vantagens

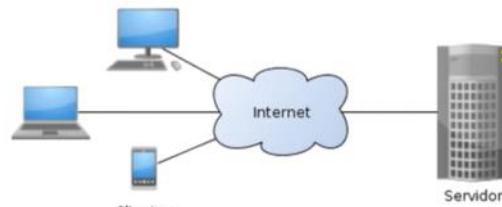
- Manutenção Simplificada;
- Reutilização de Código;
- Legibilidade Aprimorada;
- Redução de erros;
- Baixo acoplamento.

4 Cliente Servidor

sexta-feira, 27 de setembro de 2024 09:24

Modelo Cliente-Servidor

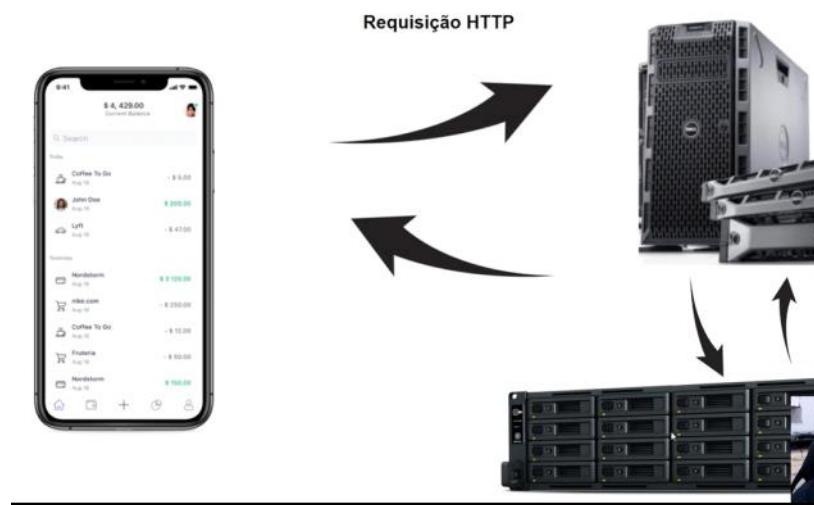
O modelo cliente-servidor é uma estrutura de aplicação que distribui as tarefas e cargas de trabalho entre os fornecedores de um recurso ou serviço, designados como servidores, e os requerentes dos serviços, designados como clientes.



Estrutura que distribui as cargas entre os fornecedores de recursos/serviços através de servidores ou computadores designados como clientes

Num software, a primeiro modo

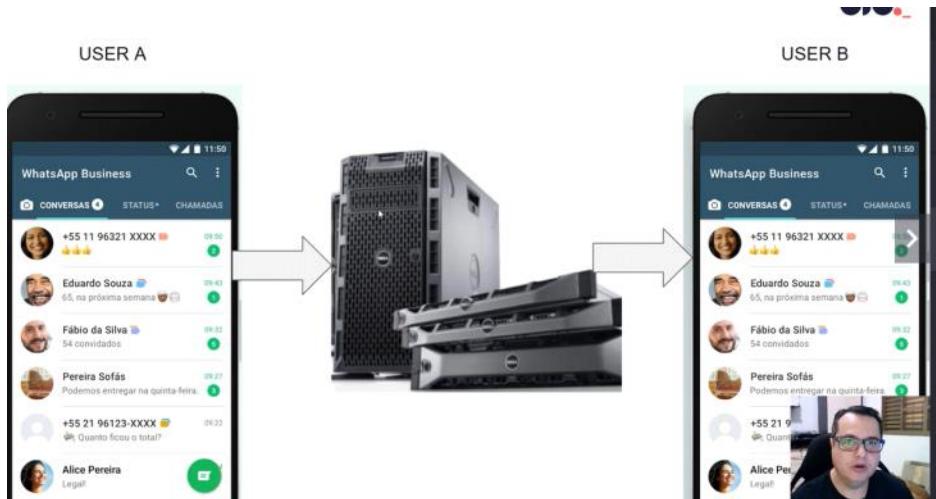
Temos que ter a clareza que temos 1 Servidor que distribuirá para vários clientes



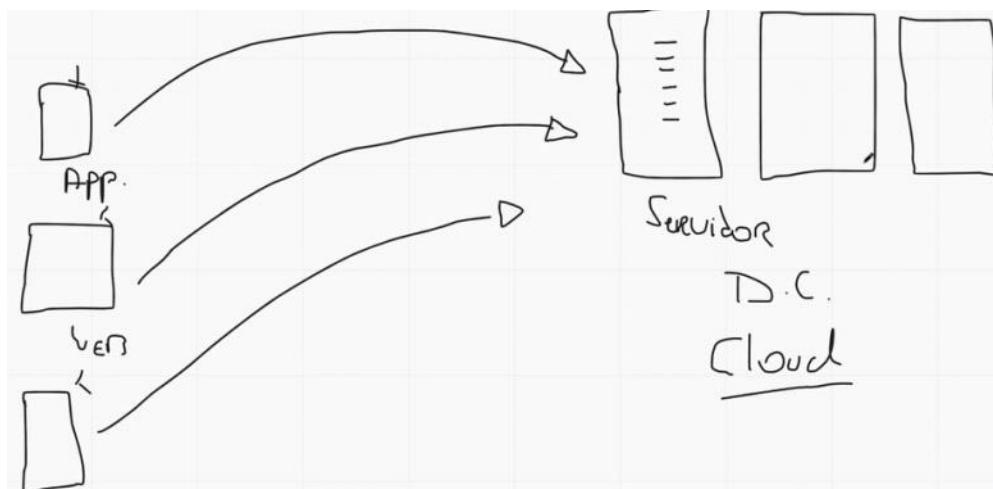
Quando é uma conversa/comunicação entre dispositivos

Os 2 precisam de uma identificação

Enviam a info p servidor o servidor pega e manda



Necessario pensar no software que desenvolve back que gera essas infor,
o front que recebe as informações
E a modelagem de storage onde ficam guardados os dados



Quando uma aplicação
Elas dependem de um servidor Banco de Dados ou Data Center

Esse servidor pode ser fisico ou Cloud
Fisico tem custos como Energia 24h, Local, refrigeração
Cloud tem custos de permanencia no host

Quando muitos servidores, o custo pode ser bem alto
Por esse motivo, o ideal é optar por Cloud

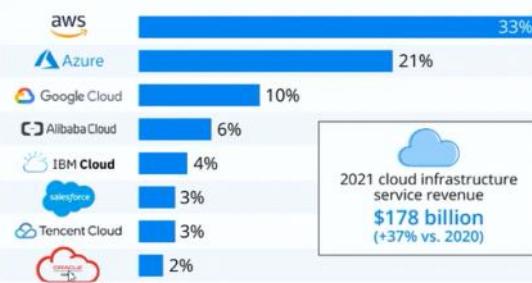
Cloud

A cloud computing é o acesso sob demanda, via internet, a recursos de computação — aplicativos, servidores (físicos e virtuais), armazenamento de dados, ferramentas de desenvolvimento, recursos de rede e muito mais — hospedados em um data center remoto gerenciado por um provedor de serviços em cloud (Cloud Solution Provider). O CSP disponibiliza esses recursos por uma assinatura mensal ou por um valor **cobrado conforme o uso**.

Servidores Cloud - Amazon, Azure, Google Cloud Platform
CSP disponibiliza mensal conforme pessoa utiliza, se utiliza 1 hora, paga por uma hora

Amazon Leads \$180-Billion Cloud Market

Worldwide market share of leading cloud infrastructure service providers in Q4 2021*



* includes platform as a service (PaaS) and infrastructure as a service (IaaS)
as well as hosted private cloud services

Source: Synergy Research Group



statista

4,1 Virtualização

sábado, 28 de setembro de 2024 00:55

Com base no que já visto em Cliente Servidor

Independente se Nuvem ou CPD

O custo de um servidor é caro, agora cada servidor um para cada aplicação, piorou...

A partir disso foi idealizado o conceito VM
Virtualização

Quando temos um Hardware, com a virtualização, podemos potencializar e dividir espaço

50 GB **TOTAL**

5 GB para uma aplicação

10 GB para outra aplicação

8 GB...

Dividimos o espaço para mais aplicações, dessa forma evitando gastos desnecessários e prevendo expansão de banco

Virtualização

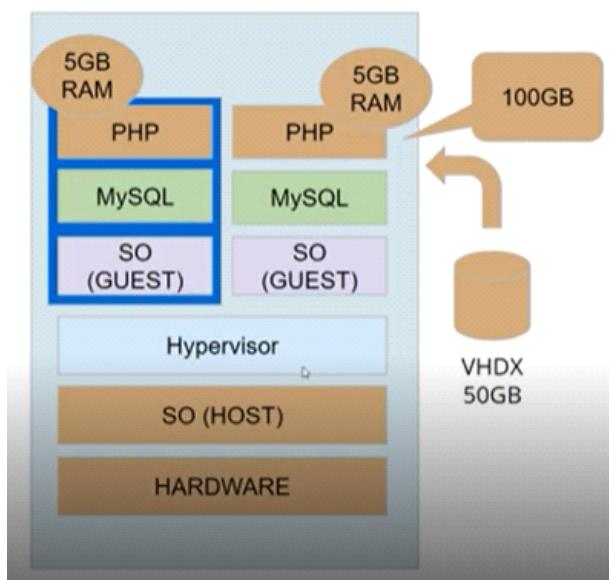
A virtualização utiliza software para criar uma camada de abstração sobre o hardware do computador, permitindo que os recursos de hardware de um único computador (processadores, memória, armazenamento, etc) sejam divididos em vários computadores virtuais.

Softwares de Virtualização

VM Ware

Microsoft Hyper V

Virtualização



VHDX 50GB é o tamanho do disco físico

Tenho um Hardware

O Hardware tem um Sistema Operacional

Esse SO tem um programa de virtualização (No caso Hypervisor)

Dentro dele, podemos criar as VMs, uma delas como exemplo Windows outra Linux

Uma tem um BD MySQL o outro PostgresSQL

Uma delas tem C# e PHP outra Java e Angular

4,2 Microservicos

sábado, 28 de setembro de 2024 01:22

Com base em Virtualização

No entanto quando separamos os discos, acabamos pesando a maquina

Podemos potencializar a VM de forma a ficar mais leve na hora de rodar a VMs

Microserviços

Microserviços são uma abordagem arquitetônica e organizacional do desenvolvimento de software na qual o software consiste em pequenos serviços independentes que se comunicam usando APIs bem definidas. Esses serviços pertencem a pequenas equipes autossuficientes.

As arquiteturas de microserviços facilitam a escalabilidade e agilizam o desenvolvimento de aplicativos, habilitando a inovação e acelerando o tempo de introdução de novos recursos no mercado.

Tendo microserviços, podemos clonar VMs a partir de "endpoints" com esse microserviço

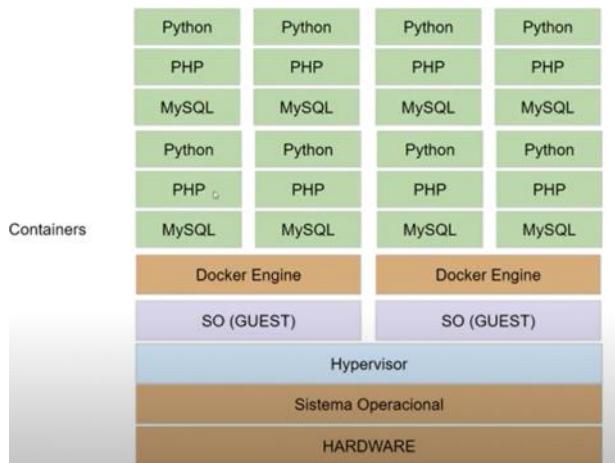
Eles podem ser feitos e montados em OVA

Os microserviços podem rodar também VMs onde os containers irão agilizar e rodar partes da aplicação

4,3 Containers

sábado, 28 de setembro de 2024 01:53

Atraves deles que os microservicos podem acontecer em componentes



Esse Container é o Docker, o Docker que resolve esse problema

Qual é a diferença entre virtualização e os containers?

As duas tecnologias são distintas porém complementares. Veja uma maneira fácil de distinguir ambas:

- Com a virtualização, é possível executar sistemas operacionais (Windows ou Linux) simultaneamente em um único sistema de hardware
- Os containers compartilham o mesmo kernel do sistema operacional e isolam os processos da aplicação do restante do sistema. Os containers Linux são extremamente portáteis, mas devem ser compatíveis com o sistema subjacente.



4,7 Docker

sábado, 28 de setembro de 2024 01:56

Para implementar o Docker dentro de uma VM Linux, necessário fazer as seguintes configurações

NÃO É DOCKER DESKTOP, SIM DOCKER ENGINE

The screenshot shows the 'Install Docker Engine' page from the Docker documentation. The left sidebar is titled 'Manuals' and lists various Docker components like Docker Build, Docker Compose, Docker Desktop, and Docker Engine, with 'Install' selected. The main content area has a heading 'Install Docker Engine' and a sub-section 'Supported platforms'. A table maps operating systems to supported architectures:

Platform	x86_64 / amd64	arm64 / aarch64	arm (32-bit)	ppc64le	s390x
CentOS	✓	✓	✓	✓	
Debian	✓	✓	✓	✓	
Fedora	✓	✓	✓	✓	
Raspberry Pi OS (32-bit)			✓		
CentOS	✓				
SLES (s390x)				✓	
Binaries	✓	✓	✓	✓	✓
Ubuntu	✓	✓	✓	✓	✓
Binaries	✓	✓	✓		

Para rodar os comandos de instalação

```
curl -fsSL https://get.docker.com -o get-docker.sh  
sudo sh ./get-docker.sh --dry-run
```

Para instalação

```
Sh get-docker.sh
```

Se rodar o comando docker version

```
andre@ubuntu-server:~$ docker version  
Client: Docker Engine - Community  
  Version: 27.3.1  
  API version: 1.47  
  Go version: go1.22.7  
  Git commit: ce12230  
  Built: Fri Sep 20 11:40:59 2024  
  OS/Arch: linux/amd64  
  Context: default  
  permission denied while trying to connect to the Docker daemon socket at unix:///var  
  l unix /var/run/docker.sock: connect: permission denied  
andre@ubuntu-server:~$ _
```

Ele já está rodando

Para descobrir se está ativo

```

andre@ubuntu-server:~$ systemctl status docker
● docker.service - Docker Application Container Engine
  Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: enabled)
  Active: active (running) since Sun 2024-09-29 05:11:10 UTC; 2min 23s ago
TriggeredBy: • docker.socket
    Docs: https://docs.docker.com
   Main PID: 2113 (dockerd)
     Tasks: 9
    Memory: 21.8M (peak: 24.0M)
      CPU: 2.192s
     CGroup: /system.slice/docker.service
             └─2113 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

set 29 05:11:08 ubuntu-server dockerd[2113]: time="2024-09-29T05:11:08.454311067Z"
set 29 05:11:08 ubuntu-server dockerd[2113]: time="2024-09-29T05:11:08.458199784Z"
set 29 05:11:08 ubuntu-server dockerd[2113]: time="2024-09-29T05:11:08.812361728Z"
set 29 05:11:10 ubuntu-server dockerd[2113]: time="2024-09-29T05:11:10.406342016Z"
set 29 05:11:10 ubuntu-server dockerd[2113]: time="2024-09-29T05:11:10.521160726Z"
set 29 05:11:10 ubuntu-server dockerd[2113]: time="2024-09-29T05:11:10.521588962Z"
set 29 05:11:10 ubuntu-server dockerd[2113]: time="2024-09-29T05:11:10.521703823Z"
set 29 05:11:10 ubuntu-server dockerd[2113]: time="2024-09-29T05:11:10.521996193Z"
set 29 05:11:10 ubuntu-server systemd[1]: Started docker.service - Docker Application
set 29 05:11:10 ubuntu-server dockerd[2113]: time="2024-09-29T05:11:10.717334755Z"
lines 1-22/22 (END)

```

Para acessar o Docker

Para acessar o Docker no Ubuntu Server, siga as orientações abaixo:

Acesso ao Docker via SSH

Se você já está acessando o **Ubuntu Server** via SSH com o IP fornecido pela VM (como **`192.168.100.83`**), você pode diretamente rodar comandos do Docker dentro dessa sessão SSH. Não é necessário acessar um IP diferente.

Comandos Docker:

Uma vez conectado via SSH, você pode verificar se o Docker está instalado e funcionando com o seguinte comando:

```

```bash
sudo docker ps
```

```

Isso listará todos os contêineres em execução. Se você quiser acessar um contêiner específico, pode usar:

```

```bash
sudo docker exec -it <container_name_or_id> /bin/bash
```

```

Isso abrirá um shell interativo dentro do contêiner.

Sobre o IP 172.17.0.1

Esse IP (**`172.17.0.1`**) faz parte da rede interna criada pelo Docker para comunicação entre os contêineres e o host. Você **não precisa acessar diretamente esse IP** para gerenciar o Docker. Ele é apenas o gateway interno para contêineres. Se você precisa gerenciar o Docker, sempre faça isso pelo IP que você usa para acessar o Ubuntu Server (como **`192.168.100.83`**), e então utilize os comandos Docker dentro da sessão SSH.

Resumo:

- Use o IP **`192.168.100.83`** (o IP da VM) para acessar o Ubuntu Server via SSH.
- Dentro do SSH, use comandos como **`sudo docker ps`** para gerenciar o Docker.
- O IP **`172.17.0.1`** é a rede interna do Docker, você não precisa acessá-lo diretamente.

Se precisar de mais algum detalhe ou comando específico, estou à disposição!

4,4 Configurando ambiente para VMs

sábado, 28 de setembro de 2024 13:34

Primeiro necessário ter ambiente de criação de VM

Utilizado no vídeo o Oracle Virtual Box

 Oracle
<https://www.oracle.com/br/virtualization/virtualbox/>

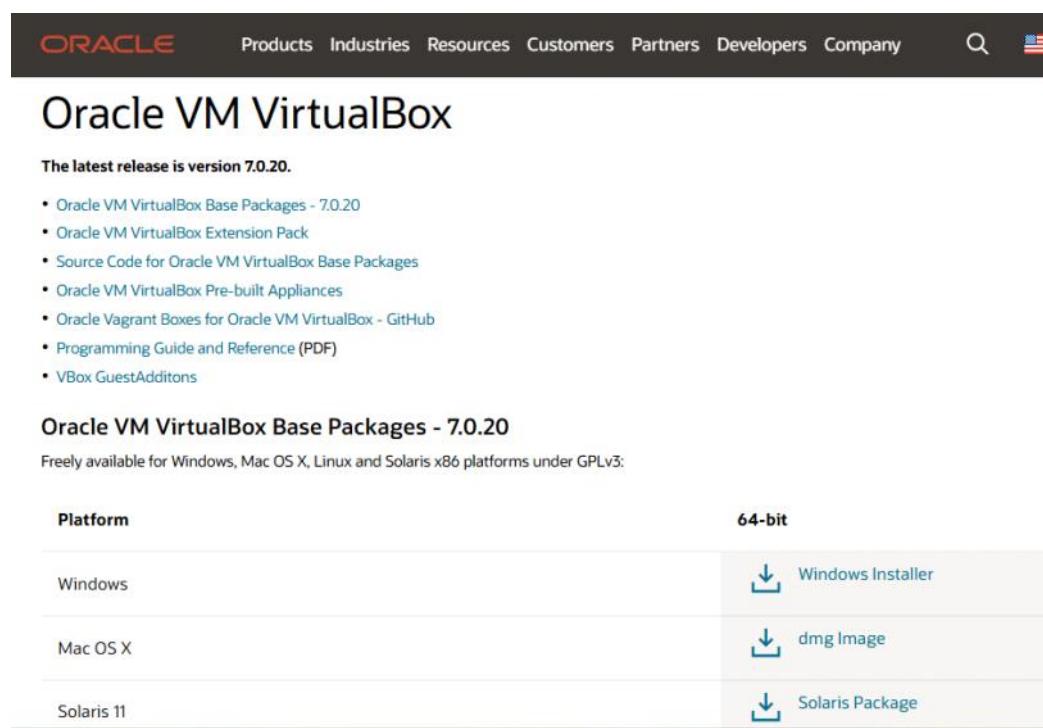
[VM VirtualBox | Virtualização | Oracle Brasil](#)

WEB A Oracle VM VirtualBox permite que os desenvolvedores e as equipes de TI executem vários sistemas operacionais em um único dispositivo, reduzindo custos e complexidade. Saiba ...

 Oracle
<https://www.oracle.com/.../vm/downloads/virtualbox-download...>

[Oracle VM VirtualBox - Downloads](#)

WEB Download the latest version of Oracle VM VirtualBox, a free and open source virtualization software for Windows, Mac OS X, Linux and Solaris x86 platforms. Choose from base ...



The screenshot shows the Oracle VM VirtualBox download page. At the top, there's a navigation bar with links for Products, Industries, Resources, Customers, Partners, Developers, Company, a search icon, and an American flag icon. The main title is "Oracle VM VirtualBox". Below it, a message says "The latest release is version 7.0.20." followed by a bulleted list of links: Oracle VM VirtualBox Base Packages - 7.0.20, Oracle VM VirtualBox Extension Pack, Source Code for Oracle VM VirtualBox Base Packages, Oracle VM VirtualBox Pre-built Appliances, Oracle Vagrant Boxes for Oracle VM VirtualBox - GitHub, Programming Guide and Reference (PDF), and VBox GuestAdditions. Underneath, a section titled "Oracle VM VirtualBox Base Packages - 7.0.20" states "Freely available for Windows, Mac OS X, Linux and Solaris x86 platforms under GPLv3:". A table lists the download options:

| Platform | 64-bit |
|------------|---|
| Windows |  Windows Installer |
| Mac OS X |  dmg Image |
| Solaris 11 |  Solaris Package |

Através de uma Virtual Box, necessário instalar o Linux Ubuntu versão Server

The screenshot shows the Canonical Ubuntu website at https://ubuntu.com/download/server. The top navigation bar includes links for Felipe Silk, Remove.bg, EDUCAÇÃO, Vagas, COMPRAS/VENDAS, Filmes Series Animes, and Leilões. The main menu has categories for Desktop, Server, Core, and Cloud, with Server being the active tab. Below the menu, there are three installation options: Manual installation, Instant VMs, and Automated provisioning, with Manual installation being selected.

Get Ubuntu Server

Manual installation Instant VMs Automated provisioning

Ubuntu 24.04.1 LTS

The latest LTS version of Ubuntu Server. LTS stands for long-term support — which means five years of free security and maintenance updates, extended to 10 years with Ubuntu Pro.

[Download 24.04.1 LTS](#) 2.6GB

[Alternative downloads >](#)

[Alternative architectures >](#)

[What's new](#) [System requirements](#) [How to install](#)

Linux 6.8 kernel with low latency kernel features enabled by default

Frame pointers enabled by default for the majority of packages on x86 architectures

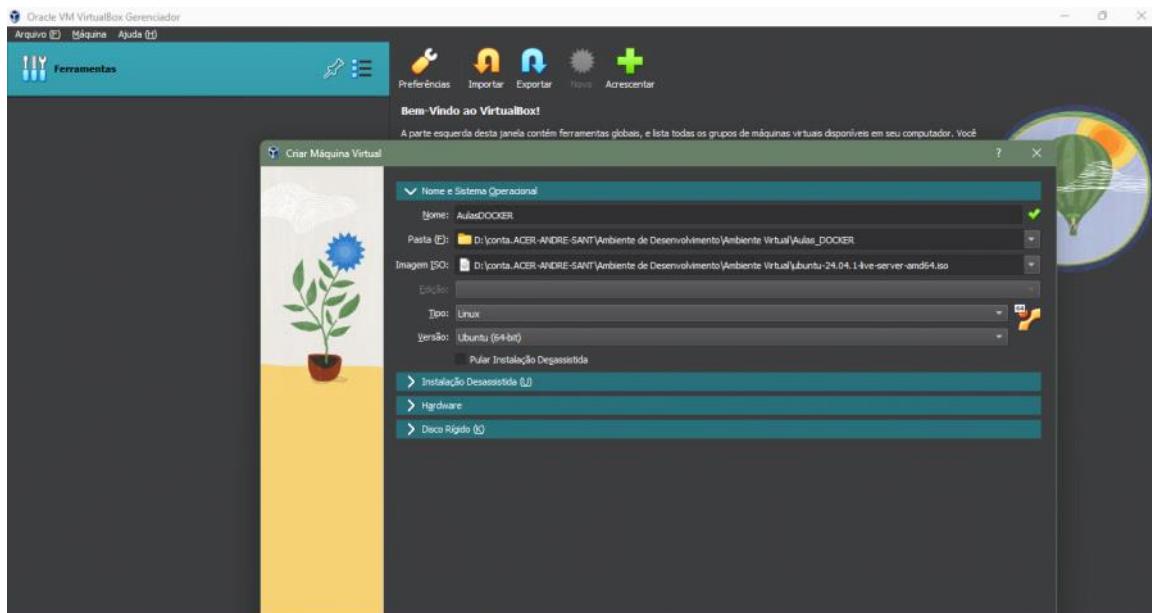
Rust 1.75, .NET 8 and OpenJDK 21 with TCK certification in addition to other toolchain upgrades

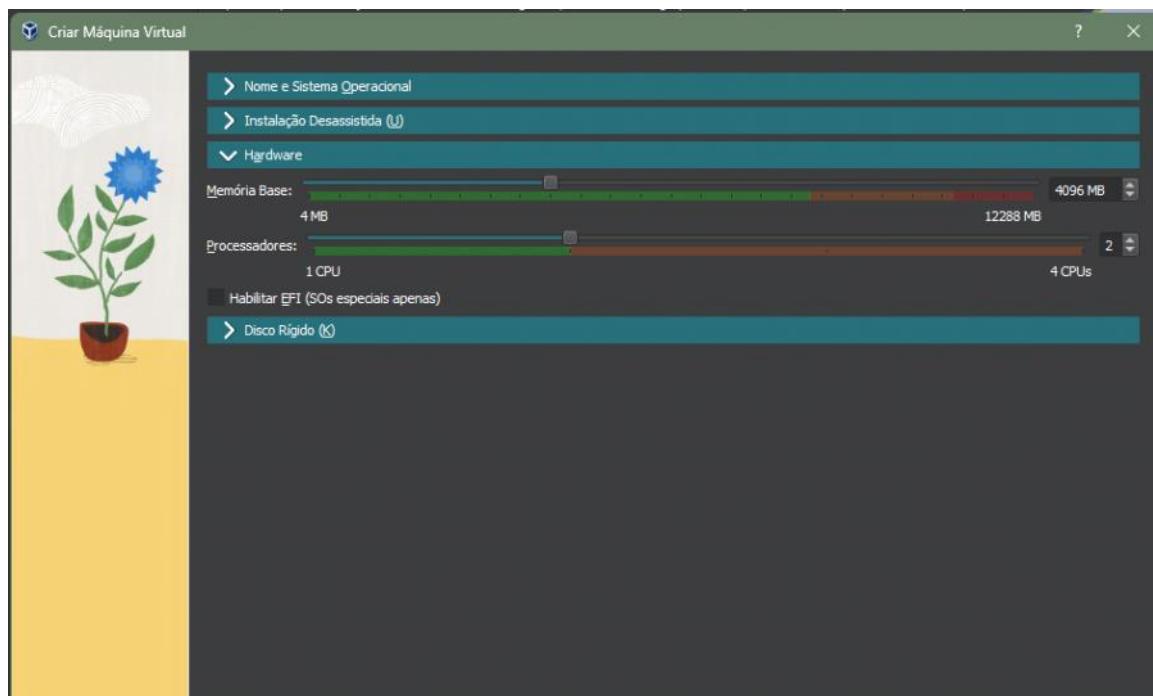
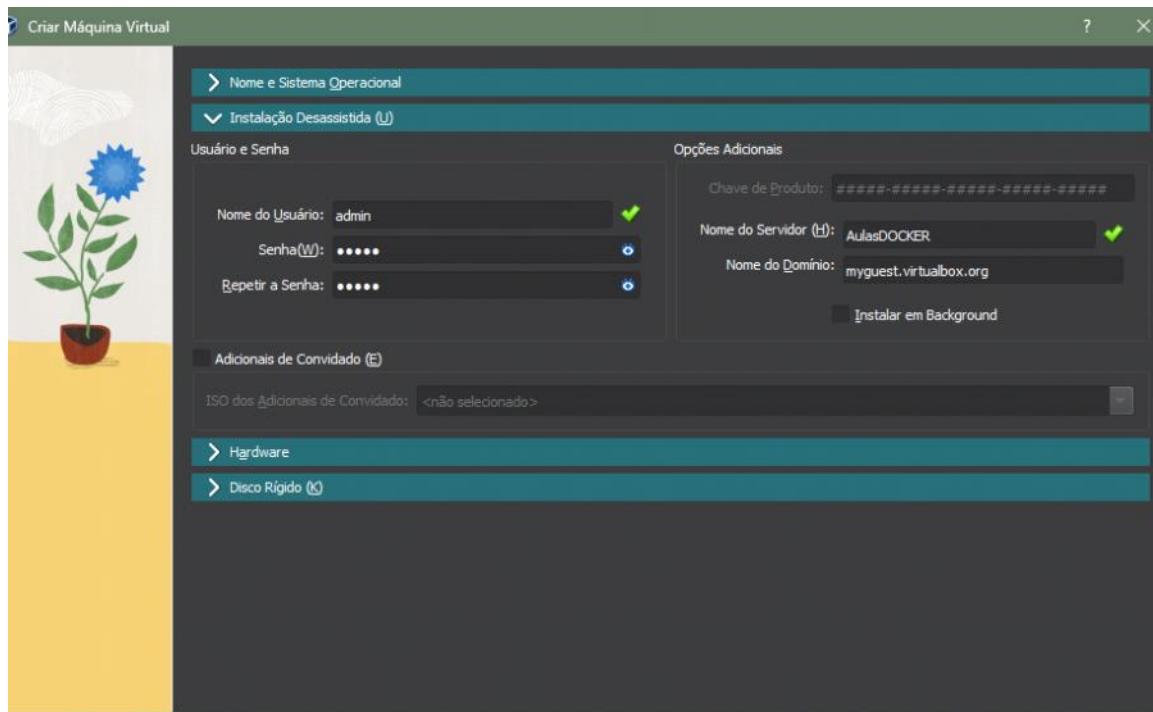
64-bit time_t by default on armhf to address the year 2038 issue

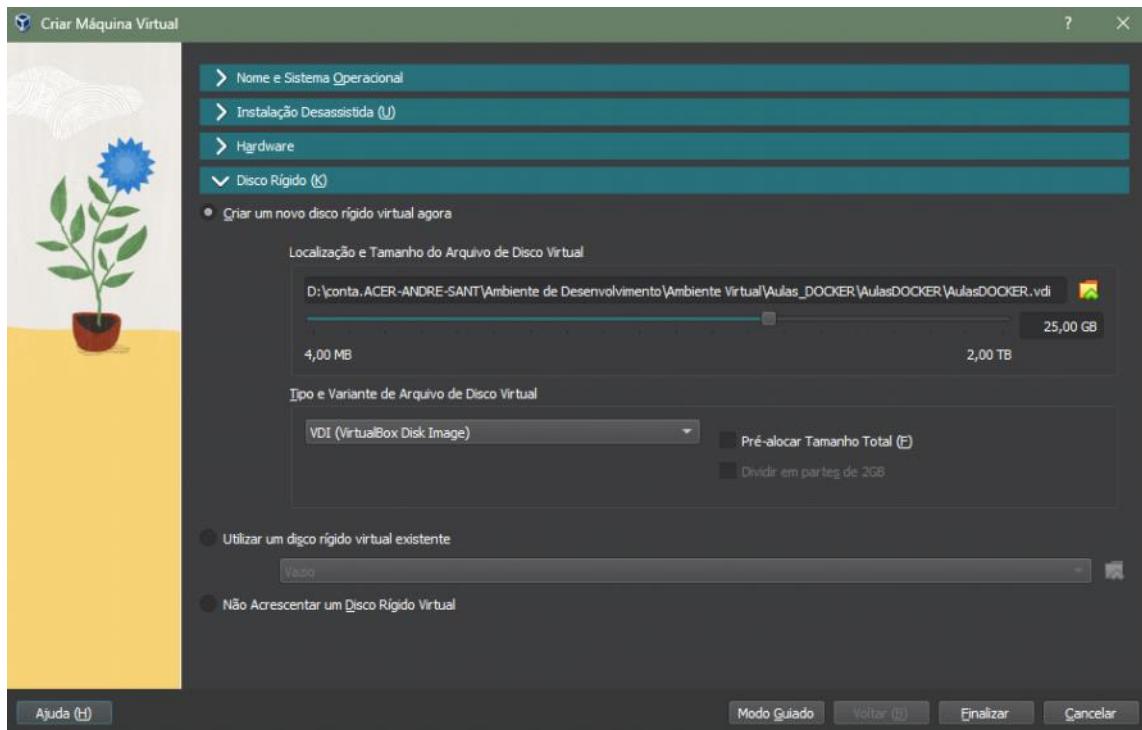
[server/thank-you?version=24.04.1&architecture=amd64<s=true](#)

Assim que a imagem estiver baixada em seu PC, necessário ter em mente que utilizara uma porcentagem da RAM do seu PC para esse S.O

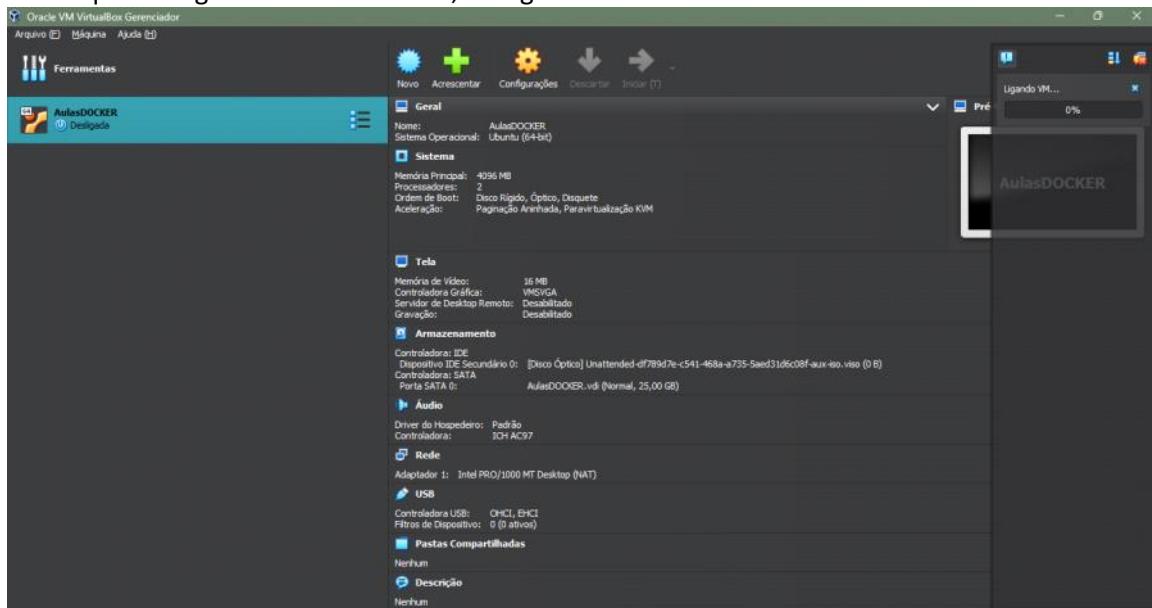
Clicando em NEW







Assim que configurado corretamente, ele ligara a VM LocalHost



4,8 Documentos para Docker

domingo, 29 de setembro de 2024 19:36

No docker hub - instalamos uma imagem do Ubuntu

Há mais informações sobre o docker
porém não consegui prosseguir com esse conteúdo
pois estou me batendo demais com o Linux
futuramente vou estudar mais voltado ao Windows
pois facilita na hora do aprendizado

4,5 Instalando e Habilitando o Remote Desktop na VM

domingo, 29 de setembro de 2024 19:42

Sim, você pode realizar todos esses passos através do **PuTTY**, acessando o Ubuntu Server via SSH, já que o **PuTTY** é uma ferramenta para conexões SSH. A única coisa que você fará no **PuTTY** é rodar os comandos para instalar e configurar o Remote Desktop (RDP) no servidor.

Aqui está um resumo do que você fará usando o **PuTTY**:

1. **Acessar o servidor via SSH** usando o **PuTTY** (você já está acessando com o IP **`192.168.100.83`**).
2. **Executar os comandos** mencionados anteriormente no **PuTTY** para instalar o ambiente gráfico (Xfce), o servidor RDP (xrdp), e configurar o sistema. Por exemplo:

- Instalar o ambiente gráfico:

```
```bash
sudo apt update
sudo apt install xfce4 xfce4-goodies
sudo apt install xrdp
````
```

- Configurar o **xrdp** para usar o **Xfce**:

```
```bash
sudo nano /etc/xrdp/startwm.sh
````
```

(Depois, modifique o arquivo conforme indicado anteriormente).

3. **Iniciar e habilitar o xrdp**:

```
```bash
sudo systemctl enable xrdp
sudo systemctl start xrdp
````
```

4. **Permitir o acesso à porta 3389 no firewall**, caso esteja usando o UFW:

```
```bash
sudo ufw allow 3389
````
```

Depois de concluir esses passos via **PuTTY**, você poderá fechar a sessão SSH e usar um cliente **Remote Desktop** (RDP) no seu PC para acessar a interface gráfica do Ubuntu Server.

Lembre-se de usar o IP **`192.168.100.83`** ao se conectar pelo cliente RDP.

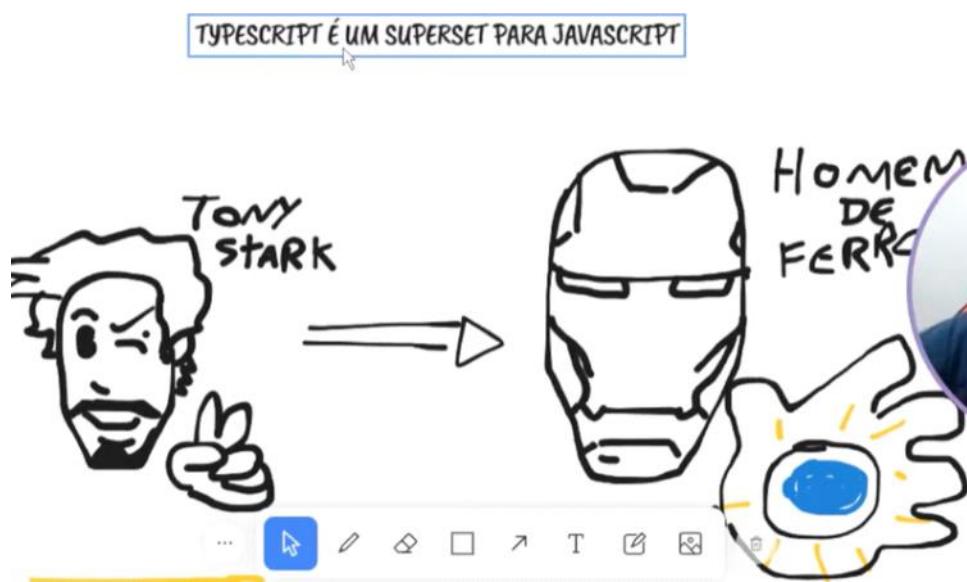
15 - Typescript - Performance do JS

segunda-feira, 7 de outubro de 2024 17:18

Da para pensar que o TS é um JS aprimorado,
Onde não permitira lançar a aplicação com erro

É literalmente uma TIPAGEM DO JAVASCRIPT == TypeScript

Para mostrarmos isso, vamos primeiro iniciar com TS para depois ingressar de fato no Angular



O JS seria o Tonny

O TS seria o IRON MAN

Com o Typescript podemos tipar o JS

Isso serve para definir tipos previsíveis da variável

No TS temos Classes como no Back

Classes podem ser interfaces que todos que trabalharem com ela, precisam respeitar

No TS temos o ENUMS também passando tipos previsíveis de maneira mais fácil

15,2 - Instalando e Configurando o TS

segunda-feira, 7 de outubro de 2024 17:49

Para instalar o TS, podemos entrar no site do TS

Clicando em Download

The screenshot shows the official TypeScript website at <https://www.typescriptlang.org/#~:text=TypeScript%20extends%20JavaScript%20by%20adding%20types>. The 'Download' section is highlighted. It contains three main points: 1) TypeScript adds additional syntax to JavaScript to support a **tighter integration with your editor**. 2) TypeScript code converts to JavaScript, which **runs anywhere**. 3) TypeScript understands JavaScript and uses **type inference** to provide **great tooling**.

Get Started



Se já temos o NODE JS instalado na maquina
Automaticamente temos o NPM e o NPX

Se caso não. Necessario instalar o node

Temos duas maneiras de baixar o TS, atraves do projeto, ou mais abaixo no site, de forma global na maquina

TypeScript em seu projeto

Ter o TypeScript configurado por projeto permite que você tenha muitos projetos com muitas versões diferentes do TypeScript, o que mantém cada projeto funcionando de forma consistente.

via npm

O TypeScript está disponível como um [pacote no registro npm](#) disponível como `."typescript"`

Você precisará de uma cópia do [Node.js](#) como um ambiente para executar o pacote. Em seguida, você usa um gerenciador de dependências como [npm](#), [yarn](#) ou [pnpm](#) para baixar o TypeScript em seu projeto.

```
npm install typescript --save-dev
```

`npm` `yo` `pnpm`

Todos esses gerenciadores de dependência oferecem suporte a arquivos de bloqueio, garantindo que todos em sua equipe estejam usando a mesma versão da linguagem. Em seguida, você pode executar o compilador TypeScript usando um dos seguintes comandos:

```
npx tsc
```

`npm` `yo` `pnpm`

com o Visual Studio

Para a maioria dos tipos de projeto, você pode obter o TypeScript como um pacote no NuGet para seus projetos do MSBuild, por exemplo, um aplicativo ASP.NET Core.

Ao usar o NuGet, você pode [instalar o TypeScript por meio do Visual Studio](#) usando:

- A janela Gerenciar Pacotes NuGet (que você pode acessar clicando com o botão direito do mouse em um nó do projeto)
- O Console do Gerenciador de Pacotes NuGet (encontrado em Ferramentas > Gerenciador de Pacotes NuGet > Console do Gerenciador de Pacotes) e, em seguida, executando:
`Install-Package Microsoft.TypeScript.MSBuild`

Para tipos de projeto que não dão suporte ao Nuget, você pode usar a [extensão TypeScript do Visual Studio](#). Você pode instalar a extensão usando no Visual Studio. [Extensions > Manage Extensions](#)

Instalando globalmente o TypeScript

Pode ser útil ter o TypeScript disponível em todos os projetos, geralmente para testar ideias pontuais. A longo prazo, as bases de código devem preferir uma instalação em todo o projeto em vez de uma instalação global para que possam se beneficiar de compilações reproduzíveis em diferentes máquinas.

via npm

Você pode usar o npm para instalar o TypeScript globalmente, isso significa que você pode usar o comando em qualquer lugar do seu terminal. `tsc`

Para fazer isso, execute `.`. Isso instalará a versão mais recente (atualmente 5.6). `npm install -g typescript`

por meio do Visual Studio Marketplace

Você pode instalar o TypeScript como uma extensão do Visual Studio, o que permitirá que você use o TypeScript em muitos projetos do MSBuild no Visual Studio.

A versão mais recente está disponível [no Visual Studio Marketplace](#).

Quando quero rodar o TS dentro do projeto, podemos fazer como da primeira print

Vai que em cada projeto, precisemos de uma versao diferente

15,1 TS o que melhora?

segunda-feira, 7 de outubro de 2024 18:15

Em JS podemos ter a seguinte performance

O primeiro retorno com o numero

O segundo sem numero

O primeiro retornou um numero, o segundo não encontrou, mas ao inves de retornar erro, retornou Undefined

The screenshot shows a Visual Studio Code interface. The top bar has tabs for 'JS caseComJS.js' and 'caseComJS.js'. The main area is a code editor with the following content:

```
1 Aulas-JS > JS caseComJS.js > ...
2   function ligar(heroi){
3     |   console.log("Ligando Para " + heroi.telefone);
4   }
5
6   ligar({
7     |   nome: "Steve Rogers",
8     |   vulgo: "Capitão América",
9   });

```

The 'TERMINAL' tab is selected at the bottom, showing command-line output:

```
PS D:\conta.ACER-ANDRE-SANT\Ambiente de Desenvolvimento\DIOS - Bootcamps\Bootcamp_Springboot_Angular DEAL\Bootcamp_Springboot_Angular DEAL\Aulas-JS> node .\caseComJS.js

Ligando Para 11 31713388
PS D:\conta.ACER-ANDRE-SANT\Ambiente de Desenvolvimento\DIOS - Bootcamps\Bootcamp_Springboot_Angular DEAL\Bootcamp_Springboot_Angular DEAL\Bootcamp_Springboot_Angular DEAL\Aulas-JS> node .\caseComJS.js

Ligando Para undefined
PS D:\conta.ACER-ANDRE-SANT\Ambiente de Desenvolvimento\DIOS - Bootcamps\Bootcamp_Springboot_Angular DEAL\Bootcamp_Springboot_Angular DEAL\Bootcamp_Springboot_Angular DEAL\Aulas-JS>
```

Com TS podemos criar tipo CONSTRUTOR onde conseguimos definir o que é necessário

```
caseComTS.ts
1  type Hero = {
2      nome: string,
3      vulgo: string,
4      telefone: string,
5  };
6
7  function ligarPara(heroi: Hero){
8      console.log("Ligando Para " + heroi.telefone);
9  }
10
11 ligarPara({
12     nome: "Rogers",
13     vulgo: "Capitão América",
14     telefone: "11 991173886",
15 });

```

Dessa forma, tipando-o, podemos definir o que é e o que não está certo, como ali acusando que não tem o Telefone.

Sabemos que precisamos inserir o telefone

The screenshot shows a code editor with two tabs: 'caseComJS.js' and 'caseComTS.ts'. The 'caseComTS.ts' tab is active, showing the following code:

```
Aulas-JS > ts caseComTS.ts > ...
1 type Hero = {
2     nome: string,
3     vulgo: string,
4     telefone: string,
5 };
6
7 Argument of type '{ nome: string; vulgo: string; }' is not assignable to type 'Hero'.
8   Property 'telefone' is missing in type '{ nome: string; vulgo: string; }'.
9 'Hero'.ts(2345)
10 caseComTS.ts(4, 5): 'telefone' is declared here.
11
12 View Problem (Alt+F8) Quick Fix... (Ctrl+.)
13
14     vulgo: "Capitão América",
15 
```

A tooltip is displayed over the line '14' with the text 'Argument of type '{ nome: string; vulgo: string; }' is not assignable to type 'Hero.'. It also mentions 'Property 'telefone' is missing in type '{ nome: string; vulgo: string; }.' and 'Hero'.ts(2345)'. Below the tooltip, there are links for 'View Problem (Alt+F8)' and 'Quick Fix... (Ctrl+.)'.

15,2,1 - Instalando e Configurando o TS

segunda-feira, 7 de outubro de 2024 19:19

Para instalar dentro do projeto, se certifique que esta na pasta padrao do projeto

RODAR "npm init -y" PARA INICIAR O PROJETO NODE NA RAIZ

```
PS D:\conta.ACER-ANDRE-SANT\Ambiente de Desenvolvimento\DIOS - Bootcamps\Bootcamp_Springboot_Angular DEAL> cd ..  
PS D:\conta.ACER-ANDRE-SANT\Ambiente de Desenvolvimento\DIOS - Bootcamps\Bootcamp_Springboot_Angular DEAL> cd ..\Bootcamp_Springboot_Angular DEAL\  
PS D:\conta.ACER-ANDRE-SANT\Ambiente de Desenvolvimento\DIOS - Bootcamps\Bootcamp_Springboot_Angular DEAL\Bootcamp_Springboot_Angular DEAL> cd ..\Aulas-TS\  
PS D:\conta.ACER-ANDRE-SANT\Ambiente de Desenvolvimento\DIOS - Bootcamps\Bootcamp_Springboot_Angular DEAL\Bootcamp_Springboot_Angular DEAL\Aulas-TS> cd ..\Aula1-InstallAndTest  
> npm install typescript -D  
  
added 1 package in 3s  
PS D:\conta.ACER-ANDRE-SANT\Ambiente de Desenvolvimento\DIOS - Bootcamps\Bootcamp_Springboot_Angular DEAL\Bootcamp_Springboot_Angular DEAL\Aulas-TS\Aula1-InstallAndTest  
>
```

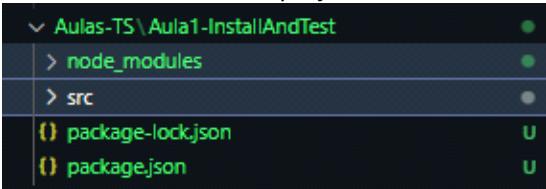
Assim que na pasta do projeto, rodar esse comando

npm install typescript -D

Esse -D insere uma dependencia de Dev/Debug

Isso impede de publicar ou fazer deploy do projeto, deixando somente em ambiente de teste

Esse comando criou um projeto TS



O SRC criei eu mas não sei se não criar ele cria sozinho

15,2,2 Iniciando com TS

segunda-feira, 7 de outubro de 2024 19:33

Note como é utilizado o index.ts

!!! Algumas dessas funções/construtores/type não existem em JS

!!! Se algumas chamadas não existem em JS, quando a chamada node src/index.ts for realizada
O código irá compilar o que ele entender e retornar erro no que não entender, vai crashar

Para isso, precisamos antes de mais nada "TRADUZIR O CÓDIGO PARA JS"

Utilizando do NPX - Node Package Execute como na print abaixo

The screenshot shows the VS Code interface with two tabs open: 'index.ts' and 'index.js'. The 'index.ts' tab contains the following TypeScript code:

```
src > index.ts > printaObjeto
1 type heroi = {
2   name: string;
3   vulgo: string;
4 };
5
6 function printaObjeto(pessoa: heroi) {
7   console.log(pessoa);
8 }
9
10 printaObjeto({
11   name: "bruce wayne",
12   vulgo: "batman".
```

The 'index.js' tab is currently inactive. Below the code editor is a terminal window showing the command line output:

```
PS C:\dio\angular\C2 - Typescript para Angular> npx tsc src/index.ts
PS C:\dio\angular\C2 - Typescript para Angular> npx tsc src/index.ts
PS C:\dio\angular\C2 - Typescript para Angular> 
```

No entanto quando traduzido, os dois estão na mesma classe SRC e dessa forma ele dá erro se identifica a mesma Function 2x ou mais repetidamente

Dá para entender que se rodarmos o TS o node não entende TS e trava o código,

Se rodarmos o JS ele entende e roda sem as travas do TS

Ou seja, o TS é um controle, uma segurança, um "gateway" para o JS mas no fim, o código traduzido e depois executado, será o JS

```
PS D:\conta.ACER-ANDRE-SANT\Ambiente de Desenvolvimento\DIOS - Bootcamps\Bootcamp_Springboot_Angular DEAL\Bootcamp_Springboot_Angular DEAL\Aulas-TS\Aula1-InstallAndTest
> npx tsc src/index.ts
PS D:\conta.ACER-ANDRE-SANT\Ambiente de Desenvolvimento\DIOS - Bootcamps\Bootcamp_Springboot_Angular DEAL\Bootcamp_Springboot_Angular DEAL\Aulas-TS\Aula1-InstallAndTest
> node src/index.ts
D:\conta.ACER-ANDRE-SANT\Ambiente de Desenvolvimento\DIOS - Bootcamps\Bootcamp_Springboot_Angular DEAL\Bootcamp_Springboot_Angular DEAL\Aulas-TS\Aula1-InstallAndTest\sr
c\index.ts:1
type Heroi = {
    ^^^^^^

SyntaxError: Unexpected identifier 'Heroi'
    at wrapSafe (node:internal/modules/cjs/loader:1378:20)
    at Module._compile (node:internal/modules/cjs/loader:1428:41)
    at Module._extensions..js (node:internal/modules/cjs/loader:1548:10)
    at Module.load (node:internal/modules/cjs/loader:1288:32)
    at Module._load (node:internal/modules/cjs/loader:1104:12)
    at Function.executeUserEntryPoint [as runMain] (node:internal/modules/run_main:174:12)
    at node:internal/main/run_main_module:28:49

Node.js v20.17.0
PS D:\conta.ACER-ANDRE-SANT\Ambiente de Desenvolvimento\DIOS - Bootcamps\Bootcamp_Springboot_Angular DEAL\Bootcamp_Springboot_Angular DEAL\Aulas-TS\Aula1-InstallAndTest
> node src/index.js
{ nome: 'Stranger', vulgo: 'Doctor' }
PS D:\conta.ACER-ANDRE-SANT\Ambiente de Desenvolvimento\DIOS - Bootcamps\Bootcamp_Springboot_Angular DEAL\Bootcamp_Springboot_Angular DEAL\Aulas-TS\Aula1-InstallAndTest
> 
```

PARA ISSO, AGORA NECESSARIO MEXER NAS CONFIGURAÇÕES DO NODE

15,2,3 Configurando TS e Node -> npx tsc --init | ts.config

segunda-feira, 7 de outubro de 2024 19:38

PRECISAMOS RESOLVER 2 CHATICES DO CODIGO

1. O FATO DELE CRIAR SEMPRE O ARQUIVO EM JS DO TS TRADUZIDO
2. O FATO DE TER QUE SEMPRE TRADUZIR UM POR UM DOS ARQUIVOS

!!!!!! PARA ESSAS CONFIGURAÇÕES, NECESSARIO UM ARQUIVO CONFIG !!!!!!!

Para iniciar esse arquivo de configuração

Necessario jogar o comando do NPX | NPX TSC --INIT

```
PS D:\conta.ACER-ANDRE-SANT\Ambiente de Desenvolvimento\DIOS - Bootcamps\Bootcamp_Springboot_Angular DEAL\Bootcamp_Springboot_Angular DEAL\Aulas-TS\Aula1-InstallAndTest
> npx 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
PS D:\conta.ACER-ANDRE-SANT\Ambiente de Desenvolvimento\DIOS - Bootcamps\Bootcamp_Springboot_Angular DEAL\Bootcamp_Springboot_Angular DEAL\Aulas-TS\Aula1-InstallAndTest
>
```

Dessa forma conseguimos gerenciar elementos e configurações

Para saber mais o que habilitar ou desabilitar, verificar em TS CONFIG dentro do site do Typescript

The screenshot shows two side-by-side browser windows. The left window is the 'TypeScript Playground' at <https://www.typescriptlang.org/>, showing a code editor with a sample TypeScript file containing a 'Hello World' console log. The right window is the 'TSConfig Reference' page, which provides detailed documentation for the 'files' section of a tsconfig.json file, listing various compiler options and their descriptions.

Quando gerado esse --init

Ele gera um config, com ele conseguimos ver as diversas configurações que podemos realizar

```

2   "compilerOptions": {
3     /* Projects */
4     // "incremental": true,           /* Save .tsbuildinfo files to allow for incremental compilation of projects. */
5     // "composite": true,            /* Enable constraints that allow a TypeScript project to be used with projects. */
6     // "tsBuildInfoFile": "./tsbuildinfo", /* Specify the path to .tsbuildinfo incremental compilation file. */
7     // "disableSourceOfProjectReferenceRedirect": true, /* Disable preferring source files instead of declaration files when referencing projects. */
8     // "disableSolutionSearching": true, /* Opt a project out of multi-project reference checking when editing. */
9     // "disableReferencedProjectLoad": true, /* Reduce the number of projects loaded automatically by TypeScript. */
10
11    /* Language and Environment */
12    "target": "es2016",             /* Set the JavaScript language version for emitted JavaScript and include features in the 'lib' option. */
13    // "lib": [],                  /* Specify a set of bundled library declaration files that describe the target environment. */
14    // "jsx": "preserve",          /* Specify what JSX code is generated. */
15    // "experimentalDecorators": true, /* Enable experimental support for legacy experimental decorators. */
16    // "emitDecoratorMetadata": true, /* Emit design-type metadata for decorated declarations in source files. */
17    // "jsxFactory": "",           /* Specify the JSX factory function used when targeting React JSX emit, e.g. 'React.createElement'. */
18    // "jsxFragmentFactory": "",    /* Specify the JSX Fragment reference used for fragments when targeting React JSX emit. */
19    // "jsxImportSource": "",       /* Specify module specifier used to import the JSX factory functions when targeting React JSX emit. */
20    // "reactNamespace": "",        /* Specify the object invoked for 'createElement'. This only applies when targeting React JSX emit. */
21    // "noLib": true,               /* Disable including any library files, including the default lib.d.ts. */
22    // "useDefineForClassFields": true, /* Emit ECMAScript-standard-compliant class fields. */
23    // "moduleDetection": "auto",   /* Control what method is used to detect module-format JS files. */
24
25    /* Modules */
26    "module": "commonjs",          /* Specify what module code is generated. */
27    "rootDir": "./src",            /* Specify the root folder within your source files. */
28
29    /* moduleResolution */
30    // "moduleResolution": "node10", /* Specify how TypeScript looks up a file from a given module specifier. */
31    // "baseUrl": "./",            /* Specify the base directory to resolve non-relative module names. */
32    // "paths": {},                /* Specify a set of entries that re-map imports to additional lookup locations. */
33    // "rootDirs": [],              /* Allow multiple folders to be treated as one when resolving modules. */
34    // "typeRoots": [],             /* Specify multiple folders that act like './node_modules/@types'. */
35    // "types": [],                 /* Specify type package names to be included without being referenced in a module. */
36    // "allowUmdGlobalAccess": true, /* Allow accessing UMD globals from modules. */
37    // "moduleSuffixes": [],         /* List of file name suffixes to search when resolving a module. */
38    // "allowImportingTsExtensions": true, /* Allow imports to include TypeScript file extensions. Requires '--moduleResolution node10'. */
39    // "resolvePackageJsonExports": true, /* Use the package.json 'exports' field when resolving package imports. */
40    // "resolvePackageJsonImports": true, /* Use the package.json 'imports' field when resolving imports. */

```

```

as-TS > Aula1-InstallAndTest > tsconfig.json > compilerOptions > outDir
2   "compilerOptions": {
3     // "declarationMap": true,           /* Create sourcemap files for emitted JavaScript files. */
4     // "emitDeclarationOnly": true,      /* Only output d.ts files and not JavaScript files. */
5     // "sourceMap": true,               /* Create source map files for emitted JavaScript files. */
6     // "inlineSourceMap": true,          /* Include sourcemap files inside the emitted JavaScript files. */
7     // "noEmit": true,                 /* Disable emitting files from a compilation. */
8     // "outFile": "./",                /* Specify a file that bundles all outputs into one file. */
9     // "outDir": "./build",             /* Specify an output folder for all emitted files. */
10    // "removeComments": true,          /* Disable emitting comments. */
11    // "importHelpers": true,           /* Allow importing helper functions from tslib once they are emitted to the 'build' folder. */

```

VEJA QUE TANTO ESSE ROOTDIR E ESSE OUTDIR, REMOVI DOS COMENTARIOS

ROOT DIR

Dessa forma, quando npx tsc src... Se eu não especificar o caminho, por padrão ele vai traduzir tudo de src

OUT DIR

O que é traduzido irá para a pasta /build, dessa forma não ficará mais junto como em SRC/INDEX.TS

Dessa forma, agora quando somente npx tsc...

Dessa forma em Build/index.js...

The screenshot shows the file structure of the project:

- Root folder: Aulas-Java
- Subfolders: Aulas-JS, caseComJS.js, caseComTS.ts, Aulas-TS\Aula1-InstallAndTest, build, src, index.js, package-lock.json, package.json, tsconfig.json, Desafios-Projetos, Bootcamp Deal - Spring Boot e Angular.docx, README.md.
- Terminal output:

```

PS D:\conta.ACER-ANDRE-SANT\Ambiente de Desenvolvimento\DIOS - Bootcamps\Bootcamp_Springboot_Angular DEAL\Bootcamp_Springboot_Angular DEAL\Aulas-TS\Aula1-InstallAndTest
> npx tsc
PS D:\conta.ACER-ANDRE-SANT\Ambiente de Desenvolvimento\DIOS - Bootcamps\Bootcamp_Springboot_Angular DEAL\Bootcamp_Springboot_Angular DEAL\Aulas-TS\Aula1-InstallAndTest
> node build/index.js
{ name: 'Stranger', vulgo: 'Doctor' }
PS D:\conta.ACER-ANDRE-SANT\Ambiente de Desenvolvimento\DIOS - Bootcamps\Bootcamp_Springboot_Angular DEAL\Bootcamp_Springboot_Angular DEAL\Aulas-TS\Aula1-InstallAndTest
>

```

Ele traduziu para Build, futuramente consequentemente rodando node para JS, ele executou normalmente

Se quisermos fazer isso como em start.io

Em PLAYGROUND já deixa tudo preparado da tambem, mas é mais informativo para o futuro, da primeira forma

The screenshot shows the TypeScript Playground configuration interface. At the top, there's a navigation bar with links for Download, Docs, Handbook, Community, Playground, and Tools. Below the navigation bar, the main area is titled "Playground". It contains a list of configuration options with checkboxes:

- pretty**: Disable wiping the console in watch mode.
- removeComments**: Generate .d.ts files from TypeScript and JavaScript files in your project.
- noErrorTruncation**: Enable color and formatting in TypeScript's output to make compiler errors easier to read.
- importHelpers**: Disable emitting comments.
- downlevelIteration**: Allow importing helper functions from tslib once per project, instead of including them per-file.
- inlineSourceMap**: Emit more compliant, but verbose and less performant JavaScript for iteration.
- inlineSources**: Include sourcemap files inside the emitted JavaScript.
- stripInternal**: Include source code in the sourcemaps inside the emitted JavaScript.
- noEmitHelpers**: Disable emitting declarations that have `@internal` in their JSDoc comments.
- noEmitDecorators**: Disable generating custom helper functions like `get`, `set`, `reflect`, etc.

15,2,4 Configurando Node

segunda-feira, 7 de outubro de 2024 20:37

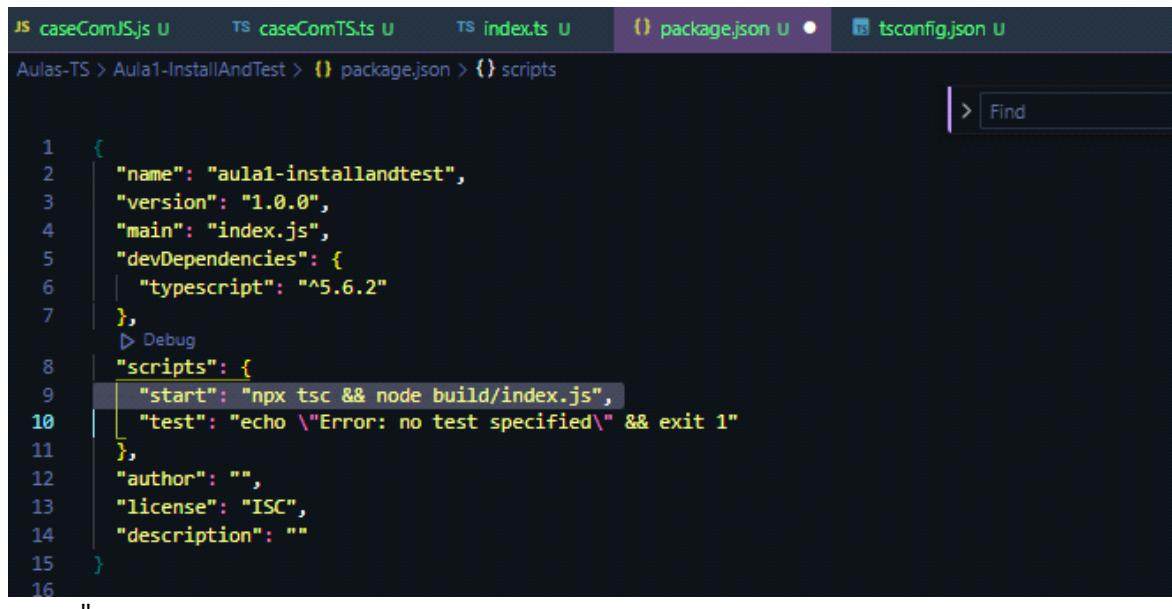
Para deixar ainda mais facil de rodar um projeto

De forma a traduzir o codigo TS e ainda rodar o que foi definido em Build/index.js

Necessario abrir o arquivo package.json (criado primeiro com o "**npm init -y**" para iniciar o projeto node eo "**npm install typescript -D**" para criar projeto TS)

Em SCRIPT, adicionar START e as config...

```
"start": "npx tsc && node build/index.js",
```



```
1  {
2    "name": "aula1-installandtest",
3    "version": "1.0.0",
4    "main": "index.js",
5    "devDependencies": {
6      "typescript": "^5.6.2"
7    },
8    "scripts": {
9      "start": "npx tsc && node build/index.js",
10     "test": "echo \\"$Error: no test specified\\" && exit 1"
11   },
12   "author": "",
13   "license": "ISC",
14   "description": ""
15 }
16 "
```

Dessa forma jogando um npm run start

Rodando o START, conseguimos iniciar o que esta dentro dele

```
PS D:\conta.ACER-ANDRE-SANT\Ambiente de Desenvolvimento\DI0 - Bootcamps\Bootcamp_Springboot_Angular
> npm run start

> aula1-installandtest@1.0.0 start
> npx tsc && node build/index.js
```

15,3 Tipos de Dados

terça-feira, 8 de outubro de 2024 19:20

Dentro do TS,
TIPOS PRIMITIVOS

Temos 3 tipos primitivos

| | | |
|---------|--------|--------|
| BOOLEAN | NUMBER | STRING |
|---------|--------|--------|

Com os tipos que definimos variaveis

Como declarar os tipos primitivos...

```
// Tipos primitivos
let ligado : false;
let nome : string = "Andre";
let idade : number = 23;
let altura : number = 1.9; /*Valores Decimais tambem*/
```

TIPOS NULL E INDEFINIDOS

Quando não queremos que o usuario altere ou por alguma razao ele deve permanecer nulo

```
// NULL / UNDEFINED
let nulo : null = null; /* Parametros que devem ser nulos */
let indefinido : undefined = undefined; /*Ninguem pode alterar esse tipo*/
```

TIPOS ABRANGENTES

Quando queremos que o retorno seja vazio ou quando não temos como saber como sera seu retorno

```
// TIPOS ABRANGENTES
let retorno : void; /* Nao aceita nada a nao ser o vazio */
let retornoView : any = false /*Recebe qualquer valor, como se fosse um VAR, um retorno imprevisivel*/
```

OBJETOS

SEM PREVISIBILIDADE

```
// OBJETOS _SEM PREVISIBILIDADE_
let produto : object = {
  name: "Andre",
  cidade: "SP",
  idade: 30,
}; /* Nao é ideal pois nao tem PREVISIBILIDADE */
```

COM PREVISIBILIDADE

Objetos normais como em outras linguagens, podendo instanciar mais de um objeto de maneira padronizada

```
// OBJETOS _COM PREVISIBILDADE_
type ProdutoLoja = {
  nome:string,
  preco:number,
  unidades:number,
};

// Instanciando um objeto
let meuProduto : ProdutoLoja = {
  nome: "Tenis",
  preco: 89.90,
  unidades: 89,
```

15,2,5 Sugestões do ChatGPT para o config do TS

terça-feira, 8 de outubro de 2024 19:45

A sua configuração de tsconfig.json está bem organizada e cobre aspectos importantes como a compatibilidade de módulos, saída dos arquivos compilados e verificação rigorosa de tipos. Aqui vão algumas sugestões que podem ser úteis, dependendo das suas necessidades:

1. **"moduleResolution": "node"**: Se você estiver trabalhando com Node.js ou bibliotecas que seguem a resolução de módulos do Node.js, isso pode ajudar a evitar problemas de resolução de dependências.

```
json
Copiar código
"moduleResolution": "node"
```

2. **"resolveJsonModule": true**: Se você precisar importar arquivos JSON diretamente no TypeScript, essa opção pode ser útil.

```
json
Copiar código
"resolveJsonModule": true
```

3. **"declaration": true**: Gera arquivos .d.ts, úteis para quem vai publicar pacotes ou se outras pessoas ou sistemas vão usar o código.

```
json
Copiar código
"declaration": true
```

4. **"sourceMap": true**: Se você precisar debugar o código JavaScript gerado, isso cria mapas de fonte que ajudam a fazer referência ao TypeScript original.

```
json
Copiar código
"sourceMap": true
```

5. **"removeComments": true**: Se você quiser que os comentários não apareçam no código JavaScript compilado, pode ativar isso.

```
json
Copiar código
"removeComments": true
```

6. **"strictNullChecks": true**: A verificação estrita de null e undefined evita muitos erros de execução, então, se estiver lidando com muitos valores opcionais, pode ser uma boa prática.

```
json
Copiar código
"strictNullChecks": true
```

Essas opções oferecem um equilíbrio entre otimizações e verificações rigorosas que podem ajudar a melhorar a qualidade do código.

15,4 TS Arrays

terça-feira, 8 de outubro de 2024 20:39

Quando queremos criar um Array

Trabalhando com Array Unidimensional

Podemos fazer de 2 formas, de forma clara ou facilitada

Podemos tambem trabalhar com arrays com mais de um dado, como definido abaixo,
TIPO STRING E NUMBER (Number independente de Decimal ou Inteiro)

```
// ARRAY / VETOR
let dados:string[] = ["andre", "felipe", "jhonathan", "isabele"];      /* TIPO SIMPLIFICADO */
let dados2: Array<string> = ["andre", "felipe", "jhonathan", "isabele"];/* TIPO DE DETALHAMENTO ARRAY */

// Array com mais de um tipo de elementos
let infos: (string | number)[] = ["andre", 23, "joao", 16]
```

NO ENTANTO ELE NÃO VAI JUNTAR O 1º COM O 2,
TANTO QUE ELES NEM PRECISAM ESTAR EM ORDEM

```
// Array com mais de um tipo de elementos
let infos: (string | number)[] = ["andre", 23, "joao", 16.9, "andre", "andre"];
|
```

15,4,1 TS Tuplas

terça-feira, 8 de outubro de 2024 20:49

Vetor com multi-elementos

Com já visto no anterior, TUPLAS são feitas da seguinte forma

```
// TUPLAS / ARRAY COM MULTIPLOS ELEMENTOS
let boleto : [string, number, number] = [
  "agua conta",
  199.90,
  3214565,
]; /** DEVE SER FEITO NA ORDEM COMO INFORMADO NO "CONSTRUTOR" */
```

A diferença de um VETOR COM MULTIPOS ELEMENTOS,

A Tupla respeita a forma que foi informado, sendo que se esquecer um "" ou informar um tipo "certo" mas na ordem errada, ele retorna um erro

```
// TUPLAS / ARRAY COM MULTIPLOS ELEMENTOS
let boleto : [string, number, number] = [
  199.90,
  "agua conta",
  3214565,
]; /** DEVE SER FEITO NA ORDEM COMO INFORMADO NO "CONSTRUTOR" */
```

SE ESQUECER UM ELEMENTO OU DEFINIR UM TIPO ERRADO É A MESMA COISA

15,5,2 Manipulação de Arrays em TS/JS

terça-feira, 8 de outubro de 2024 21:29

Dentro dos ARRAYS, temos métodos que conseguimos manipular os Arrays

The screenshot shows a code editor with the following TypeScript code:

```
let dados:string[] = ["andre", "felipe", "jhonathan", "isabele"]
let dados2: Array<string> = ["andre", "felipe", "jhonathan", "i"]

// Array com mais de um tipo de elementos
let infos: (string | number)[] = ["andre", 23, "joao", 16.9, "a"]

// TUPLAS / ARRAY COM MULTIPLOS ELEMENTOS
let boleto : [string, number, number] = [
  "agua conta",
  199.90,
  3214565,
]; /** DEVE SER FEITO NA ORDEM COMO INFORMADO NO "CONSTRUTOR" */

dados.m
```

A tooltip for the `.map` method is displayed on the right side of the screen:

`(method) Array<string>.map<U>(callbackfn: (value: string, index: number, arra`
`y: string[]) => U, thisArg?: any): U[]`

Calls a defined callback function on each element of an array, and returns an array that contains the results.

`@param callbackfn — A function that accepts up to three arguments. The map method calls the callbackfn function one time for each element in the array.`

`@param thisArg — An object to which the this keyword can refer in the callbackfn function. If thisArg is omitted, undefined is used as the this value.`

Comandos mais comuns para mapear elementos/ filtrá-los, remove-los... Enfim, gerenciar os elementos

.map -- mapeia objetos
.reduce
.pop -- remove o último elemento
.filter -- filtra algo específico

15,5 Trabalhando com Dates

terça-feira, 8 de outubro de 2024 21:37

Para organizar datas, precisamos utilizar da biblioteca e filtrando do padrão americano

```
51 let aniversario : Date;  
52 let aniversario1 : Date;  
53  
54 aniversario = new Date("2022-01-30 05:00");  
55 console.log (aniversario.toString());  
56  
57 aniversario1 = new Date("2022-01-30");  
58 console.log (aniversario1.toString());
```

The screenshot shows a terminal window with the following content:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS  
> npx tsc && node build/index.js  
[Function: toString]  
PS D:\conta.ACER-ANDRE-SANT\Ambiente de Desenvolvimento\DIOS - Boot  
  
> aula2-tiposprimitivos@1.0.0 start  
> npx tsc && node build/index.js  
  
Sun Jan 30 2022 05:00:00 GMT-0300 (Horário Padrão de Brasília)  
PS D:\conta.ACER-ANDRE-SANT\Ambiente de Desenvolvimento\DIOS - Boot  
  
> aula2-tiposprimitivos@1.0.0 start  
> npx tsc && node build/index.js  
  
Sun Jan 30 2022 05:00:00 GMT-0300 (Horário Padrão de Brasília)  
Sat Jan 29 2022 21:00:00 GMT-0300 (Horário Padrão de Brasília)
```

Como bem observado, melhor definir com a hora, senão necessário uma configuração futura

15,6 Funções

terça-feira, 8 de outubro de 2024 22:11

Assim como no JS, para criar uma Função, podemos simplesmente criar uma FUNCTION

Temos duas maneiras dai de manipula-las,
visando futuramente,
de maneira explicita é melhor

```
60 console.log("FUNÇÕES");
61 function addNumber(x : number, y : number) : number /*D
62     return x + y;
63 }
64
65 let soma : number = addNumber(4,7.02);
66
67 function addToHello(name:string)/*NAO ESTA DECLARADO, M
68     return `Hello ${name}`;
69 }
70
71 console.log(addToHello("Andre"));
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS D:\conta.ACER-ANDRE-SANT\Ambiente de Desenvolvimento\DI0 - Boot
> aula2-tiposprimitivos@1.0.0 start
> npx tsc && node build/index.js

Sun Jan 30 2022 05:00:00 GMT-0300 (Horário Padrão de Brasília)
Sat Jan 29 2022 21:00:00 GMT-0300 (Horário Padrão de Brasília)
FUNÇÕES
Hello Andre
PS D:\conta.ACER-ANDRE-SANT\Ambiente de Desenvolvimento\DI0 - Boot
```

15,6,1 Funções Multi-Tipos

terça-feira, 8 de outubro de 2024 22:22

Quando caso em uma aplicação ou manipulação do Banco de Dados

Se definimos que ele deve retornar em um tipo e ele retorna em outro tipo

Ou definimos que ele é nulo, mas ele retorna String ou outro,

Para evitar isso

É sempre bom definir quais os tipos que ele pode retornar

```
console.log(addHello("Andre"));
💡
function CallToPhone (phone:number | string): number | string /*Aceita como String ou numero*/{
  return phone;
}
console.log(CallToPhone("41995368352"))
```

15,6,2 Funções Assíncronas

terça-feira, 8 de outubro de 2024 22:32

São funções que entra um tipo de dados

E queremos que retorne outro tipo de dados ou um conjunto de tipos, (Seja String ou Number)

Async function... :

LEMBRE-SE SEMPRE QUE DEPOIS DO : É O QUE ELE RECEBE

Em Java é feito no começo (*PUBLIC VOID / PUBLIC INTEGER ...*)

Em JS é no final da Function

```
async function getDatabase(id:number): Promise <string> {
    return "Andre";
};
```

15,7 TS Interfaces Single or Multityple

quarta-feira, 9 de outubro de 2024 18:33

Interfaces são parecidas com TYPE

TYPE fazem produtos, objetos agregados de varias propriedades

```
type robot = {
  id: number;
  name: string;
};

const bot: robot = [
  id: 1,
  name: "megaman",
];
```

Mais utilizado quando queremos tipar algo

INTERFACES, parecido com TYPE, mas são bastante usados quando trabalhando com classes

```
// criando uma tipagem === um modelo de objeto
interface robot{
  id: number;
  name: string;
};

const bot: robot = [
  id: 1,
  name: "bomberman",
];
```

Da para relacionar como um contrato compra

Se vc quer utilizar daquilo, deve seguir as clausulas desse contrato

Dessa forma, se utiliza da Interface, necessario utilizar das suas variaveis

Podemos fazer essa interface aceitar mutiiplos tipos em uma variavel

Podemos tambem fazer um elemento ser somente leitura

```
interface robot2{
  /*Torna esse campo somente LEITURA>>> */
  readonly id: number | string;
  /*<<< Ele pode aceitar tanto NUMERO OU STRING*/
  name: string;
};

const ironman: robot2 = [
  id: "2",
  name: "iron man"
];
```

Com esse modo READ ONLY

Ele não deixa alterar a variavel

Perceba, que no BOT ele permitira a alteração do ID mas em IRONMAN que foi definido como READONLY ele impedira essa alteração

```
interface robot{
  name: string;
};

const bot: robot = {
  id: 1,
  name: "bomberman",
};

interface robot2{
  /*Torna esse campo somente LEITURA>>> */
  readonly id: number | string;
  /*<<< Ele pode aceitar tanto NUMERO OU STRING*/
  name: string;
};

const ironman: robot2 = {
  id: "2",
  name: "iron man"
};

console.log(ironman.id=1013);
console.log(bot.id=1013);
```

Cannot assign to 'id' because it is a read-only property. ts(2540)
(property) robot2.id: any
View Problem (Alt+F8) No quick fixes available

15,8 Type or Interface

quinta-feira, 10 de outubro de 2024 20:56

INTERFACE

Quando estivermos mexendo com classes

Utilizamos A INTERFACE

DE FORMA QUE IMPLEMENTS

```
5 }; Class 'Pessoa' incorrectly implements interface 'robot2'.
6   Type 'Pessoa' is missing the following properties from type 'robot2': id, name ts(2420)
7 console class Pessoa
8 console
9 View Problem (Alt+F8) Quick Fix... (Ctrl+.)
0 class Pessoa implements robot2/* SE NAO IMPLEMENTADO AS COISAS DA INTERFACE, VAI DAR ERRO */
1 [
2 ];
```

UMA INTERFACE DENTRO DE UM OBJETO DEVE SER ASSIM

```
class Pessoa implements robot2/* SE NAO IMPLEMENTADO AS COISAS DA INTERFACE, VAI DAR ERRO */
  id: string | number;
  name: string;

  constructor(id:string|number, name:string){
    this.id = id,
    this.name = name,
  }

const pessoa = new Pessoa(1,"Andre")
```

Como implementar uma pessoa com a interface robot em Pessoa

const pessoa = new Pessoa(1,"Andre");

TYPE

Quando queremos somente tipar um objeto

Definir a forma e as constantes de variavel

Utilizamos O TYPE

15,2,2 Comandos Iniciais para programação com TS no Node

sexta-feira, 11 de outubro de 2024 20:48

INICIALMENTE, PRECISAMOS INICIAR O PROJETO NODE Dentro do Diretório do Projeto

"npm init -y"

DEPOIS PRECISAMOS INSTALAR O TYPESCRIPT EM MODO DE DESENVOLVEDOR PARA NÃO FICAR JOGANDO ATUALIZAÇÕES EM AMBIENTE PRODUÇÃO

"npm install typescript -D"

EM SEGUIDA ABRIR UM ARQUIVO CONFIG TS NO PROJETO

"npx tsc --init"

| | | | |
|-----|-------------------|--------|-----|
| --- | PREPARAR O CONFIG | 15,2,3 | --- |
|-----|-------------------|--------|-----|

DENTRO DA CLASSE "package.json" INSERIR EM "script"

"start": "npx tsc && node build/index.js",

Em seguida, dentro do diretório do projeto

npm start

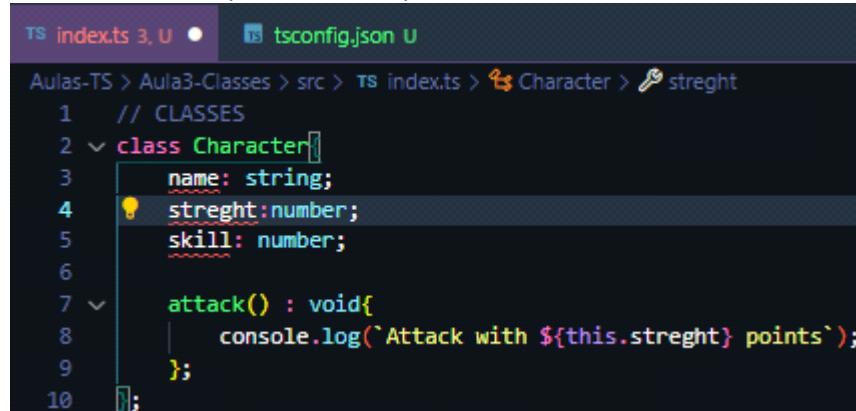
15,9 Classes em TS

sexta-feira, 11 de outubro de 2024 21:30

Imagine que temos uma character/personagem

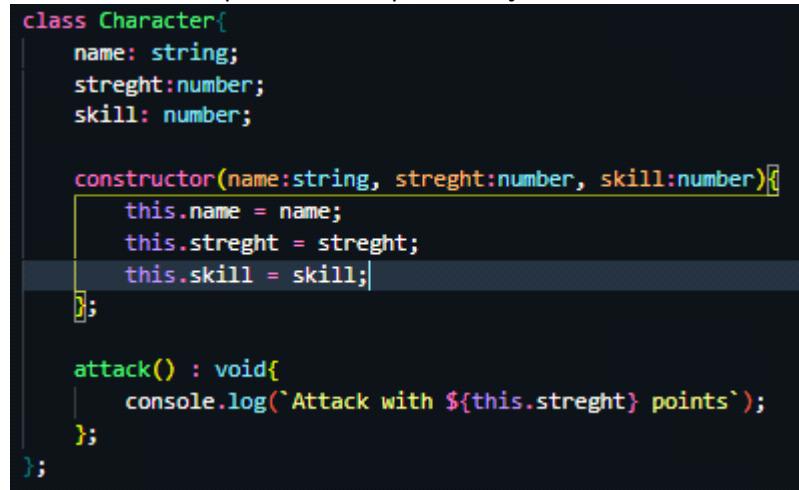
Todos tem o mesmo padrao

Declaramos da seguinte forma a prmeiro momento



```
1 // CLASSES
2 class Character{
3     name: string;
4     stregh: number;
5     skill: number;
6
7     attack(): void{
8         console.log(`Attack with ${this.stregh} points`);
9     };
10}
```

Ele esta com erro pois falta a implementação do Construtor



```
class Character{
    name: string;
    stregh: number;
    skill: number;

    constructor(name:string, stregh:number, skill:number){
        this.name = name;
        this.stregh = stregh;
        this.skill = skill;
    }

    attack(): void{
        console.log(`Attack with ${this.stregh} points`);
    };
}
```

```
Character { name: 'Shark', stregh: 10, skill: 98 }
Attack with 10 points
```

OBSERVAÇÃO DO JS e TS

Uma coisa legal é que se vc quer dizer que um método é necessário ou não

Pode simplesmente colocar um "?" na frente dele

Dessa forma, não precisa nem colocá-lo em um Construtor

Pois afinal ele não dará essa obrigatoriedade de chamá-lo

Outra coisa, já mencionado antes mas vale a pena ressaltar

É o READONLY onde conseguimos fazer um CONSOLE.LOG mas não conseguimos setar esse valor quando já em produção pois é Somente Leitura

```
class Character{
    /**ATRIBUTOS */
    private name?: string;
    private readonly streght:number;
    private skill: number;

    constructor(/*name:string,*/ streght:number, skill:number){
        /*this.name = name; // nome*/
        this.streight = streght; // força
        this.skill = skill; // habilidade
    };

    /**METODO */
    attack() : void{
        console.log(`Attack with ${this.streight} points`);
    };
};

const person1 = new Character(/*"Shark",*/ 10, 98);

console.log(person1);
person1.attack();
```

15,9,1 Modificadores de acesso

sexta-feira, 11 de outubro de 2024 21:47

| | | |
|---------|--------|-----------|
| PRIVATE | PUBLIC | PROTECTED |
|---------|--------|-----------|

Esses tipos de modificadores falam

| | |
|--|-----------|
| se quer que a classe se comunique com o projeto todo | PUBLIC |
| Se vai se comunicar com somente os seus "filhos" | PROTECTED |
| Ou somente naquela classe | PRIVATE |

Ou seja, refatorando o código 15,9

Vamos utilizar o método PRIVATE NOS ATRIBUTOS DA CLASSE

```
class Character{
    /**ATRIBUTOS */
    private name?: string;
    private readonly stregh: number;
    private skill: number;

    constructor(/*name:string,*/ stregh:number, skill:number){
        /*this.name = name; // nome*/
        this.stregh = stregh; // força
        this.skill = skill; // habilidade
    };
}
```

15,9,2 Declarando Subclasses (HERANÇA)

sexta-feira, 11 de outubro de 2024 22:05

Temos alguns personagens

Uns Magos,
Outros Soldados

Dessa forma, como eles são todos personagens,
mais contem subcategorias diferentes

Em vez de colocar tudo em todos

Podemos chamar a SUPER e atribuir mais elementos

```
// SUBCLASS
class Magician extends Character{
    magicPoints: number;

    constructor(name:string, streght:number, skill:number, magicPoints: number){
        super(name, streght, skill); /*Chamando a classe PAI / SUPERCLASS */
        this.magicPoints = magicPoints;
    }

};

const person1 = new Character("Shark", 10, 98);
const person2 = new Magician("Mago", 9,30,100);

console.log(person1);
person1.attack();

console.log(person2);
person2.attack();
```

15,9,5 Trabalhando com Generics

sábado, 12 de outubro de 2024 21:14

Quando estamos lidando com o generico de algo

Quando vamos a farmacia e não tem um determinado remedio/pomada

Pedimos entao o generico daquela marca e conseguimos ter uma com a mesma finalidade

Na programação é a mesma coisa

Lembrando de JAVA onde as Generics são as "tags" <> onde dentro delas definimos o titulo, em TS não é muito diferente

NOTA!!!: AS RETICENCIAS EM TS significa que

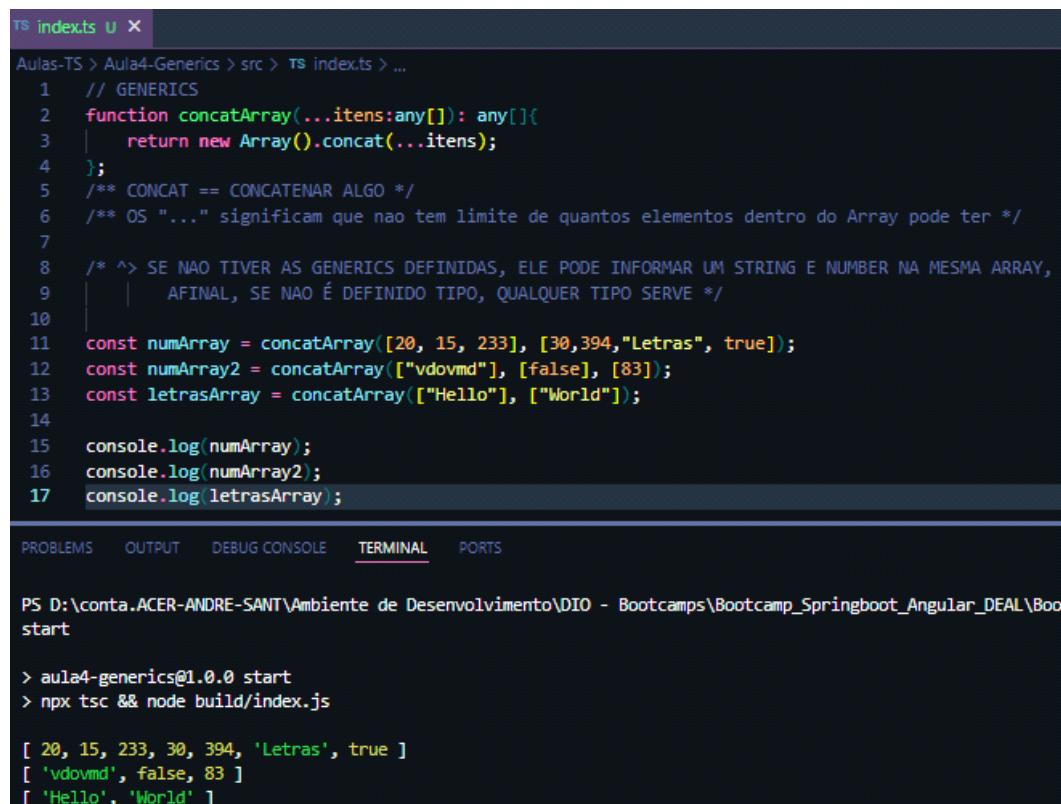
Senao tiver o GENERICS, FICA DESSA FORMA

```
// GENERICS
function concatArray(...itens:any[]): any[]{
  return new Array().concat(...itens);
}
/** CONCAT == CONCATENAR ALGO */
/** OS "..." significam que nao tem limite de quantos elementos dentro do Array pode ter */

/* ^> SE NAO TIVER AS GENERICS DEFINIDAS, ELE PODE INFORMAR UM STRING E NUMBER NA MESMA ARRAY,
   |   |   AFINAL, SE NAO É DEFINIDO TIPO, QUALQUER TIPO SERVE */
```

^> SE NAO TIVER AS GENERICS DEFINIDAS, ELE PODE INFORMAR UM STRING E NUMBER NA MESMA ARRAY,
AFINAL, SE NAO É DEFINIDO TIPO, QUALQUER TIPO SERVE

AFINAL, SE NAO É DEFINIDO TIPO, QUALQUER TIPO SERVE



```
ts index.ts ✘
Aulas-TS > Aula4-Generics > src > ts index.ts > ...
1 // GENERICS
2 function concatArray(...itens:any[]): any[]{
3   return new Array().concat(...itens);
4 }
5 /** CONCAT == CONCATENAR ALGO */
6 /** OS "..." significam que nao tem limite de quantos elementos dentro do Array pode ter */
7
8 /* ^> SE NAO TIVER AS GENERICS DEFINIDAS, ELE PODE INFORMAR UM STRING E NUMBER NA MESMA ARRAY,
9   |   |   AFINAL, SE NAO É DEFINIDO TIPO, QUALQUER TIPO SERVE */
10
11 const numArray = concatArray([20, 15, 233], [30,394,"Letras", true]);
12 const numArray2 = concatArray(["vдовмд"], [false], [83]);
13 const letrasArray = concatArray(["Hello"], ["World"]);
14
15 console.log(numArray);
16 console.log(numArray2);
17 console.log(letrasArray);

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\conta.ACER-ANDRE-SANT\Ambiente de Desenvolvimento\DI0 - Bootcamps\Bootcamp_Springboot_Angular DEAL\Boo
start

> aula4-generics@1.0.0 start
> npx tsc && node build/index.js

[ 20, 15, 233, 30, 394, 'Letras', true ]
[ 'vдовмд', false, 83 ]
[ 'Hello', 'World' ]
```

QUANDO APPLICADO O GENERICS, PARA DEFINIR O TIPO

Fica dessa forma

```
11 | function concatArray<T>(...itens:T[]): T[]{
12 |   return new Array().concat(...itens);
13 | };
14 |
15 |
16 | const numArray = concatArray <number[]> ([20, 15, 233], [30,394,/*"Letras", true*/]);
17 | const numArray2 = concatArray <number[]> ([/*["vдовмд"], [false],*/ [83]]);
18 | const letrasArray = concatArray <string[]> (["Hello"], ["World"]);
19 |
20 | console.log(numArray);
21 | console.log(numArray2);
22 | console.log(letrasArray);
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS D:\conta.ACER-ANDRE-SANT\Ambiente de Desenvolvimento\DIOS - Bootcamps\Bootcamp_Springboot_Angular
start

> aula4-generics@1.0.0 start
> npx tsc && node build/index.js

[ 20, 15, 233, 30, 394 ]
[ 83 ]
[ 'Hello', 'World' ]
```

15,10 TS Facilitando ainda mais o RUN do TS

quinta-feira, 17 de outubro de 2024 19:58

Até agora nós vimos como fazer O modo de Runtime do typescript

Primeiro jogando comando de NPX TSC E depois o comando de node buscando a Bild/index.HTML

Mas existe um comando mais fácil do que isso ainda

Esse comando é basicamente um servidor interno gerado a cada vez que é salvo o documento
Ele gera um servidor que le o typescript

Mas para implementar ele no projeto, rodar o comando

"NPM INSTALL TS-NODE-DEV -D"

Lembrando que o -D Instala uma dependência de desenvolvedor não projeto

VISTO EM 15,2,2

```
  "scripts": {  
    "start": "npx tsc && node build/index.js",  
  }  
}
```

Depois de implementado esse comando, o próximo passo é criar um script dentro do start
Que seria dentro da configuração package.json

The screenshot shows a code editor with three tabs at the top: 'index.ts', 'tsconfig.json', and 'package.json'. The 'package.json' tab is active, displaying the following JSON configuration:

```
1  {
2    "name": "aula5-aprimorandorun",
3    "version": "1.0.0",
4    "main": "index.js",
5    "scripts": {
6      "start": "npx tsc && node build/index.html",
7      "start-dev": "ts-node-dev --respawn --transpile-only src/index.ts",
8      "test": "echo \"Error: no test specified\" && exit 1"
9    },
10   "keywords": [],
11   "author": "",
12   "license": "ISC",
13   "description": "",
14   "devDependencies": {
15     "typescript": "^5.6.3"
16   }
}
```

A cursor is visible on the line 'start-dev' in the 'scripts' section. Below the editor, a command-line interface shows the same command being typed:

```
"start-dev": "ts-node-dev --respawn --transpile-only src/index.ts"
```

Dessa forma, em vez de dar um START e de forma bunita fazer ele rodar normalzinho, obedecendo os comandos

E sempre DANDO RUN, DANDO RUN...

Ele não atualiza a não ser que restart a aplicação

```

ts index.ts  x  tsconfig.json  package.json
Aula5-AprimorandoRUN > src > ts index.ts > ...
1 let dado,dado2 : string;
2
3 dado = "Andre";
4 dado2 = "Felipe";
5
6 console.log(dado + dado2);
7
8 dado="Julian";
9 dado2="Romero";
10
11 console.log(dado + dado2);
12

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\conta.ACER-ANDRE-SANT\Ambiente de Desenvolvimento\DI> npm start

> aula5-aprimorandorun@1.0.0 start
> npx tsc && node build/index.js

AndreFelipe
PS D:\conta.ACER-ANDRE-SANT\Ambiente de Desenvolvimento\DI> npm start

> aula5-aprimorandorun@1.0.0 start
> npx tsc && node build/index.js

AndreFelipe
JulianRomero
PS D:\conta.ACER-ANDRE-SANT\Ambiente de Desenvolvimento\DI>

```

Podemos fazer dessa forma

Quando iniciado, ele permanece "online" vendo e como o comando diz
Gerando um RESPAW

```

index.ts  x  tsconfig.json  package.json
Aula5-AprimorandoRUN > src > index.ts > ...
1 let dado,dado2 : string;
2
3 dado = "Andre";
4 dado2 = "Felipe";
5
6 console.log(dado + dado2);
7
8 dado="Julian";
9 dado2="Romero";
10
11 console.log(dado + dado2);
12

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\conta.ACER-ANDRE-SANT\Ambiente de Desenvolvimento\DI> npm run start:dev

> aula5-aprimorandorun@1.0.0 start:dev
> ts-node-dev --respawn --transpile-only src/index.ts

[INFO] 20:47:45 ts-node-dev ver. 2.0.0 (using ts-node ver. 10.9.2, typescript ver. 5.6.3)
AndreFelipe

```

```
[INFO] 20:47:45 ts-node-dev ver. 2.0.0 (using ts-node ver. 10.9.2, typescript ver. 5.6.3)
AndreFelipe
[INFO] 20:47:53 Restarting: D:\conta.ACER-ANDRE-SANT\Ambiente de Desenvolvimento\DI0 - Boot
s-TS\Aula5-AprimorandoRUN\src\index.ts has been modified
AndreFelipe
JulianRomero
□
```

Depois de feito isso, para cancelar ou parar o projeto,

necessario um **CTRL C**

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

16 Intro Decorators

quinta-feira, 17 de outubro de 2024 21:03

Quando é decorado um código para ser reutilizado

Como a simulação vamos utilizar uma ordem da mãe

Quando você vai ao supermercado e a mãe sempre fala se você ver o leite a menos de 3 Reais você pode trazer porque é difícil dar esse valor

Sempre que você for na supermercado e ver o leite e ele estiver a menos de 3 BRL

você vai trazer pois foi um comando gatido que foi passado

Dá para levar em consideração que isso também é uma condicional certo?

Na programação esse decorators é basicamente a mesma coisa

O decorators é realizado em cima de uma function

Você define o que você quer que ela faça ou que tipo de ações vai ter quando ela for chamada

E ela É injetada quando tem um "@"

```
index.ts
Aula6-Decorators > src > index.ts > ...
1 function ExibirNome(target:any){
2 |   console.log(target);
3 };
4
5 class Funcionario{};
6
7 @ExibirNome
8 class Quincas{}; /* TIOZINHO DA PA */

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

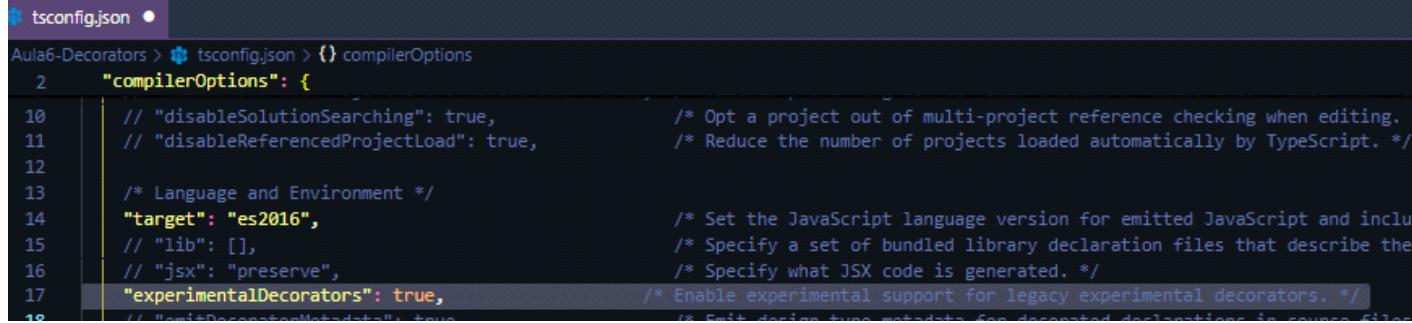
PS D:\conta.ACER-ANDRE-SANT\Ambiente de Desenvolvimento
└─ start
  └─ aula6-decorators@1.0.0 start
    └─ npx tsc && node build/index.js

[class Quincas]
PS D:\conta.ACER-ANDRE-SANT\Ambiente de Desenvolvimento
```

16,1 Implementando Decorators

quinta-feira, 17 de outubro de 2024 21:15

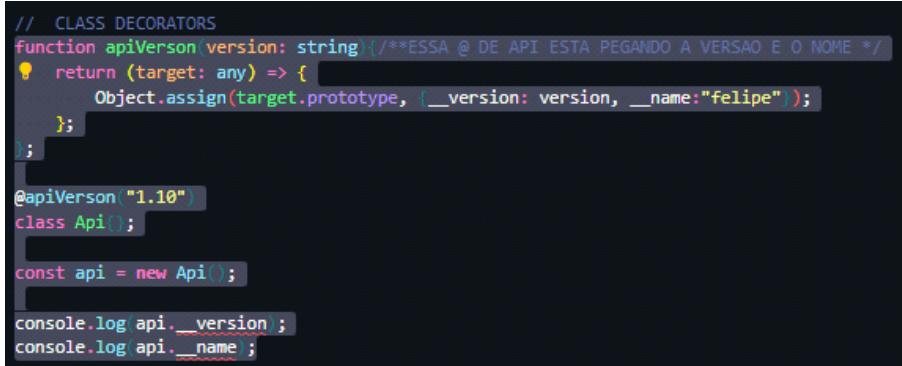
Para implementá-lo, necessário Ativa-lo dentro de CONFIG TS



```
tsconfig.json
Aula6-Decorators > tsconfig.json > {} compilerOptions
  2   "compilerOptions": {
  10  // "disableSolutionSearching": true,           /* Opt a project out of multi-project reference checking when editing. */
  11  // "disableReferencedProjectLoad": true,        /* Reduce the number of projects loaded automatically by TypeScript. */
  12
  13  /* Language and Environment */
  14  "target": "es2016",                          /* Set the JavaScript language version for emitted JavaScript and include files. */
  15  // "lib": [],                                /* Specify a set of bundled library declaration files that describe the runtime environment for emitted JavaScript files. */
  16  // "jsx": "preserve",                         /* Specify what JSX code is generated. */
  17  "experimentalDecorators": true,              /* Enable experimental support for legacy experimental decorators. */
  18  // "emitDecoratorMetadata": true,               /* Emit design-time metadata for decorated declarations in source files. */

  
```

Uma Implementação a nível de classe é feita da seguinte forma



```
// CLASS DECORATORS
function apiVersion(version: string){/**ESSA @ DE API ESTA PEGANDO A VERSAO E O NOME */}
  return (target: any) => {
    Object.assign(target.prototype, {__version: version, __name:"felipe"});
  };
}

@apiVersion("1.10")
class Api{};

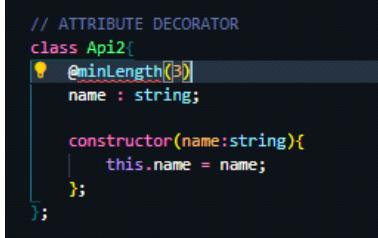
const api = new Api();

console.log(api.__version__);
console.log(api.__name__);
```

A implementação a nível de classe eu não entendi muito bem

No entanto ela superou quando vai tirar de uma forma sempre que a classe Chamar aquele método

Implementação feita através de atributo é feita da seguinte forma



```
// ATTRIBUTE DECORATOR
class Api2{
  @minLength(3)
  name : string;

  constructor(name:string){
    this.name = name;
  }
}
```

Esse atributo decorator em cima de um atributo Serve para determinar uma ação àquele atributo

Nesse caso como ele está dando erro
será necessário implementar A Decorator minLength

```

// ATTRIBUTE DECORATOR
function minLength(length:number){
    return (target:any, key: string)=>{
        let _value = target[key];
        // LEITURA
        const getter = ()=> _value;
        // SETAR VALOR SE...
        const setter = (value:string)=> {
            if (value.length< length){
                throw new Error(`Tamanho menor do que ${length}`);
            } else {
                _value = value;
            }
        };
        Object.defineProperty(target, key, {
            get: getter,
            set: setter,
        });
    };
}

class Api2{
    @minLength(3)
    name : string;

    constructor(name:string){
        this.name = name;
    };
}

```

Com esses métodos foi possível fazer uma adição de lógica ao adicionar algum elemento

```

62  const nome2 = new Api2("Andre");
63  console.log(nome2.name);
64

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\conta.ACER-ANDRE-SANT\Ambiente de Desenvolvimento\DI - Bootcamps\Bootcamp_Springboot_Angular
m run start:dev

> aula6-decorators@1.0.0 start:dev
> ts-node-dev --respawn --transpile-only src/index.ts

[INFO] 23:09:22 ts-node-dev ver. 2.0.0 (using ts-node ver. 10.9.2, typescript ver. 5.6.3)
[class Quincas]
1.10
felipe
Andre

```

Esse tipo de coisa por default não existe no TS mas com o @ foi possível

Dessa forma se adicione um elemento que não tem o tamanho maior que 3

Ele dara erro

```
// POR CONTA DO @ DEFINIDO, COMO O TAMANHO É MENOR QUE 3, DARA ERRO
const nome1 = new Api2("oi");
console.log(nome1.name);

// POR CONTA DO @ DEFINIDO, COMO O TAMANHO É MAIOR OU IGUAL QUE 3, DARA CERTO
const nome2 = new Api2("Andre");
console.log(nome2.name);

e

] 23:09:58 Restarting: D:\conta.ACER-ANDRE-SANT\Ambiente de Desenvolvimento\DI0 - Bootcamps\Bootcamp_Springboot_Angular DEAL\Bootcamp_Springboot_Angular
Aula6-Decorators\src\index.ts has been modified
s Quincas]

e
: Tamanho menor do que 3
t Api2.setter (D:\conta.ACER-ANDRE-SANT\Ambiente de Desenvolvimento\DI0 - Bootcamps\Bootcamp_Springboot_Angular DEAL\Bootcamp_Springboot_Angular\Au
corators\src\index.ts:35:23)
t new Api2 (D:\conta.ACER-ANDRE-SANT\Ambiente de Desenvolvimento\DI0 - Bootcamps\Bootcamp_Springboot_Angular DEAL\Bootcamp_Springboot_Angular\Aulas
ators\src\index.ts:53:18)
t Object.<anonymous> (D:\conta.ACER-ANDRE-SANT\Ambiente de Desenvolvimento\DI0 - Bootcamps\Bootcamp_Springboot_Angular DEAL\Bootcamp_Springboot_Angular\I
ula6-Decorators\src\index.ts:58:15)
t Module._compile (node:internal/modules/cjs/loader:1469:14)
t Module._compile (D:\conta.ACER-ANDRE-SANT\Ambiente de Desenvolvimento\DI0 - Bootcamps\Bootcamp_Springboot_Angular DEAL\Bootcamp_Springboot_Angular\DE
A6-Decorators\node_modules\source-map-support\source-map-support.js:568:25)
t Module.m._compile (C:\Users\CONTA-1.ACE\AppData\Local\Temp\ts-node-dev-hook-05652645221191155.js:69:33)
t Module._extensions..js (node:internal/modules/cjs/loader:1548:10)
t require.extensions..jsx.require.extensions..js (C:\Users\CONTA-1.ACE\AppData\Local\Temp\ts-node-dev-hook-05652645221191155.js:114:20)
t require.extensions.<computed> (C:\Users\CONTA-1.ACE\AppData\Local\Temp\ts-node-dev-hook-05652645221191155.js:71:20)
t Object.nodeDevHook [as .ts] (D:\conta.ACER-ANDRE-SANT\Ambiente de Desenvolvimento\DI0 - Bootcamps\Bootcamp_Springboot_Angular DEAL\Bootcamp_Springboot
ulas-TSAula6-Decorators\node_modules\ts-node-dev\lib\hook.js:63:13)
R] 23:09:58 Error: Tamanho menor do que 3
```

É difícil mas com a implementação e com muito estudo é possível a aplicação

Embora como uma API faz esse trabalho é meio desnecessário quando você faz projetos fullstack

17,1 Angular DEV

quinta-feira, 17 de outubro de 2024 21:22

Entrando no site da angular.dev

Conseguimos verificar algumas coisas bem maneiras

- como a documentação
- tutoriais de como utilizar algum laço de repetição ou algo do tipo
- e uma parte chamada playground onde nós conseguimos rodar alguns códigos
ver eles sendo executados na prática

TUTORIAIS

The screenshot shows the 'Tutorials' section of the angular.dev website. It features three cards:

- Learn Angular in your browser via the Playground**: Shows a browser window with a blank white page. [Start coding](#)
- Build your first Angular app locally via npm**: Shows a terminal window with a command being typed. [Start coding](#)
- Deferable views via the Playground**: Shows a browser window with a complex UI component. [Start coding](#)

17 ANGULAR JS - Intro

sexta-feira, 18 de outubro de 2024 18:39

Um framework auxilia na construção de um código

Tudo o que é feito ou facilitado dentro de um framework pode ser feito sem ele

Documentação / Versões / Instaladores

```
*** angular-versions.md ✘
1 # ANGULAR.JS (VERSÃO 1)
2
3 documentação: https://angularjs.org/
4
5 # ANGULAR (VERSÃO 2 a 16)
6
7 documentação: https://angular.io/
8
9 # ANGULAR VERSÃO (VERSÃO 17+)
10
11 documentação: https://angular.dev/
12
```

Angular 1 - foi a primeira versão onde não existia React, não existia outros frameworks

Angular 2-16 - É também uma versão do Angular, porém uma versão totalmente nova pois a primeira versão dependia do JavaScript

e a versão 11 até os dias de hoje
é feita através do TypeScript

Angular 17 - Na versão 17 é compatível com a versão de 2 a 16, no entanto foram adicionados mais algumas facilidades

Angular.dev

Atualmente algumas coisas como estrutura de repetição e loop podem ser feitas da seguinte maneira

```
@angular/core

@if Block

The @if block conditionally displays its content when its condition expression is truthy.

Description

The @if block conditionally displays its content when its condition expression is truthy.
```

The `@if` block conditionally displays its content when its condition expression is truthy.

Syntax

```
@if (a > b) {  
    {{a}} is greater than {{b}}  
} @else if (b > a) {  
    {{a}} is less than {{b}}  
} @else {  
    {{a}} is equal to {{b}}  
}
```

Com um simples if é possível fazer a condição seguindo com a lógica

17,2 Instalação Angular 17+

sexta-feira, 18 de outubro de 2024 20:40

1. PRIMEIRAMENTE, VER SE O NODE ESTA BAIXADA NO PC

```
C:\Users\conta.ACER-ANDRE-SANT>node --version  
v20.17.0
```

```
C:\Users\conta.ACER-ANDRE-SANT>
```

- a. SE ESTIVER INSTALADO OK
- b. SE NÃO, BAIXAR A VERSAO LTS



2. INSTALAÇÃO ANGULAR

Com a instalação do node ele automaticamente instala o npm

Agora para instalar o angular é necessário

DENTRO DO CMD ***COMO ADMIN
rodar o comando...

```
C:\Windows\System32>npm install -g @angular/cli  
added 266 packages in 20s  
49 packages are looking for funding  
  run `npm fund` for details
```

```
npm install -g @Angular/cli  
-G para instalar globalmente na maquina  
CLI é Comand Line Interface
```

Como foi implementado de forma GLOBAL no Windows, o Windows entenderá comandos Angular

TODO COMANDO ANGULAR COMEÇA COM NG

Para verificar se foi instalado e a versão
Ng version ou

```

ng v
C:\Windows\System32>ng v

Angular CLI: 18.2.9
Node: 20.17.0
Package Manager: npm 10.8.2
OS: win32 x64

Angular:
...
Package          Version
-----
@angular-devkit/architect    0.1802.9 (cli-only)
@angular-devkit/core         18.2.9 (cli-only)
@angular-devkit/schematics   18.2.9 (cli-only)
@schematics/angular          18.2.9 (cli-only)

```

E SE DER ERRO?

Quando aparece uma mensagem de erro de política

```

ng : File C:\Users\ASUS\AppData\Roaming\npm\ng.ps1 cannot be loaded because running scripts
is disabled on this system. For more information, see about_Execution_Policies at
https://go.microsoft.com/fwlink/?LinkID=135170.
At line:1 char:1
+ ng version
+ ~~~
+ CategoryInfo          : SecurityError: () [], PSSecurityException
+ FullyQualifiedErrorId : UnauthorizedAccess

```

Normalmente é a política do computador Pois entende que aquele script pode ser malicioso e por padrão, o bloqueia

Caso esteja você não for esse problema necessário rodar esse comando para permitir que seja liberado scripts com assinaturas seguras

Set-ExecutionPolicy RemoteSigned -Scope CurrentUser

```

frontend-editor-2024-07-10T23_47_16.817Z on [!] main via [v20.10.0]
→ Set-ExecutionPolicy RemoteSigned -Scope CurrentUser
.
.
.
frontend-editor-2024-07-10T23_47_16.817Z on [!] main via [v20.10.0]
→ [!]

```

Para saber se deu certo necessário rodar o comando de list para listar a política RemoteSigned

Get-ExecutionPolicy -list

```

frontend-editor-2024-07-10T23_47_16.817Z on [!] main via [v20.10.0]
→ Get-ExecutionPolicy -list

Scope ExecutionPolicy
-----
MachinePolicy      Undefined
UserPolicy        Undefined
Process           Undefined
CurrentUser       RemoteSigned
LocalMachine      RemoteSigned

```

17,3 Projetos Angular

sábado, 19 de outubro de 2024 00:08

Antes de começar a programação em angular
é bom ter uma base em HTML | CSS | javascript ou typescript

Comandos NG === Comandos Angular

Ng help -> Mostra os comandos do Angular

17,4 Comandos Angular GERAL

sábado, 19 de outubro de 2024 00:36

Comando para verificar os comandos --> ng help

Commands:

| | |
|-------------------------------|---|
| ng add <collection> | Adds support for an external library to your project. |
| ng analytics | Configures the gathering of Angular CLI usage metrics. |
| ng build [project] | Compiles an Angular application or library into an output directory named dist/ at the given output path.
[aliases: b] |
| ng cache | Configure persistent disk cache and retrieve cache statistics. |
| ng completion | Set up Angular CLI autocompletion for your terminal. |
| ng config [json-path] [value] | Retrieves or sets Angular configuration values in the angular.json file for the workspace. |
| ng deploy [project] | Invokes the deploy builder for a specified project or for the default project in the workspace. |
| ng e2e [project] | Builds and serves an Angular application, then runs end-to-end tests
[aliases: e] |
| ng extract-i18n [project] | Extracts i18n messages from source code. |
| ng generate | Generates and/or modifies files based on a schematic.
[aliases: g] |
| ng lint [project] | Runs linting tools on Angular application code in a given project folder. |
| ng new [name] | Creates a new Angular workspace.
[aliases: n] |
| ng run <target> | Runs an Architect target with an optional custom builder configuration defined in your project. |
| ng serve [project] | Builds and serves your application, rebuilding on file changes.
[aliases: dev, s] |
| ng test [project] | Runs unit tests in a project.
[aliases: t] |
| ng update [packages..] | Updates your workspace and its dependencies. See https://update.angular.dev/ . |
| ng version | Outputs Angular CLI version.
[aliases: v] |

Options:

| | | |
|-----------|---|-----------|
| --help | Shows a help message for this command in the console. | [boolean] |
| --version | Show Angular CLI version. | [boolean] |

17,4,1 Principais Comandos

sábado, 19 de outubro de 2024

00:42

| | |
|-------------------|---|
| NG HELP | Verifica todos os comandos possíveis |
| NG NEW | Crie um novo espaço ou projeto para o Angular |
| NG NEW WITH ALIAS | NG N ... É igual o NG NEW... |

Lembrando que no front diferentemente do java ou back

Quando quero jogar uma frase em vez de palavra em vez de ficar colocando maiúsculos meio da frase eu posso separá-los por traço
my-first-project

Então relembrando

para criar um novo projeto angular necessário

Ng new project-angular...

Selecionar escopo CSS [<https://developer.mozilla.org/docs/Web/CSS>]

17,5 O que faz os Arquivos TS de Config?

sábado, 19 de outubro de 2024 20:05

Como visto em TS

O arquivo **TS CONFIG** é disponibilizado para funções que podem ser ativadas ou desativadas
Que visam facilitar na hora do RUN ou outro momento

tsconfig.spec.json

Esse arquivo é realizado para TESTES

The screenshot shows the VS Code interface with the following details:

- EXPLORER** view: Shows the project structure under "AULAS-ANGULAR". The "my-first-project" folder contains files like .angular, .vscode, node_modules, public, and src. Inside src, there's an app folder containing index.html, main.ts, styles.css, .editorconfig, .gitignore, angular.json, package-lock.json, package.json, README.md, tsconfig.app.json, tsconfig.json, and tsconfig.spec.json. The tsconfig.spec.json file is currently selected.
- CODE EDITOR**: Displays the content of tsconfig.spec.json. The code is:

```
1  /* To learn more about Typescript configurat
2  /* To learn more about Angular compiler opti
3  {
4      "extends": "./tsconfig.json",
5      "compilerOptions": {
6          "outDir": "./out-tsc/spec",
7          "types": [
8              "jasmine"
9          ],
10     },
11     "include": [
12         "src/**/*spec.ts",
13         "src/**/*d.ts"
14     ]
15 }
```
- PANELS**: Shows tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is selected), and PORTS.

Note que o Arquivo EXTENDS busca de um arquivo SUPER algumas configurações

Ou seja, quando chamado o EXTENDS basicamente vc pega o arquivo pai
e muda somente o que precisa utiizarn aquele codigo

17,6 Angular e Node.js

sábado, 19 de outubro de 2024

21:14

Alguns arquivos de configuração são feitos não pelo NG mas pelo Node

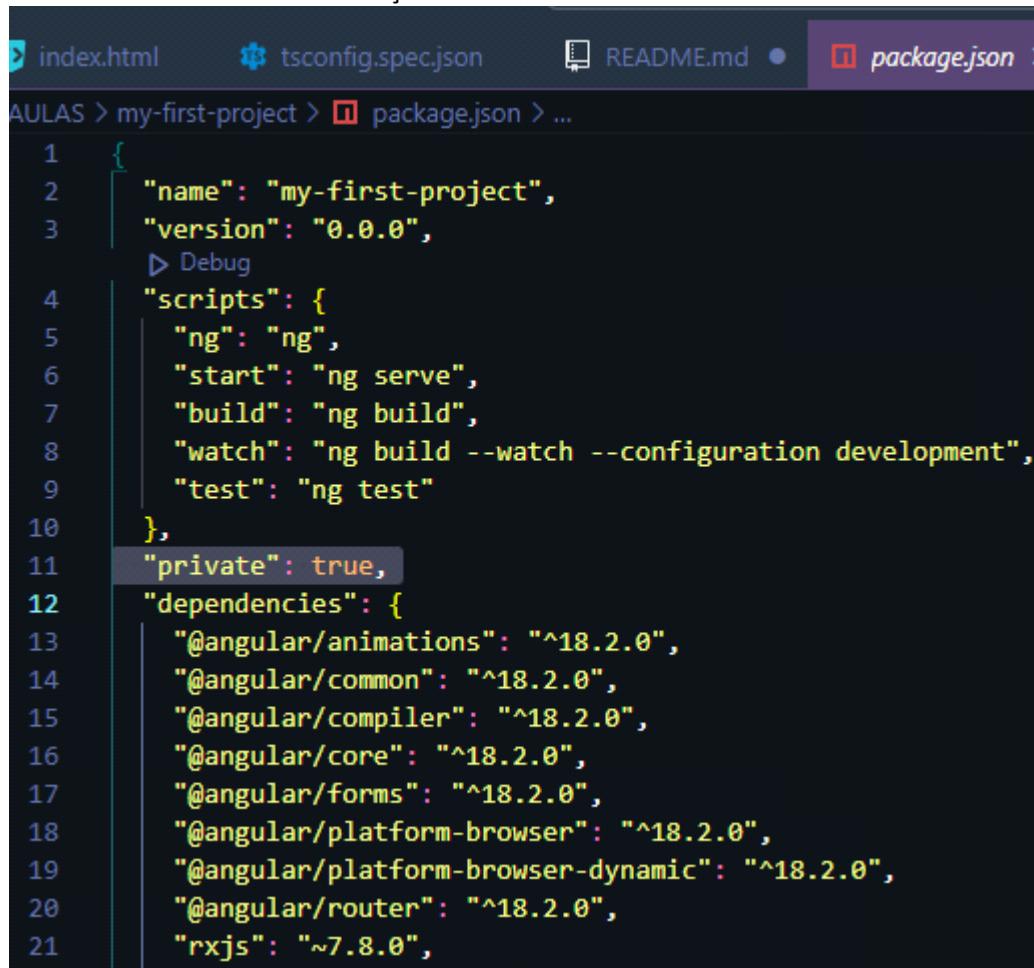
Pois basicamente eles andam de mãos dadas

Como por exemplo o PACKAGE.JSON que empacota/encapsula alguns detalhes do projeto e scripts

Através dele conseguimos dar um "run" no projeto como NPM Start

Uma parte muito importante é o PRIVATE, que faz o arquivo por default ser PRIVADO

Evitando vazamento de informações



```
index.html tsconfig.spec.json README.md package.json
AULAS > my-first-project > package.json > ...
1  {
2    "name": "my-first-project",
3    "version": "0.0.0",
4    "scripts": {
5      "ng": "ng",
6      "start": "ng serve",
7      "build": "ng build",
8      "watch": "ng build --watch --configuration development",
9      "test": "ng test"
10   },
11   "private": true,
12   "dependencies": {
13     "@angular/animations": "^18.2.0",
14     "@angular/common": "^18.2.0",
15     "@angular/compiler": "^18.2.0",
16     "@angular/core": "^18.2.0",
17     "@angular/forms": "^18.2.0",
18     "@angular/platform-browser": "^18.2.0",
19     "@angular/platform-browser-dynamic": "^18.2.0",
20     "@angular/router": "^18.2.0",
21     "rxjs": "~7.8.0",
```

No entanto como ele faz esse private true

Quando você vai puxar novamente o projeto ou dar um update ali dar um git clone

Ele vai perder o projeto de configuração ali do package Jason e as outras configurações

Aí para voltar normal e puxar todas as dependências novamente é só abrir o terminal no local onde está o arquivo e rodar o npm instal ele vai verificar o projeto e vai ver que as dependências que faltam são as que fazem o angular rodar

"npm install"

17,7 Entendendo a Configuração do ANGULAR

sábado, 19 de outubro de 2024 22:09

Como já foi baixado o projeto angular

não é necessário você configurá-lo,

No entanto a nível de curiosidade, ele fica dentro de um arquivo escrito angular.json

Dentro dele já fica todas as configurações

seja do título do projeto o esquema que ele tem que seguir entre outras

configurações

como Build,

como as configurações do TS config...

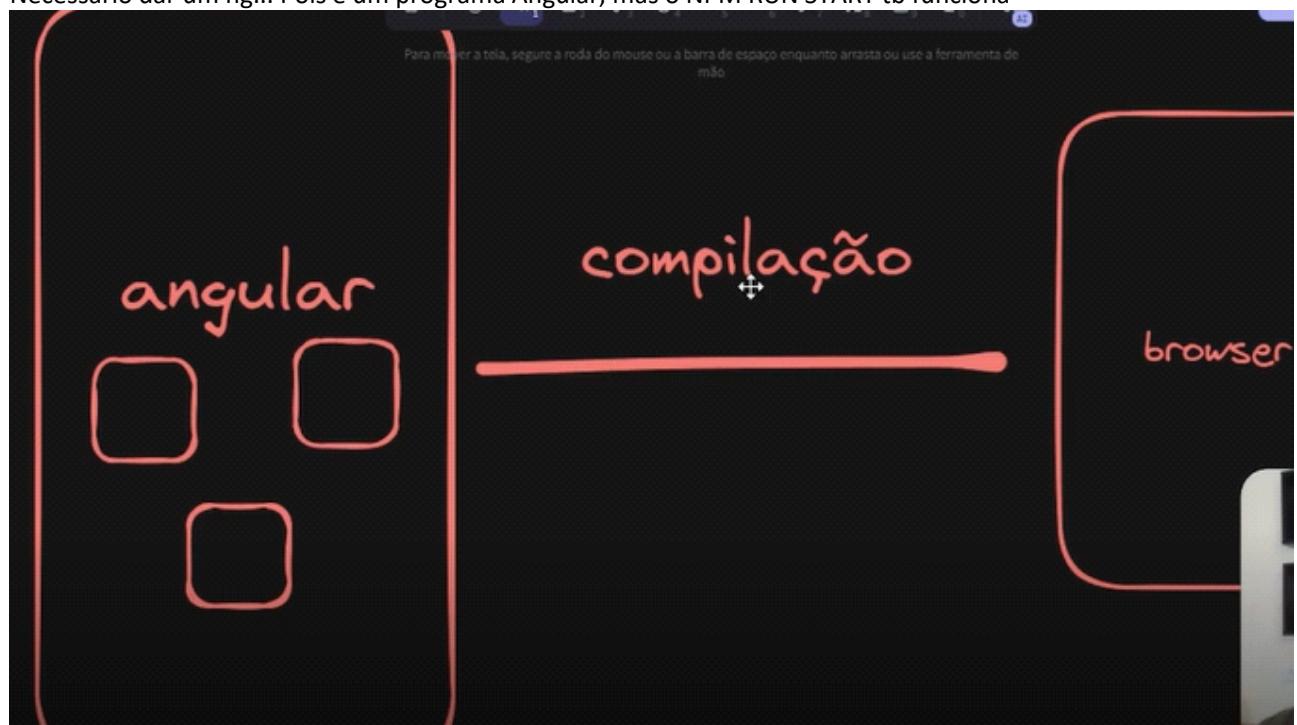
The screenshot shows the VS Code interface with the following details:

- EXPLORER** sidebar: Shows the project structure under "AULAS-ANGULAR". The "angular.json" file is selected.
- EDITOR**: The "angular.json" file is open, displaying its JSON content:

```
1 {  
2   "$schema": "./node_modules/@angular/cli/lib/config/schema.json",  
3   "version": 1,  
4   "newProjectRoot": "projects",  
5   "projects": {  
6     "my-first-project": {  
7       "projectType": "application",  
8       "schematics": {},  
9       "root": "",  
10      "sourceRoot": "src",  
11      "prefix": "app",  
12      "architect": {  
13        "build": {  
14          "builder": "@angular-devkit/build-angular:application",  
15          "options": {  
16            "outputPath": "dist/my-first-project",  
17            "index": "src/index.html",  
18            "browser": "src/main.ts",  
19            "polyfills": [  
20              "zone.js"  
21            ],  
22            "tsConfig": "tsconfig.app.json",  
23          }  
24        }  
25      }  
26    }  
27  }  
28 }
```
- PAINEL DE FERRAMENTAS**: Problems, Output, Debug Console, Terminal, Ports.
- STATUS BAR**: PS D:\conta.ACER-ANDRE-SANT\Ambiente de Desenvolvimento\DI - Bootcamps\Bootcamp_Springboot_Angular

Agora para rodar o programa

Necessário dar um ng... Pois é um programa Angular, mas o NPM RUN START tb funciona



O Como o Angular é um framework

ele simplesmente ajuda o código a ficar mais fácil então quando faz essa compilação ele simplesmente transforma novamente o angular em HTML e JavaScript

NG SERVER

É o processo de buildar/construir a aplicação e logo em seguida o lança em server

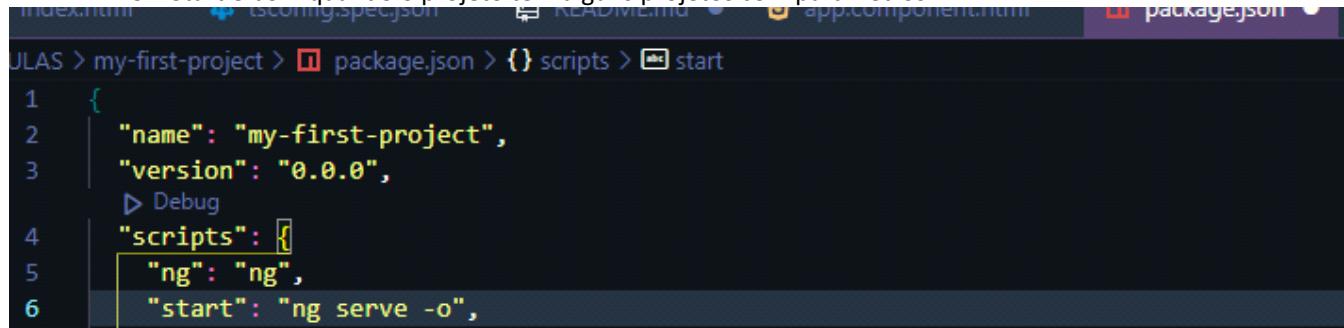
ng serve [project] Builds and serves your application, rebuilding on file changes.

[aliases: dev, s]

NG S -O

-o == Open, então ele compila, roda e abre

Em NPM RUN Start é bom quando o projeto tem alguns projetos com parâmetros



```
index.html  tsconfig.json  README.md  app.component.html  package.json
JLAS > my-first-project > package.json > {} scripts > start
1  {
2    "name": "my-first-project",
3    "version": "0.0.0",
4    "scripts": {
5      "ng": "ng",
6      "start": "ng serve -o",
```

PARA PARAR O SERVIDOR ELE DA UMA OPÇÃO MAIS CERTA EM VEZ DE CTRL C

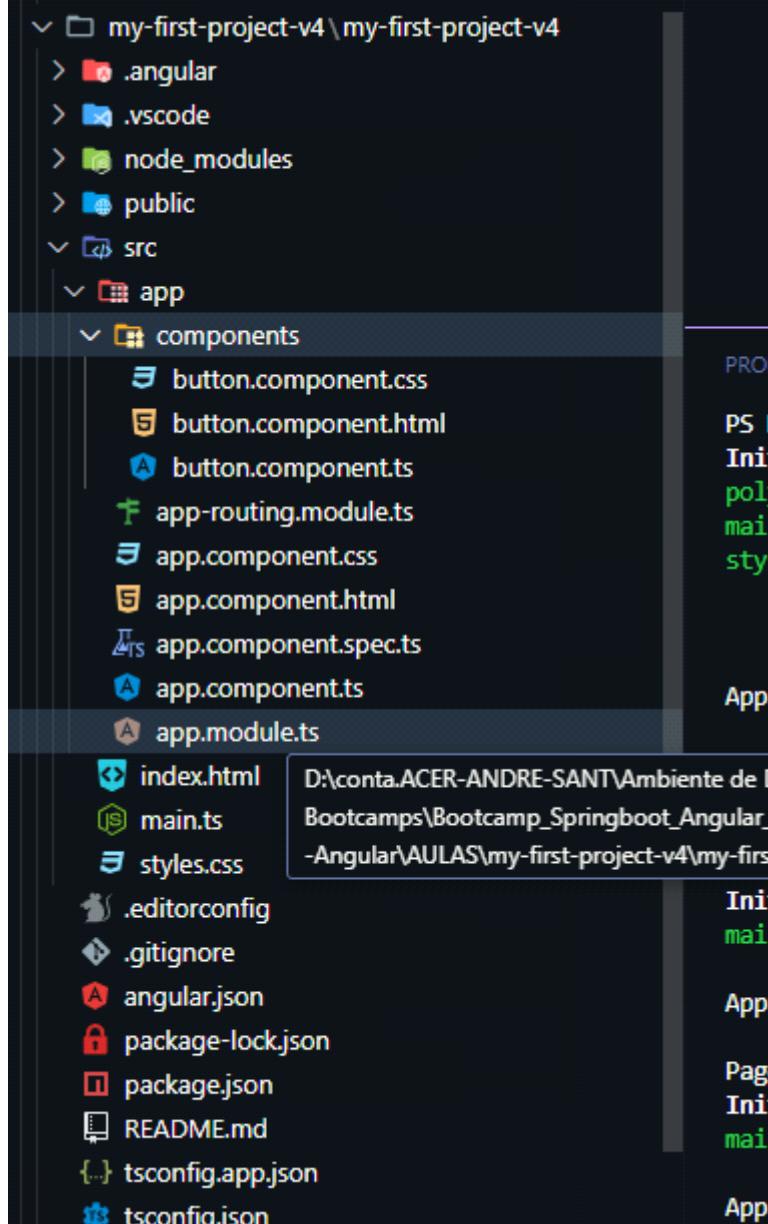
```
→ press h + enter to show help
h

Shortcuts
press r + enter to force reload browser
press u + enter to show server url
press o + enter to open in browser
press c + enter to clear console
press q + enter to quit
q
```

17,8 Criando Components em Angular

domingo, 20 de outubro de 2024 00:33

Note que quando criado um projeto, ele já vem com uma estrutura pre-montada



Nesse caso quando queremos criar um novo componente, para ser chamado como se fosse um METODO

Podemos criar da seguinte forma

!!! NOTA: Quando vamos criar um componente, uma boa pratica é deixar todos com o mesmo nome...

Button.component.html | Button.component.css | **Button.component.js** |
Button.component.ts

Para definirmos que ele é um component, precisamos informar dentro do .TS atraves do ANGULAR/CORE

JUNTO A ISSO CHAMAR O @Component

E exportar a classe

```

import { Component } from '@angular/core';


- @Component ({})


export class Button{}

```

^> Esse é o modelo padrão que um COMPONENTE deve ser

NO TYPESCRIPT

Finalizando-o no arquivo .TS, fica dessa forma

```

import { Component } from '@angular/core';

@Component ({
  selector: 'my-button',
  templateUrl: './button.component.html',
  // styleUrls: ['./button.component.css'], /*SE FOR MAIS DE UMA FOLHA DE STYLING */
  styleUrls: ['./button.component.css'],
})

export class Button{}

```

NO APP.MODULE.TS

Como mesmo fazendo dessa forma ele da erro por não estar em nenhuma "gaveta"
Necessário implementar dentro de um módulo

```

> my-first-project-v3 > my-first-project-v3 > src > app > A app.module.ts > ...
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { FormsModule } from '@angular/forms';

import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { Button } from './components/button.component';

@NgModule({
  declarations: [AppComponent, Button],
  imports: [BrowserModule, AppRoutingModule],
  providers: [],
  bootstrap: [AppComponent],
})
export class AppModule {}

```

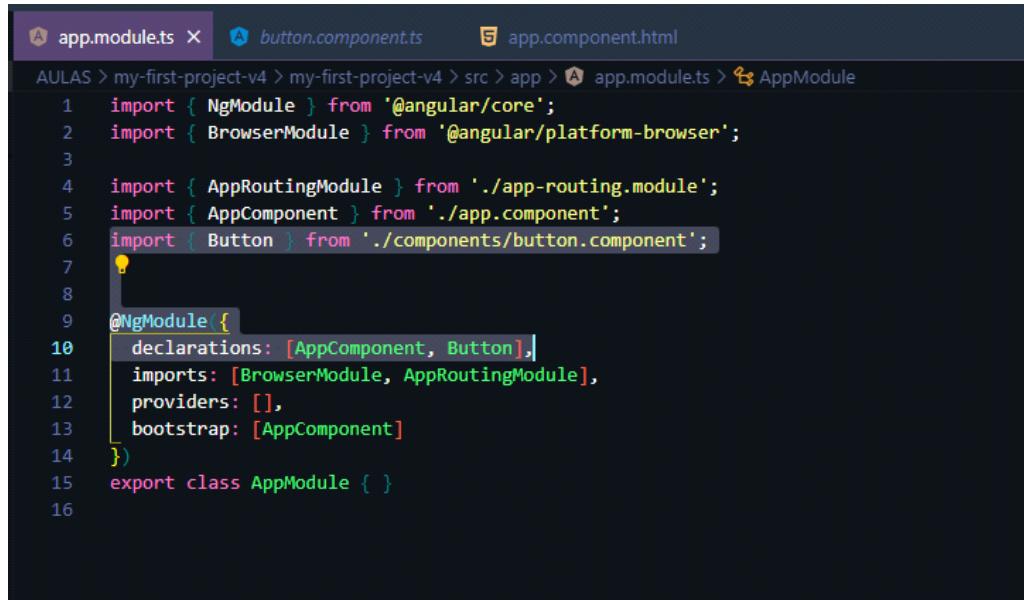
NO HTML DO BUTTON FICA DESSA FORMA

```

<div>
  <button>
    Click Aqui
  </button>
</div>

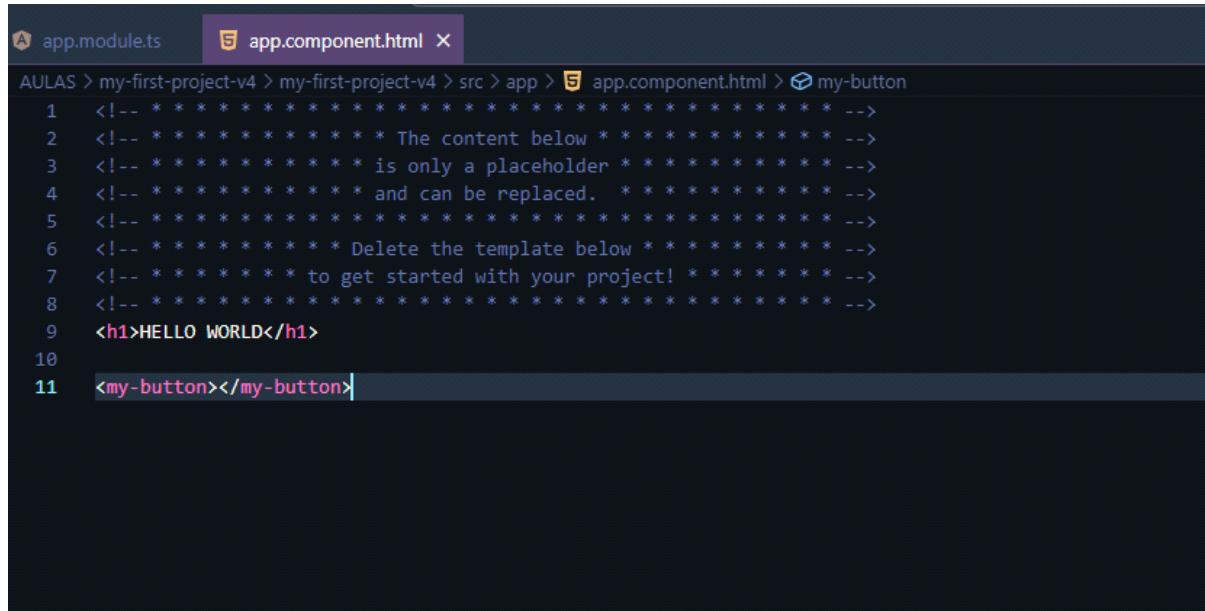
```

NO APP.MODULE.TS DESSA FORMA



```
AULAS > my-first-project-v4 > my-first-project-v4 > src > app > A app.module.ts > AppModule
  1 import { NgModule } from '@angular/core';
  2 import { BrowserModule } from '@angular/platform-browser';
  3
  4 import { AppRoutingModule } from './app-routing.module';
  5 import { AppComponent } from './app.component';
  6 import { Button } from './components/button.component';
  7
  8
  9 @NgModule({
10   declarations: [AppComponent, Button],
11   imports: [BrowserModule, AppRoutingModule],
12   providers: [],
13   bootstrap: [AppComponent]
14 })
15 export class AppModule { }
```

NO HTML PRINCIPAL DO PROJETO



```
AULAS > my-first-project-v4 > my-first-project-v4 > src > app > S app.component.html > my-button
  1 <!-- * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * -->
  2 <!-- * * * * * * * * * * * * * * * * * The content below * * * * * * * * * * * -->
  3 <!-- * * * * * * * * * * * is only a placeholder * * * * * * * * * * * -->
  4 <!-- * * * * * * * * * * * and can be replaced. * * * * * * * * * * * -->
  5 <!-- * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * -->
  6 <!-- * * * * * * * * * * * Delete the template below * * * * * * * * * * * -->
  7 <!-- * * * * * * * * * * * to get started with your project! * * * * * * * * * -->
  8 <!-- * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * -->
  9 <h1>HELLO WORLD</h1>
10
11 <my-button></my-button>
```

HELLO WORLD

[Click Aqui](#)

17,2,1 Criando projeto Angular com Erro no Module (RESOLVIDO)

quarta-feira, 23 de outubro de 2024 21:20

Depois de muito esforço e discussão com o Chat GPT

Conseguimos resolver varios conflitos ao criar o APP.MODULE.TS

CRIAÇÃO DE PROJETO ANGULAR

Lembre que em front, o certo é separar frases com ' - '

Nome-sobrenome-alguma-coisa

PARA CRIAR PROJETO ==

Ng new nome-projeto --no-standalone

SELECIIONAR ESCOPO

Css

COMO É PARA PROJETO PESSOAL

No

**CONFIRA SE EM SRC/APP
ESTA O APP.MODULE.TS**

**Se igual o meu não tem, pode cria-lo manualmente ou com ajuda
do angular/cli ou ver se criou com o comando -no-standalone**

1. Navegue até a pasta `src/app`:

Gere o Módulo Corretamente

ng generate module app --flat

Essa opção `--flat` garante que o `app.module.ts` seja criado diretamente na pasta `src/app` sem criar uma subpasta.

Verificando a Estrutura

Depois de executar o comando acima, verifique se o arquivo `app.module.ts` foi criado corretamente na pasta `src/app`:

2. Adicionar Código ao Módulo

Se o arquivo foi criado com sucesso, você pode abrir `app.module.ts` e adicionar o código do módulo que mencionei anteriormente.

17,8,1 Aperfeiçoando o Componente

quinta-feira, 24 de outubro de 2024 20:22

Interpolação de dados

Quando passamos os dados dinamicamente para o HTML

Nessa parte, alem de criar um componente, vamos inserir logica nele
De forma que poderemos utilizar ele em diferentes cenarios com situações diferentes

VEJA COMO PODEMOS UTILIZA-LO NO TS

```
export class Button{
  buttonText1: string = 'ACESSAR';
  buttonTexts: string[] = ['VENDER', 'COMPRAR']; /*INDEX 0,1 */
  buttonNumber: number = 1;
  // SE EU QUISER CRIAR UM OBJETO
  buttonObject={
    label: 'Adicionar ao Carrinho',
  };
}
```

NO HTML DO BUTTON, FAZENDO UM TESTE DESSA FORMA

```
AULAS > my-first-project-v4 > my-first-project-v4 > src > app > components > button.component.html > div > button
1  <div><button>
2  |   Click Aqui
3  </button></div>
4
5  <div><button>
6  |   {{buttonText1}}
7  </button></div>
8
9  <div><button>
10 |   {{buttonTexts[1]}}
11 </button></div>
12
13 <div><button>
14 |   {{buttonNumber}}
15 </button></div>
16
17 <div><button>
18 |   {{buttonObject.label}}
19 </button></div>
20
```

Dessa forma o que é como implementarmos, quando definido no HTML do projeto fica dessa forma

```
1  <!-- * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * -->
2  <!-- * * * * * * * * * * * * * The content below * * * * * * * * * * * -->
3  <!-- * * * * * * * * * * is only a placeholder * * * * * * * * * * -->
4  <!-- * * * * * * * * * and can be replaced. * * * * * * * * * -->
5  <!-- * * * * * * * * * * * * * * * * * * * * * * * * * * * * * -->
6  <!-- * * * * * * * * * Delete the template below * * * * * * * * * -->
7  <!-- * * * * * * * * to get started with your project! * * * * * * * -->
8  <!-- * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * -->
9  <h1>HELLO WORLD</h1>
10
11 <my-button></my-button>
12
```

HELLO WORLD

[Click Aqui](#)
[ACESSAR](#)
[COMPRAR](#)
1
[Adicionar ao Carrinho](#)

17,8,2 Estilizando Componentes

quinta-feira, 24 de outubro de 2024 20:58

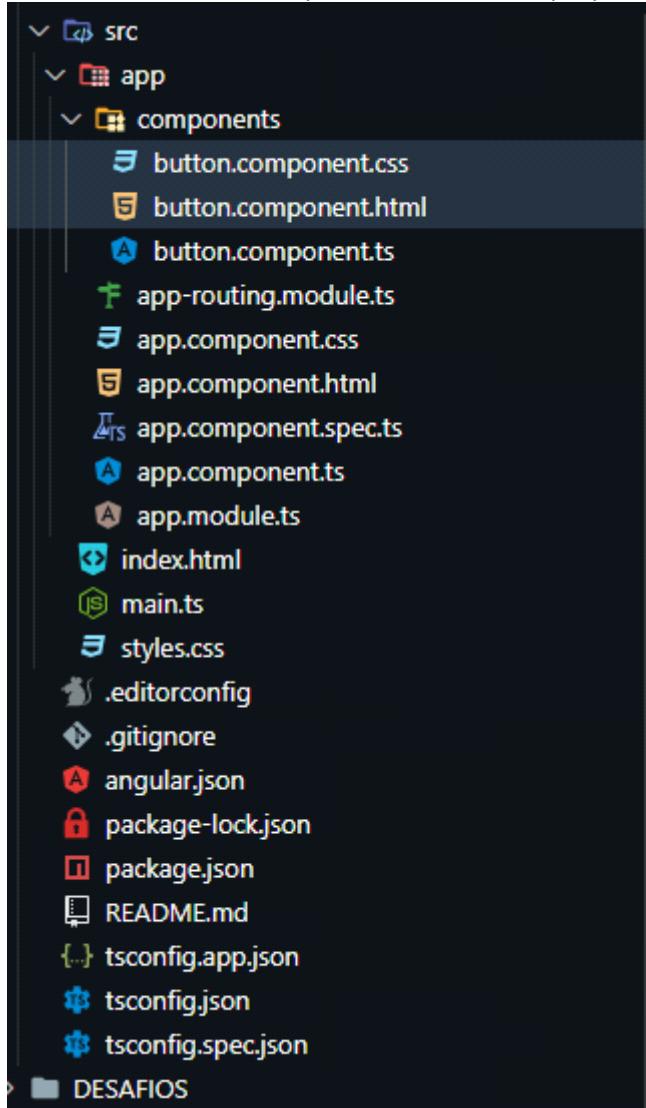
O projeto em si, tem um STYLES CSS

Esse CSS é um CSS padrao e global do projeto

Tudo o que não tiver implementação propria e reservada, entra nesse padrao

No entanto

Podemos fazer cada componente ter seu CSS proprio



17,8,3 Comunicação entre Componentes

quinta-feira, 24 de outubro de 2024 21:25

Levando em consideração que o componente APP.COMPONENT.TS está acima dos components setados

Podemos levar em conta que o AppComponent é o componente pai

Dessa forma, conseguimos fazer com que o elemento Pai converse com seus elementos filhos

Dessa forma, conseguimos reutilizar aqueles códigos mencionados antes, fazendo por LABELS

Então um componente conversará com o outro
para dessa forma a parecer a informação de forma dinâmica

Para isso

NO COMPONENTE PAI (APP.COMPONENT.TS)

buttonLabel : string = 'CARRINHO'

```
1 import { Component } from '@angular/core';
2
3 @Component({
4   selector: 'app-root',
5   templateUrl: './app.component.html',
6   styleUrls: ['./app.component.css']
7 })
8 export class AppComponent {
9   title = 'my-first-project-v4';
10   buttonLabel : string = 'CARRINHO'
11 }
```

AGORA DENTRO DO COMP. FILHO

Para imputar dados de uma propriedade em outra propriedade,

O NG tem uma biblioteca que faz isso

```
1 import { Component, Input } from '@angular/core';
2
3 @Component({
```

Dessa forma, quando importado essa biblioteca Input

Temos que utilizar da Annotation @Input

Essa annotation deve ser implementada dentro da classe filho

Ou seja quando ela é implementada, a sua iniciação dentro da classe é opcional

Se não deixar essa classe como opcional é bem provável que dê erro na hora da execução se não definir um parâmetro e mesmo definindo o ele acabará dando erro

Dentro das classes pai

APP.componente.TS | App.componente.HTML

```

app.component.ts
AULAS > my-first-project-v4 > my-first-project-v4 > src > app > app.component.ts > ...
1 import { Component } from '@angular/core';
2
3 @Component({
4   selector: 'app-root',
5   templateUrl: './app.component.html',
6   styleUrls: ['./app.component.css']
7 })
8 export class AppComponent {
9   buttonLabel : string = 'CARRINHO';
10  buttonSecond : string = 'ADICIONAR AOS FAVORITOS';
11 }
12

app.component.html
AULAS > my-first-project-v4 > my-first-project-v4 > src > app > app.component.html > ...
1 <!-- * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * -->
2 <!-- * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * -->
3 <!-- * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * -->
4 <!-- * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * -->
5 <!-- * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * -->
6 <!-- * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * -->
7 <!-- * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * -->
8 <!-- * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * -->
9 <h1>HELLO WORLD</h1>
10 <my-button label="{{buttonLabel}}></my-button>
11 <my-button label="{{buttonSecond}}></my-button>
12
13

```

Dentro das subClasses

```

app.component.ts
AULAS > my-first-project-v4 > my-first-project-v4 > src > app > components > button.component.ts > ...
1 import { Component, Input } from '@angular/core';
2
3 @Component({
4   selector: 'my-button',
5   templateUrl: './button.component.html',
6   styleUrls: ['./button.component.css'],
7 })
8
9 export class Button{
10   @Input() label?:string;
11   // IGNORE THIS BELOW AGORA
12   // buttonText1: string = 'ACESSAR';
13   // buttonTexts: string[] = ['VENDER', 'COMPRAR'];
14   // buttonNumber: number = 1;
15   // SE EU QUISER CRIAR UM OBJETO
16   // buttonObject={
17   //   label: 'Adicionar ao Carrinho',
18   // };
19 }

button.component.html
AULAS > my-first-project-v4 > my-first-project-v4 > src > app > components > button.component.html > ...
1 <div>
2   <button>
3     {{ label }}
4   </button>
5 </div>
6
7 <!-- <div><button>
8   Click Aqui
9 </button></div>
10 <div><button>
11   {{buttonText1}}
12 </button></div>
13 <div><button>
14   {{buttonTexts[1]}}
15 </button></div>
16 <div><button>
17   {{buttonNumber}}
18 </button></div>
19 <div><button>
20   {{buttonObject.label}}
21 </button></div>
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
259
260
261
262
263
264
265
266
267
268
269
269
270
271
272
273
274
275
276
277
278
279
279
280
281
282
283
284
285
286
287
287
288
289
289
290
291
292
293
294
295
296
297
297
298
299
299
300
301
302
303
304
305
306
307
308
309
309
310
311
311
312
313
313
314
315
315
316
316
317
317
318
318
319
319
320
320
321
321
322
322
323
323
324
324
325
325
326
326
327
327
328
328
329
329
330
330
331
331
332
332
333
333
334
334
335
335
336
336
337
337
338
338
339
339
340
340
341
341
342
342
343
343
344
344
345
345
346
346
347
347
348
348
349
349
350
350
351
351
352
352
353
353
354
354
355
355
356
356
357
357
358
358
359
359
360
360
361
361
362
362
363
363
364
364
365
365
366
366
367
367
368
368
369
369
370
370
371
371
372
372
373
373
374
374
375
375
376
376
377
377
378
378
379
379
380
380
381
381
382
382
383
383
384
384
385
385
386
386
387
387
388
388
389
389
390
390
391
391
392
392
393
393
394
394
395
395
396
396
397
397
398
398
399
399
400
400
401
401
402
402
403
403
404
404
405
405
406
406
407
407
408
408
409
409
410
410
411
411
412
412
413
413
414
414
415
415
416
416
417
417
418
418
419
419
420
420
421
421
422
422
423
423
424
424
425
425
426
426
427
427
428
428
429
429
430
430
431
431
432
432
433
433
434
434
435
435
436
436
437
437
438
438
439
439
440
440
441
441
442
442
443
443
444
444
445
445
446
446
447
447
448
448
449
449
450
450
451
451
452
452
453
453
454
454
455
455
456
456
457
457
458
458
459
459
460
460
461
461
462
462
463
463
464
464
465
465
466
466
467
467
468
468
469
469
470
470
471
471
472
472
473
473
474
474
475
475
476
476
477
477
478
478
479
479
480
480
481
481
482
482
483
483
484
484
485
485
486
486
487
487
488
488
489
489
490
490
491
491
492
492
493
493
494
494
495
495
496
496
497
497
498
498
499
499
500
500
501
501
502
502
503
503
504
504
505
505
506
506
507
507
508
508
509
509
510
510
511
511
512
512
513
513
514
514
515
515
516
516
517
517
518
518
519
519
520
520
521
521
522
522
523
523
524
524
525
525
526
526
527
527
528
528
529
529
530
530
531
531
532
532
533
533
534
534
535
535
536
536
537
537
538
538
539
539
540
540
541
541
542
542
543
543
544
544
545
545
546
546
547
547
548
548
549
549
550
550
551
551
552
552
553
553
554
554
555
555
556
556
557
557
558
558
559
559
560
560
561
561
562
562
563
563
564
564
565
565
566
566
567
567
568
568
569
569
570
570
571
571
572
572
573
573
574
574
575
575
576
576
577
577
578
578
579
579
580
580
581
581
582
582
583
583
584
584
585
585
586
586
587
587
588
588
589
589
590
590
591
591
592
592
593
593
594
594
595
595
596
596
597
597
598
598
599
599
600
600
601
601
602
602
603
603
604
604
605
605
606
606
607
607
608
608
609
609
610
610
611
611
612
612
613
613
614
614
615
615
616
616
617
617
618
618
619
619
620
620
621
621
622
622
623
623
624
624
625
625
626
626
627
627
628
628
629
629
630
630
631
631
632
632
633
633
634
634
635
635
636
636
637
637
638
638
639
639
640
640
641
641
642
642
643
643
644
644
645
645
646
646
647
647
648
648
649
649
650
650
651
651
652
652
653
653
654
654
655
655
656
656
657
657
658
658
659
659
660
660
661
661
662
662
663
663
664
664
665
665
666
666
667
667
668
668
669
669
670
670
671
671
672
672
673
673
674
674
675
675
676
676
677
677
678
678
679
679
680
680
681
681
682
682
683
683
684
684
685
685
686
686
687
687
688
688
689
689
690
690
691
691
692
692
693
693
694
694
695
695
696
696
697
697
698
698
699
699
700
700
701
701
702
702
703
703
704
704
705
705
706
706
707
707
708
708
709
709
710
710
711
711
712
712
713
713
714
714
715
715
716
716
717
717
718
718
719
719
720
720
721
721
722
722
723
723
724
724
725
725
726
726
727
727
728
728
729
729
730
730
731
731
732
732
733
733
734
734
735
735
736
736
737
737
738
738
739
739
740
740
741
741
742
742
743
743
744
744
745
745
746
746
747
747
748
748
749
749
750
750
751
751
752
752
753
753
754
754
755
755
756
756
757
757
758
758
759
759
760
760
761
761
762
762
763
763
764
764
765
765
766
766
767
767
768
768
769
769
770
770
771
771
772
772
773
773
774
774
775
775
776
776
777
777
778
778
779
779
780
780
781
781
782
782
783
783
784
784
785
785
786
786
787
787
788
788
789
789
790
790
791
791
792
792
793
793
794
794
795
795
796
796
797
797
798
798
799
799
800
800
801
801
802
802
803
803
804
804
805
805
806
806
807
807
808
808
809
809
810
810
811
811
812
812
813
813
814
814
815
815
816
816
817
817
818
818
819
819
820
820
821
821
822
822
823
823
824
824
825
825
826
826
827
827
828
828
829
829
830
830
831
831
832
832
833
833
834
834
835
835
836
836
837
837
838
838
839
839
840
840
841
841
842
842
843
843
844
844
845
845
846
846
847
847
848
848
849
849
850
850
851
851
852
852
853
853
854
854
855
855
856
856
857
857
858
858
859
859
860
860
861
861
862
862
863
863
864
864
865
865
866
866
867
867
868
868
869
869
870
870
871
871
872
872
873
873
874
874
875
875
876
876
877
877
878
878
879
879
880
880
881
881
882
882
883
883
884
884
885
885
886
886
887
887
888
888
889
889
890
890
891
891
892
892
893
893
894
894
895
895
896
896
897
897
898
898
899
899
900
900
901
901
902
902
903
903
904
904
905
905
906
906
907
907
908
908
909
909
910
910
911
911
912
912
913
913
914
914
915
915
916
916
917
917
918
918
919
919
920
920
921
921
922
922
923
923
924
924
925
925
926
926
927
927
928
928
929
929
930
930
931
931
932
932
933
933
934
934
935
935
936
936
937
937
938
938
939
939
940
940
941
941
942
942
943
943
944
944
945
945
946
946
947
947
948
948
949
949
950
950
951
951
952
952
953
953
954
954
955
955
956
956
957
957
958
958
959
959
960
960
961
961
962
962
963
963
964
964
965
965
966
966
967
967
968
968
969
969
970
970
971
971
972
972
973
973
974
974
975
975
976
976
977
977
978
978
979
979
980
980
981
981
982
982
983
983
984
984
985
985
986
986
987
987
988
988
989
989
990
990
991
991
992
992
993
993
994
994
995
995
996
996
997
997
998
998
999
999
1000
1000
1001
1001
1002
1002
1003
1003
1004
1004
1005
1005
1006
1006
1007
1007
1008
1008
1009
1009
1010
1010
1011
1011
1012
1012
1013
1013
1014
1014
1015
1015
1016
1016
1017
1017
1018
1018
1019
1019
1020
1020
1021
1021
1022
1022
1023
1023
1024
1024
1025
1025
1026
1026
1027
1027
1028
1028
1029
1029
1030
1030
1031
1031
1032
1032
1033
1033
1034
1034
1035
1035
1036
1036
1037
1037
1038
1038
1039
1039
1040
1040
1041
1041
1042
1042
1043
1043
1044
1044
1045
1045
1046
1046
1047
1047
1048
1048
1049
1049
1050
1050
1051
1051
1052
1052
1053
1053
1054
1054
1055
1055
1056
1056
1057
1057
1058
1058
1059
1059
1060
1060
1061
1061
1062
1062
1063
1063
1064
1064
1065
1065
1066
1066
1067
1067
1068
1068
1069
1069
1070
1070
1071
1071
1072
1072
1073
1073
1074
1074
1075
1075
1076
1076
1077
1077
1078
1078
1079
1079
1080
1080
1081
1081
1082
1082
1083
1083
1084
1084
1085
1085
1086
1086
1087
1087
1088
1088
1089
1089
1090
1090
1091
1091
1092
1092
1093
1093
1094
1094
1095
1095
1096
1096
1097
1097
1098
1098
1099
1099
1100
1100
1101
1101
1102
1102
1103
1103
1104
1104
1105
1105
1106
1106
1107
1107
1108
1108
1109
1109
1110
1110
1111
1111
1112
1112
1113
1113
1114
1114
1115
1115
1116
1116
1117
1117
1118
1118
1119
1119
1120
1120
1121
1121
1122
1122
1123
1123
1124
1124
1125
1125
1126
1126
1127
1127
1128
1128
1129
1129
1130
1130
1131
1131
1132
1132
1133
1133
1134
1134
1135
1135
1136
1136
1137
1137
1138
1138
1139
1139
1140
1140
1141
1141
1142
1142
1143
1143
1144
1144
1145
1145
1146
1146
1147
1147
1148
1148
1149
1149
1150
1150
1151
1151
1152
1152
1153
1153
1154
1154
1155
1155
1156
1156
1157
1157
1158
1158
1159
1159
1160
1160
1161
1161
1162
1162
1163
1163
1164
1164
1165
1165
1166
1166
1167
1167
1168
1168
1169
1169
1170
1170
1171
1171
1172
1172
1173
1173
1174
1174
1175
1175
1176
1176
1177
1177
1178
1178
1179
1179
1180
1180
1181
1181
1182
1182
1183
1183
1184
1184
1185
1185
1186
1186
1187
1187
1188
1188
1189
1189
1190
1190
1191
1191
1192
1192
1193
1193
1194
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1200
1201
1201
1202
1202
1203
1203
1204
1204
1205
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1210
1211
1211
1212
1212
1213
1213
1214
1214
1215
1215
1216
1216
1217
1217
1218
1218
1219
1219
1220
1220
1221
1221
1222
1222
1223
1223
1224
1224
1225
1225
1226
1226
1227
1227
1228
1228
1229
1229
1230
1230
1231
1231
1232
1232
1233
1233
1234
1234
1235
1235
1236
1236
1237
1237
1238
1238
1239
1239
1240
1240
1241
1241
1242
1242
1243
1243
1244
1244
1245
1245
1246
1246
1247
1247
1248
1248
1249
1249
1250
1250
1251
1251
1252
1252
1253
1253
1254
1254
1255
1255
1256
1256
1257
1257
1258
1258
1259
1259
1260
1260
1261
1261
1262
1262
1263
1263
1264
1264
1265
1265
1266
1266
1267
1267
1268
1268
1269
1269
1270
1270
1271
1271
1272
1272
1273
1273
1274
1274
1275
1275
1276
1276
1277
1277
1278
1278
1279
1279
1280
1280
1281
1281
1282
1282
1283
1283
1284
1284
1285
1285
1286
1286
1287
1287
1288
1288
1289
1289
1290
1290
1291
1291
1292
1292
1293
1293
1294
1294
1295
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1300
1301
1301
1302
1302
1303
1303
1304
1304
1305
1305
1306
1306
1307
1307
1308
1308
1309
1309
1310
1310
1311
1311
1312
1312
1313
1313
1314
1314
1315
1315
1316
1316
1317
1317
1318
1318
1319
1319
1320
1320
1321
1321
1322
1322
1323
1323
1324
1324
1325
1325
1326
1326
1327
1327
```

The screenshot shows a code editor interface with several tabs at the top: 'app.component.html' (active), 'button.component.ts', 'button.component.html', and 'angular Ivy-x3s6cr.stackblitz.io'. The main content area displays the following HTML code:

```
1 <hello name="{{ name }}></hello>
2 <menu-component></menu-component>
3
4 <p>Start editing to see some magic happen :)</p>
5 <h1>OLÁ MUNDÃO</h1>
6
7 <my-button label="{{ buttonLabel }}></my-button>
8 <my-button [label]="buttonSecond"></my-button>
9 <my-button label="{{noticia}}></my-button>
```

A red error message is visible on the right side of the editor: 'Error in src/app/app.component.html (9:21) Property 'noticia' does not exist on type 'AppComponent''. The cursor is positioned over the 'noticia' placeholder in the last line of the code.

17,9 Data-Binding Entendendo o que é

sexta-feira, 25 de outubro de 2024 19:44

Data Binding é um tipo de definição (EM PT == ASSOCIAÇÃO DE DADOS)
quando em

um elemento é criado a lógica
e em outro local é criado a parte visual

Esses 2 se unem no HTML

Podemos fazer dessa forma, no entanto, uma boa prática é o primeiro jeito quando utilizado o PROPERTY BINDING

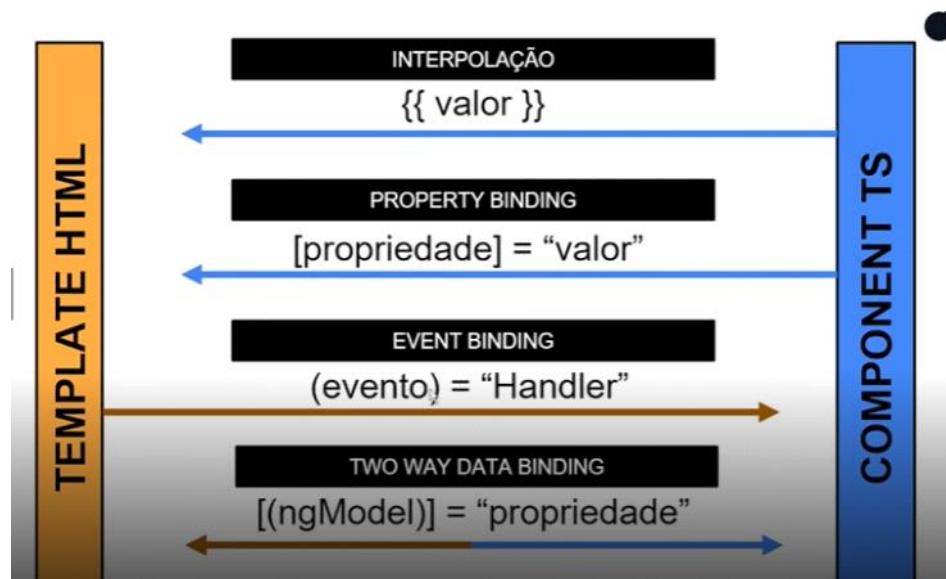
```
<h1>HELLO WORLD</h1>

<my-button [label]="buttonLabel"></my-button>
<my-button label="{{buttonSecond}}></my-button>
```

Então entendendo mais a fundo

Temos a PROPERTY BINDING

São 2 componentes que serão juntados e interpolados no TEMPLATE HTML



EVENT BINDING

Quando mandamos uma ação que faz alguma operação
Tipo: Click aqui para "Adicionar aos Favoritos"

17,9 Event Binding

sexta-feira, 25 de outubro de 2024 21:48

Ação ao clicar em algum botão

Uma lógica relacionada a alguma coisa

Um evento de associação

Para fazer isso, será necessário criar uma regra dentro da classe onde esse botão está

Chamando esse dado e mesclando com essa ação

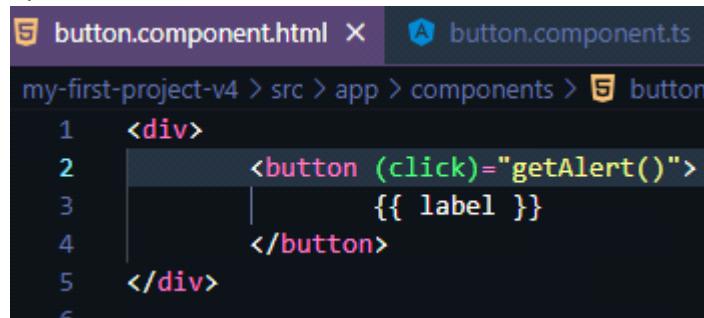
VAMOS IMPLEMENTAR

```
export class Button{
    @Input() label?:string;
    // // IGNORAR ISSO ABAIXO A
    // buttonText1: string = 'A
    // buttonTexts: string[] =
    // buttonNumber: number = 1
    // // SE EU QUISER CRIAR UM
    // buttonObject={
    //     label: 'Adicionar ao
    // };

    getAlert(){
        alert('Olá');
    }
}
```

Note que GET ALERT claramente é um método

Quem irá chamar ele é o COMP...BUTTON...HTML



The screenshot shows two tabs in a code editor: 'button.component.html' and 'button.component.ts'. The 'button.component.html' tab contains the following code:

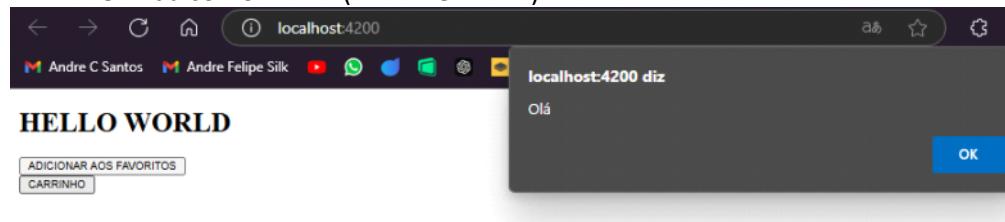
```
<div>
    <button (click)="getAlert()">
        {{ label }}
    </button>
</div>
```

The 'button.component.ts' tab contains the code from the previous block.

Quando rodando a aplicação no servidor

Aperto no botão e surge uma ação

Definida como ALERT (PÁDRAO HTML)



Para saber mais dos eventos,

Conseguimos ver isso no próprio site da Mozilla sobre os eventos em HTML como CLICK,

MOUSEOVER

Em vez de OLA se queremos que ele apareça o nome daquele botão...

```
getAlert(){
  alert(this.label);
}
```



button.component.html button.component.ts

```
my-first-project-v4 > src > app > components > button.component.ts
1   <div>
2     <button (click)="getAlert()">
3       {{ label }}
4     </button>
5   </div>
```

localhost:4200 diz

ADICIONAR AOS FAVORITOS

HELLO WORLD

ADICIONAR AOS FAVORITOS

CARRINHO

Ele também pode ser implementado, pegando alguma informação do TS para a VIEW

```
app.component.html    button.component.ts    button.component.css
o export class Button {
9   @Input() label: string;
10  buttonText: string = 'ACESSAR';
11  buttonTexts: string[] = ['VENDER', 'COMPRAR'];
12  buttonNumber: number = 1;
13  buttonObject = {
14    label: 'adicionar ao carrinho'
15  };
16
17  getAlert(num: number) {
18    alert(num);
19  }
20}
```

Dessa forma ele retorna o número 1

17,10 Two Way - Comunicação nos 2 lados

sexta-feira, 25 de outubro de 2024 22:20

Essa implementação faz aparecer a comunicação entre os 2 lados
seja no typescript e no HTML

Eles se conversam dentro do proprio componente entre a parte logica e o visual

Implementando em um novo componente para conseguirmos visualizar melhor

Em app component html (main do projeto)

```
app.component.html • button.component.ts button.component.html P ⌂
9  <!-- interlocação -->
10 <my-button label="noticia"></my-button>
11 <my-button label="{{ buttonLabel }}"></my-button>
12
13 <!-- property binding -->
14 <my-button [label]="buttonSecond"></my-button>
15 <my-button [label]="'CARRINHO'"></my-button>
16
17 <!-- two way data binding -->
18 <app-entry-data></app-entry-data>
19
```

Em TS

APLICANDO O TWO WAY DATA BINDING

```
app.component.html entry-data.component.ts entry-data.component.html P ⌂ ...
1 import { Component } from '@angular/core';
2
3 @Component({
4   selector: 'app-entry-data',
5   templateUrl: './entry-data.component.html',
6   styleUrls: ['./entry-data.component.css'],
7 })
8 export class EntryDataComponent {}
```

A PRIMEIRA PARTE ESTA OK, AGORA ADICIONANDO LOGICA,
Inserir um atributo PLACE HOLDER e ele comunicará com o HTML e vice versa

Em App module

```

5  import { AppComponent } from './app.component';
6  import { HelloComponent } from './hello.component';
7  import { MenuComponent } from './components/menu.component';
8  import { Button } from './components/button.component';
9  import { EntryDataComponent } from './components/entry-data.component';
10
11 @NgModule({
12   imports: [BrowserModule, FormsModule],
13   declarations: [
14     AppComponent,
15     HelloComponent,
16     MenuComponent,
17     Button,
18     EntryDataComponent,
19   ],
20   bootstrap: [AppComponent],
21 })
22 export class AppModule {}

```

Ele dará erro se não implementar o TS antes

Em HTML, adicionar um input que é um evento HTML

<code>input</code>	Emmet Abbreviation
<code>input:b</code>	<code><input type="button" value=" "></code>
<code>input:btn</code>	
<code>input:button</code>	
<code>input:c</code>	
<code>input:checkbox</code>	

```

1 <p>E-mail Aqui</p>
2 <input type="text" value="E-mail"/>

```

Esse Input aparece na tela como uma caixa de inserção por parte do usuário

HELLO WORLD

[ADICIONAR AOS FAVORITOS](#)

[CARRINHO](#)

E-mail Aqui

No entanto isso é setado de maneira fixa no HTML

Quero que seja de maneira de ele pegar o que está no TS

```

1 <p>{{placeHolder}}</p>
2 <input type="text" [value]="placeHolder"/>

```

HELLO WORLD

[email](#)

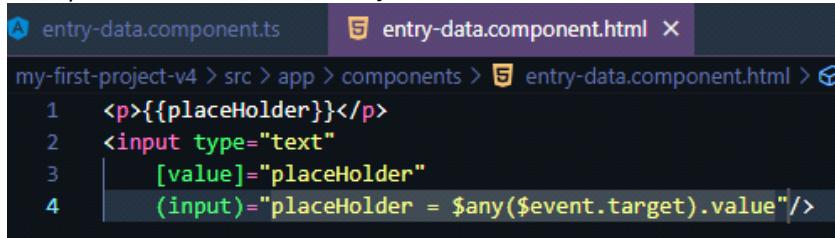
So que dessa forma como setado no HTML ele não atualizara o campo debaixo
DA TAG P, pois não tem nenhum lugar setado isso

Entao para que o TS comunique com o HTML
e ao mesmo tempo o HTML tambem mande informações para o TS

Enquanto o evento [value] puxa do TS para o HTML
O (input) comunica do Script para a parte logica

Necessario implementar dessa forma

O \$any PERMITE UMA IMPLEMENTAÇÃO JS DENTRO DO HTML



```
my-first-project-v4 > src > app > components > entry-data.component.html > ↗
1   <p>{{placeHolder}}</p>
2   <input type="text"
3     [value]="placeHolder"
4     (input)="placeHolder = $any($event.target).value"/>
```

Bom...

A primeira parte faz um EVENT BINDING pegando a comunicação com o TS dentro do HTML

A segunda parte faz um TWO WAY EVE...Bin... Gerando a comunicação em ambos os lados

O que esta acontecendo na tag P é a INTERPOLAÇÃO, onde ele pega o valor setado no TS e puxa daquela forma

mmmmmmmm :u

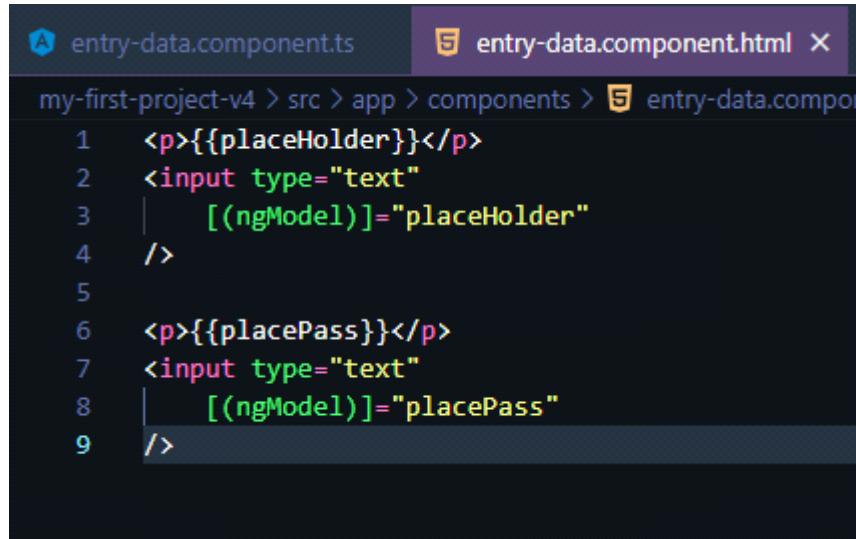
17,10,1 Two Way... Com Diretiva do Angular

sexta-feira, 25 de outubro de 2024 23:51

No passo a passo anterior

nós vimos que podemos fazer a comunicação entre 2 arquivos
sendo o TS e o HTML

agora nós vamos ver como fazer isso de forma mais fácil
utilizando uma diretiva do angular
através do NG model



The screenshot shows a code editor with two tabs: 'entry-data.component.ts' and 'entry-data.component.html'. The 'entry-data.component.html' tab is active, displaying the following code:

```
my-first-project-v4 > src > app > components > entry-data.compon
1   <p>{{placeHolder}}</p>
2   <input type="text"
3   |   [(ngModel)]="placeHolder"
4   />
5
6   <p>{{placePass}}</p>
7   <input type="text"
8   |   [(ngModel)]="placePass"
9   />
```

Dessa forma ele faz o mesmo que foi feito dentro de 17,10 Onde com aqueles scripts foram enviando e puxando a comunicação

Agora com esse ngModel, conseguimos fazer tudo em uma só

No entanto ele só é possível se vc importar aquele FORMS, itens de formulários do Angular

`import { FormsModule } from '@angular/forms';`

17,10,2 Aplicando Binding em Style (CSS)

sábado, 26 de outubro de 2024 00:29

Para aplicarmos um Style de maneira dinâmica para somente aquele atributo,

Podemos fazer isso de maneira dinâmica, dentro do HTML que puxara a definição através do TS

COMO O **ENTRY DATA HTML** É COMO SE FOSSE UM IRMAO OU FILHO DO **ENTRY DATA TS**, ELE PODE UTILIZAR DOS ELEMENTOS DO TS A QUALQUER HORA E LUGAR, NÃO NECESSARIAMENTE DENTRO DO ATRIBUTO x

DENTRO DO TS

```
export class EntryDataComponent{
  placeHolder:string = "Email";
  placePass:string = "Senha";
  color:string="red";
}
```

DENTRO DO HTML

Note que ele esta chamando em um lugar nada verrr, DENTRO DA TAG P

```
<p [style.color]="color">{{placePass}}</p>
<input type="text"
  [(ngModel)]="placePass"
/>
```

Dessa forma só o SENHA mudou, mas o restante esta normal

HELLO WORLD

ADICIONAR AOS FAVORITOS
CARRINHO

Email

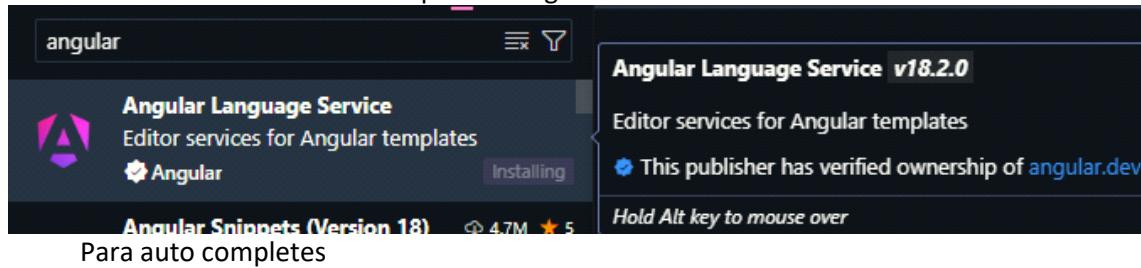
Senha

17,1,1 Plugin Necessário para Programação no VSCode

sábado, 26 de outubro de 2024

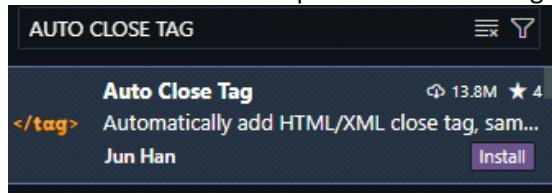
00:49

ANGULAR LANGUAGE SERVICE Proprio do Angular



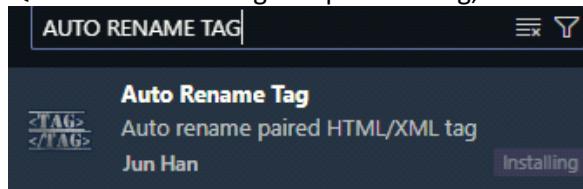
AUTO CLOSE TAG

Para automaticamente quando abre uma tag ele automaticamente a fecha



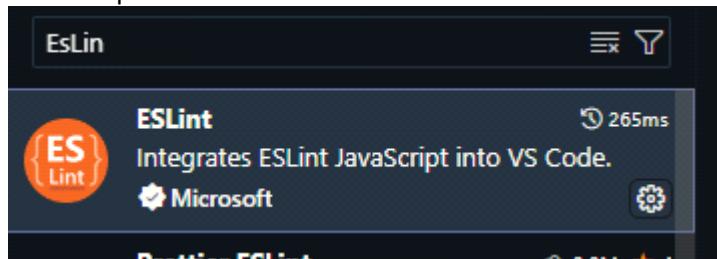
AUTO RENAME TAG

Quando renomeio algo da primeira tag, ele automaticamente renomeia a de fechamento



ESLint

Melhora práticas de Boas Práticas



PRETTIER

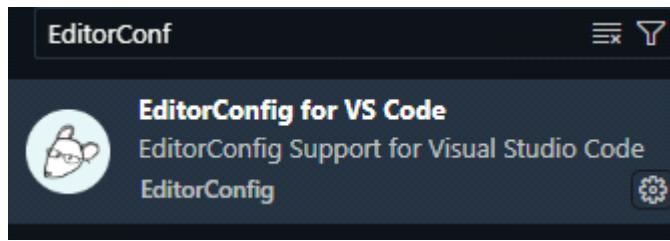
Um formatador de código



EDITOR CONFIG

Ajuda na boa prática da programação

Quando acordar um tab ou quando vai colocar aspas simples ou aspas duplos pra sempre manter uma regra padrão definir o que deve ou não deve ser feito



AURA TEMA E O DRACULA THEME

Tema de sua escolha

ICON

Alguma de ICONES para se achar mais facil na distinção de arquivos HTML/CSS/JS/TS/NODE/NG...

O BOOTCAMP DEAL - SPRINGBOOT COM ANGULAR

sábado, 26 de outubro de 2024

00:49

17,11 Ciclo de componente

segunda-feira, 28 de outubro de 2024 22:04

Criando um componente com NG GENERATE

Entendendo como funciona

Ele é um comando para criar estrutura de código

Para saber o que podemos criar --help

NG GENERATE --HELP

>>>

Commands:

ng generate <schematic>	Run the provided schematic. [default]
ng generate app-shell	Generates an application shell for running a server-side version of an app.
ng generate application [name]	Generates a new basic application definition in the "projects" subfolder of the workspace. [aliases: app]
ng generate class [name]	Creates a new, generic class definition in the given project. [aliases: cl]
ng generate component [name]	Creates a new, generic component definition in the given project. [aliases: c]
ng generate config [type]	Generates a configuration file in the given project.
ng generate directive [name]	Creates a new, generic directive definition in the given project. [aliases: d]
ng generate enum [name]	Generates a new, generic enum definition in the given project. [aliases: e]
ng generate environments	Generates and configures environment files for a project.
ng generate guard [name]	Generates a new, generic route guard definition in the given project. [aliases: g]
ng generate interceptor [name]	Creates a new, generic interceptor definition in the given project.
ng generate interface [name] [type]	Creates a new, generic interface definition in the given project. [aliases: i]
ng generate library [name]	Creates a new, generic library project in the current workspace. [aliases: lib]
ng generate module [name]	Creates a new, generic NgModule definition in the given project. [aliases: m]
ng generate pipe [name]	Creates a new, generic pipe definition in the given project. [aliases: p]
ng generate resolver [name]	Generates a new, generic resolver definition in the given project. [aliases: r]
ng generate service [name]	Creates a new, generic service definition in the given project. [aliases: s]
ng generate service-worker	Pass this schematic to the "run" command to create a service worker
ng generate web-worker [name]	Creates a new, generic web worker definition in the given project.

Arguments:

schematic The [collection:schematic] to run.
[string]

```

Options:
  --help      Shows a help message for this command in the console.
  [boolean]
    --interactive  Enable interactive input prompts.          [boolean]
  [default: true]
    -d, --dry-run   Run through and reports activity without writing out results.
  [boolean] [default: false]
    active input prompts for options with a default.        [boolean] [default: false]  --
  force      Force overwriting of existing files.

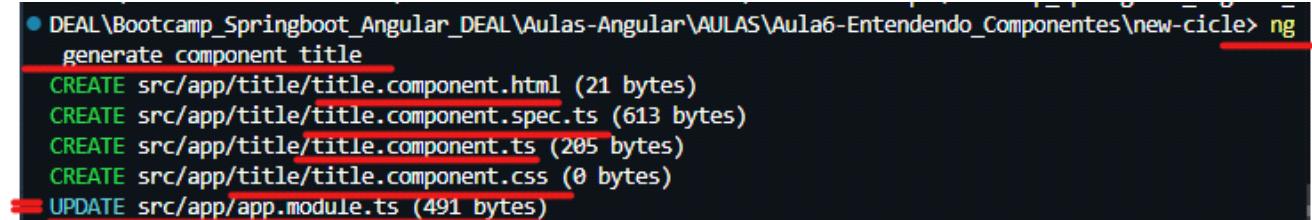
```

<<<

LEMBRANDO QUE NÃO PRECISA ESTAR DENTRO DE APP OU DE SRC PARA DAR UM GENERATE
Por ser algo padrao, ele faz isso de forma automatica

Entao...

== NG GENERATE COMPONENT 'NOME'
ou com ALIAS == NG G C 'NOME'

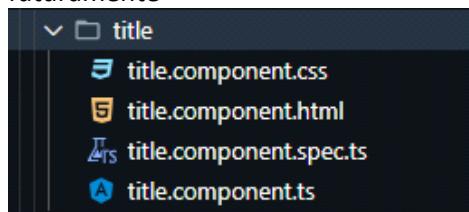
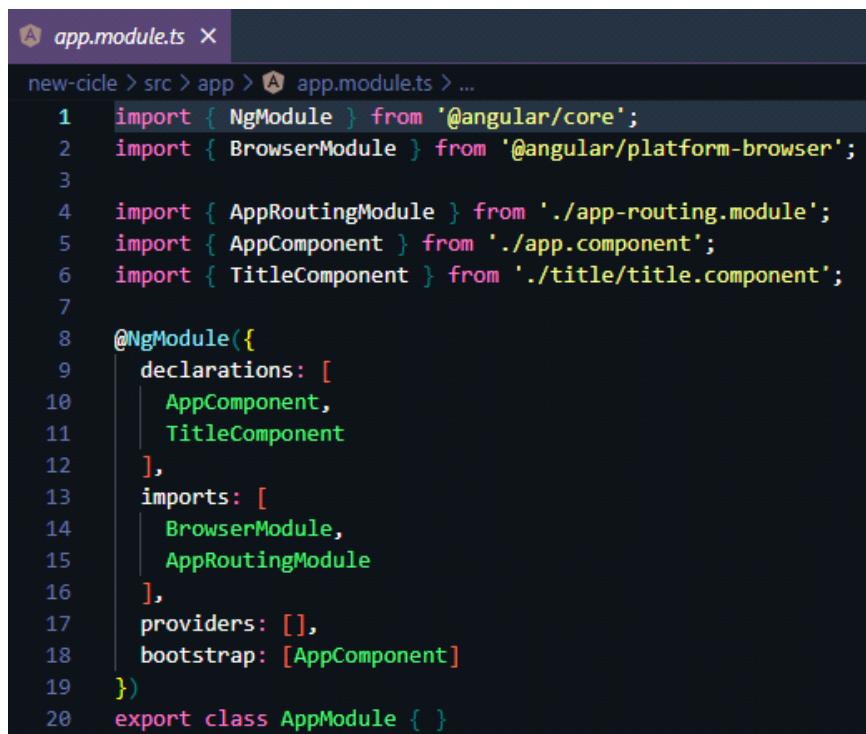


```

DEAL\Bootcamp_Springboot_Angular_DEAL\Aulas-Angular\AULAS\Aula6-Entendendo_Componentes\new-cicle> ng generate component title
CREATE src/app/title/title.component.html (21 bytes)
CREATE src/app/title/title.component.spec.ts (613 bytes)
CREATE src/app/title/title.component.ts (205 bytes)
CREATE src/app/title/title.component.css (0 bytes)
UPDATE src/app/app.module.ts (491 bytes)

```

Note que dessa forma, não precisa criar na mao o modulo e as classes,
ele ate cria um SPEC com alguns comandos e funções especiais que serão vistas
futuramente

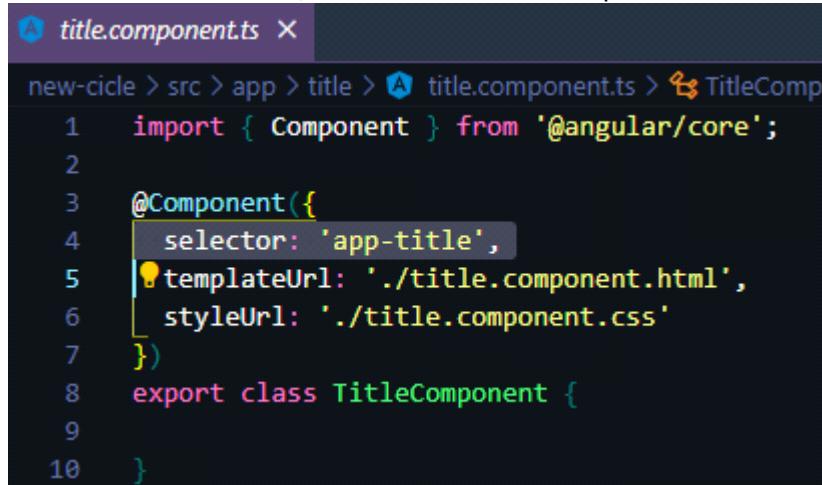



```

new-cicle > src > app > app.module.ts > ...
1 import { NgModule } from '@angular/core';
2 import { BrowserModule } from '@angular/platform-browser';
3
4 import { AppRoutingModule } from './app-routing.module';
5 import { AppComponent } from './app.component';
6 import { TitleComponent } from './title/title.component';
7
8 @NgModule({
9   declarations: [
10     AppComponent,
11     TitleComponent
12   ],
13   imports: [
14     BrowserModule,
15     AppRoutingModule
16   ],
17   providers: [],
18   bootstrap: [AppComponent]
19 })
20 export class AppModule { }

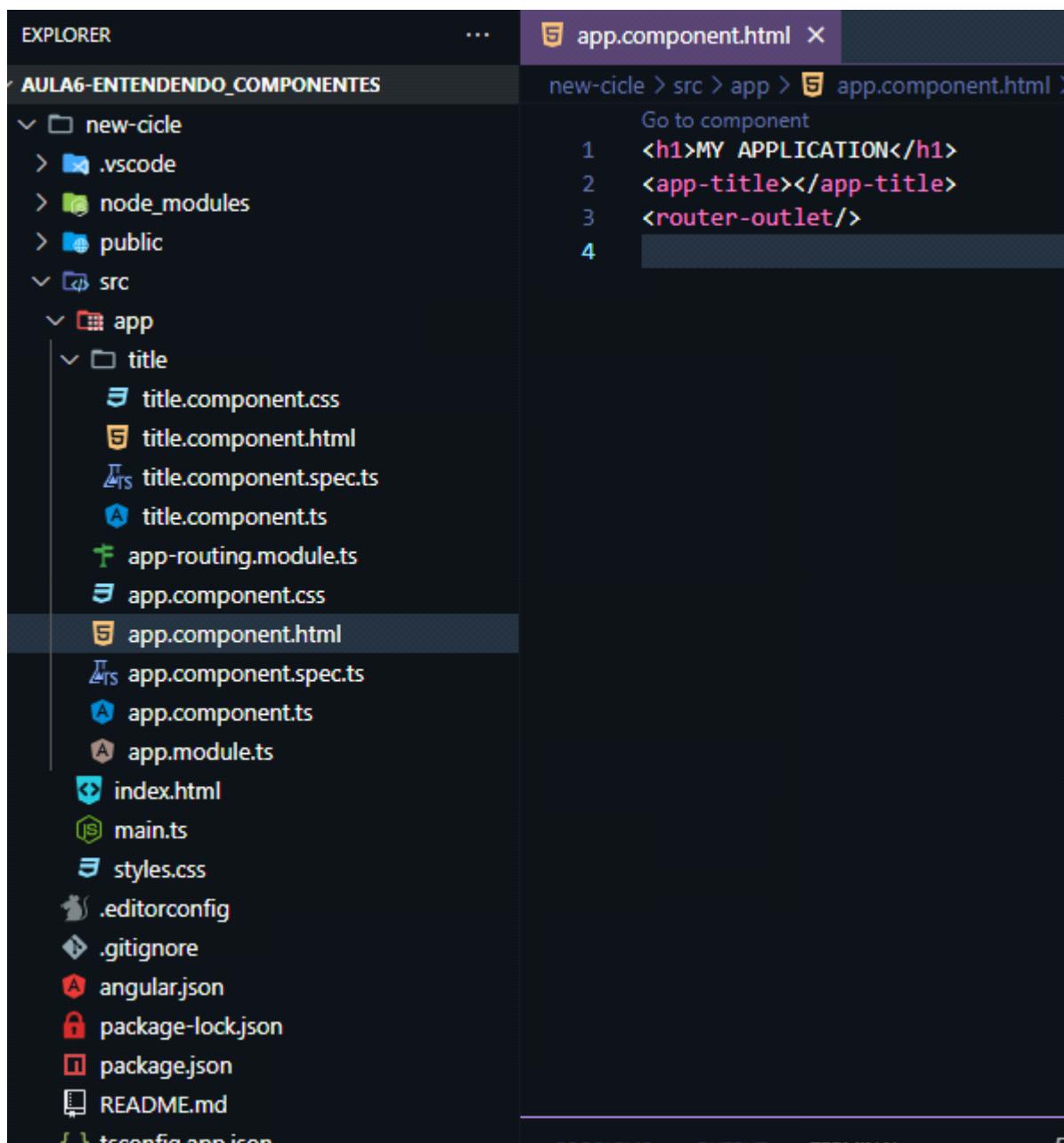
```

Para chamarmos o TITLE, vamos ver como esta implementado dentro de title...ts



```
1 import { Component } from '@angular/core';
2
3 @Component({
4   selector: 'app-title',
5   templateUrl: './title.component.html',
6   styleUrls: ['./title.component.css']
7 })
8 export class TitleComponent {
9
10 }
```

No html geral

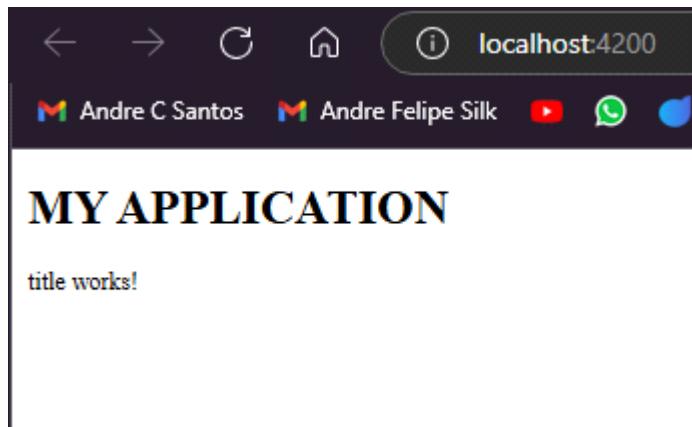


EXPLORER ...

AULA6-ENTENDENDO_COMPONENTES

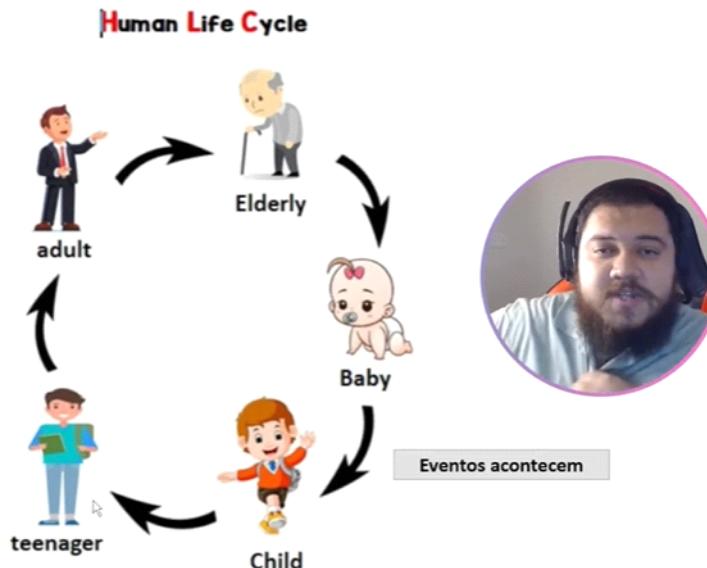
- new-cicle
 - .vscode
 - node_modules
 - public
- src
 - app
 - title
 - title.component.css
 - title.component.html
 - title.component.spec.ts
 - title.component.ts
 - app-routing.module.ts
 - app.component.css
 - app.component.html
 - app.component.spec.ts
 - app.component.ts
 - app.module.ts
 - index.html
 - main.ts
 - styles.css
 - .editorconfig
 - .gitignore
 - angular.json
 - package-lock.json
 - package.json
 - README.md
 - tsconfig.app.json

Rodando vai chamar ok o sistema



17,11,1 Entendendo Ciclo de Vida

segunda-feira, 28 de outubro de 2024 22:47



Entendemos que sempre em um ciclo de vida,
Muitos eventos acontecem

O que diferencia um bb de uma criança, de um jovem...

O componente é a mesma coisa

Quando criado, ele é um BB, ele passa por modificações até ser destruído "morto"

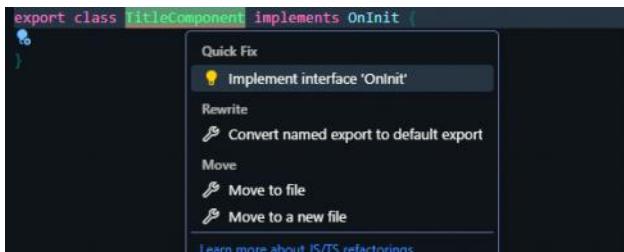
A captura de tela mostra o código de um componente Angular chamado "title.component.ts". O código importa o "Component" decorador e o "OnChanges" interface. A interface OnChanges é expandida para mostrar suas implementações: OnInit, OnDestroy, ngOnChangesFeature, Optional, OptionalDecorator, eedeforOnHover, eedeforOnIdle, eedeforOnImmediate, eedeforOnInteraction, eedeforOnTimer e eedeforOnViewport. À direita do código, uma janela de documentação para a interface OnChanges é aberta, fornecendo detalhes sobre sua função, exemplos e outras interfaces relacionadas.

Para criação de Evento

Utilizar o ONINIT

Esse comando faz uma ação sempre que o componente for iniciado

Seja listar algo
Seja iniciar outra ação



Entao

```
export class TitleComponent implements OnInit {
    ngOnInit(): void {
        throw new Error('Method not implemented.');
    }
}
```

17,12,1 Ciclo OnInit

terça-feira, 29 de outubro de 2024 21:35

Agora fazendo testes dentro do OnInit

```
export class TitleComponent implements OnInit {
  nome:string = 'Andre';

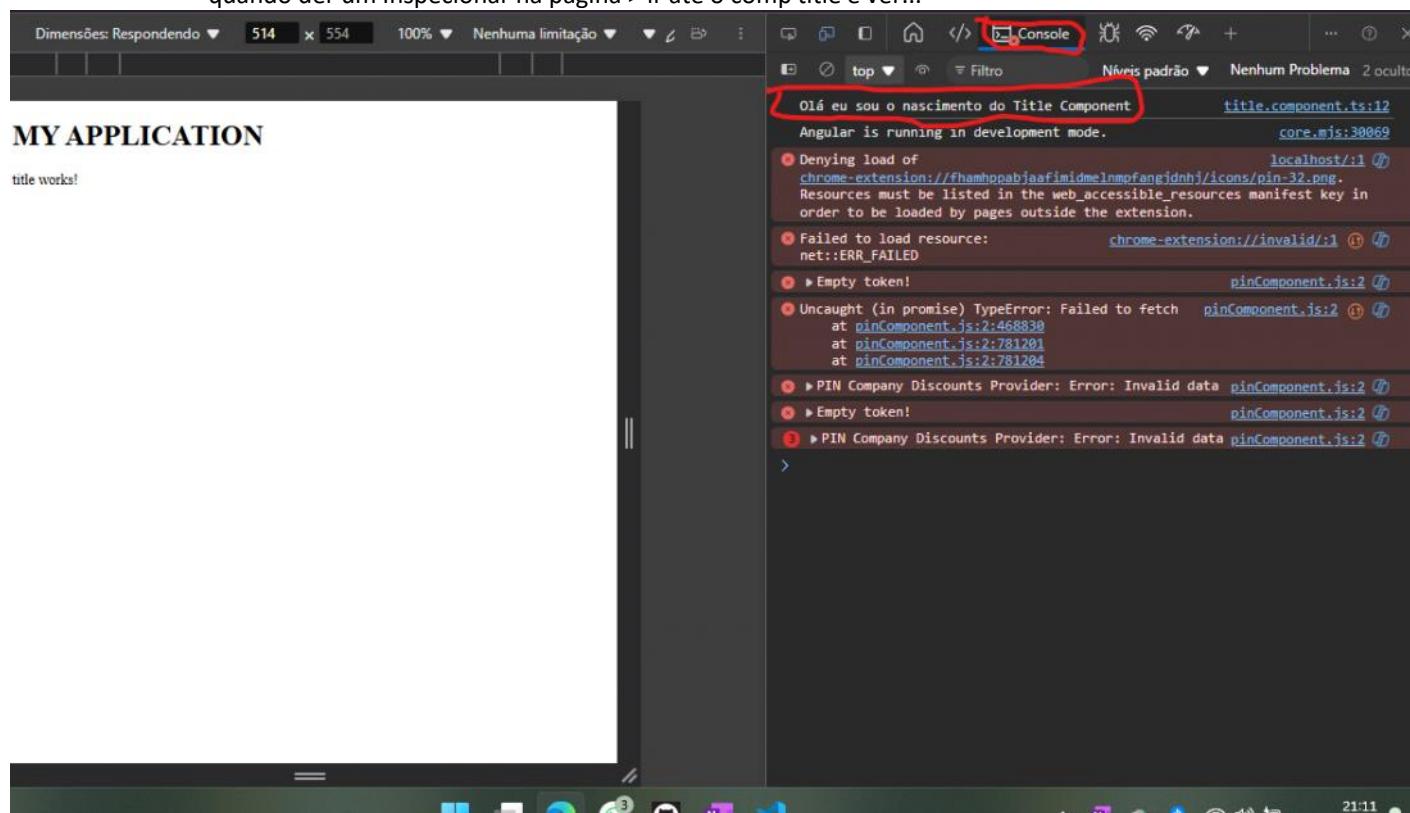
  ngOnInit(): void {
    console.log('Olá eu sou o nascimento do Title Component');

    this.nome = `Olá ${this.nome}`;
  }
}
```

Vamos atribuir algo lógico nele

Se somente implementar assim,

ele vai aparecer erro no This.nome pois não está definindo nada em nome, e o log
vai aparecer normal
quando der um Inspecionar na pagina > ir ate o comp title e ver...



Alem de fazer itens ocultos conseguimos fazer como já vimos, como por exemplo

Selecionar sua Cidade, fazendo um componente que só apareça quando vc for clicar para selecionar esse campo,

Ou aperfeiçoando. Criar um que forneça essa funcionalidade, mas na hora da criação disso vc diz que é de Cidade, Estado, Animal, Gênero, e assim vai

Se eu quero fazer ele aparecer na tela alguma variável, vamos chama-la dentro do HTML do comp.title...



```
new-cicle > src > app > title > title.component.html > p
    Go to component
1   <!-- <p>title works!</p> -->
2   <p>{{nome}}</p>
```

No app component é simplesmente a mesma chamada de metodo

17,12 LifeCicles

terça-feira, 29 de outubro de 2024 21:51

Dentro tanto do REACT OU DO ANGULAR
Components em geral

Life Cicles Hooks
Eventos de ciclos de vidas

Dentro do Angular, temos alguns como...:

1. ggOnInit --> Toda vez que o comp é iniciado
2. ngOnChange --> Toda vez que ocorre uma modificação no comp
3. ngDoCheck --> Eventos que devem ser seguidos dentro do Check
4. ngOnDestroy --> Sempre que o ciclo dele acaba, como um "close componente"



EM ANGULAR IO LIFE CICLE ELE TEM A DOCUMENTAÇÃO QUE EXPLICA E MOSTRA COMO IMPLEMENTAR CADA UM DELES

17,12,2 Ciclo DoCheck

terça-feira, 29 de outubro de 2024 21:58

Dentro do DoCheck temos 4 estações

Toda vez que é feito uma modificação, é realizado um check

Depois que o componente da um INIT

```
//executa quando Angular realiza qualquer projeção de conteúdo em seus componentes  
3.1 ngAfterContentInit()
```

Depois que o componente é verificado pelo Angular se ouve alguma manipulação

```
//executa sempre que conteúdo do componente é verificado pelo mecanismo de detecção de  
alteração do Angular  
3.2 ngAfterContentChecked()
```

Depois de carregado tudo, se o CSS esta ok, se o componente esta tudo funcionando

```
//Executa depois que um componente é totalmente inicializada (carregou tudo)  
3.3 ngAfterViewInit()
```

Verificado depois da visualização de alteração do Angular

```
//Executa quando a visualização de um componente é verificada pelo algoritmo de detecção de alterações do  
Angular  
3.4 ngAfterViewChecked()
```

DEU PRA VER QUE NÃO FICOU MUITO BEM EXPLICADO, MAS...

PARA TESTE VAMOS CRIAR UM NOVO COMPONENTE

Ng generate component check-sample

Como já visto a configuração que ele cria

```
PS D:\conta.ACER-ANDRE-SANT\Ambiente de Desenvolvimento\DIOS - Bootcamps\Bootcamp_Springboot_Angular_DEAL\Bootcamp_Springboot_Angular_DEAL\Aulas-Angular\AULAS\Aula6-Entendendo_Componentes\new-cicle> ng g c check-sample  
CREATE src/app/check-sample/check-sample.component.html (28 bytes)  
CREATE src/app/check-sample/check-sample.component.spec.ts (656 bytes)  
CREATE src/app/check-sample/check-sample.component.ts (232 bytes)  
CREATE src/app/check-sample/check-sample.component.css (0 bytes)  
UPDATE src/app/app.module.ts (597 bytes)
```

Implementando dentro do html geral (app.component.html)

VAMOS A ALTERAÇÃO DO COMPONENTE

PRIMEIRO EM TS

VAMOS ADICIONAR O INIT E O DOCHECK(1.2.3.4)

Importando todos os contratos...

```
import { Component, OnInit, DoCheck, AfterContentChecked, AfterContentInit, AfterViewChecked, AfterViewInit } from '@angular/core';
```

Importar eles com implements

LEMBRANDO QUE O IMPLEMENTS PRECISA IMPLEMENTAR TODAS AS FUNCOES, ALTERANDO
O CONTEUDO DAS FUNCOES

The screenshot shows a code editor with a class definition:

```
export class CheckSampleComponent implements OnInit, DoCheck, AfterContentChecked, AfterContentInit, AfterViewChecked, AfterViewInit {  
}  
}
```

A context menu titled "Quick Fix" is open over the class body, listing the following suggestions:

- Implement interface 'OnInit'
- Implement interface 'DoCheck'
- Implement interface 'AfterContentChecked'
- Implement interface 'AfterContentInit'
- Implement interface 'AfterViewChecked'
- Implement interface 'AfterViewInit'
- Implement all unimplemented interfaces

The code editor shows several methods defined with a throw statement:

```
export class CheckSampleComponent implements  
OnInit,  
DoCheck,  
AfterContentChecked,  
AfterContentInit,  
AfterViewChecked,  
AfterViewInit []  
  
ngAfterViewInit(): void {  
| throw new Error('Method not implemented.');  
}  
ngAfterViewChecked(): void {  
| throw new Error('Method not implemented.');  
}  
ngAfterContentInit(): void {  
| throw new Error('Method not implemented.');  
}  
ngAfterContentChecked(): void {  
| throw new Error('Method not implemented.');  
}  
ngDoCheck(): void {  
| throw new Error('Method not implemented.');  
}  
ngOnInit(): void {  
| throw new Error('Method not implemented.');  
}  
}
```

Mesmo implementando, ele vai dar erro pois eles ainda não estao implementados
"Throw new Error" basicamente força ele a dar erro pois não está implementado

17,12,2,1 DoCheck na pratica

terça-feira, 29 de outubro de 2024 22:24

Para visualizarmos melhor, vamos ver na pratica, criando alguns elementos

Para esse cado, vamos criar um Button onde a cada click ele adiciona ou diminui o valor da quantidade

Como já foi tudo pre montado pelo Angular ALIAS ng
Vamos focar na pimplementação do Component

EM TS

já adicionamos o essencial, agora o metodo de Adicionar e Decrementar o elemento QUANTIDADE

```
export class CheckSampleComponent implements  
OnInit,  
DoCheck,  
AfterContentChecked,  
AfterContentInit,  
AfterViewChecked,  
AfterViewInit {  
  
    quantidade : number = 0;  
  
    adicionar(){  
        this.quantidade += 1;  
    }  
    decrementar(){  
        this.quantidade -= 1;  
    }  
}
```

EM HTML

```
<button (click)="decrementar()"> </button>  
  
&{{quantidade}}  
  
<button (click)="adicionar()"> > </button>
```

NO HTML DO GERAL

```
<app-check-sample/>
```

Dessa forma, quandi realizado o essas implementações, ele aparece dessa forma no html

check-sample works!

< 7 >

Ignore o erro PIN pois vem de um plugin do meu navegador

Como clicado 7 vezes, ele aparece as LISTAS DE CHECKADOS 7 VEZES DE CADA UM DOS CHECK-LIST

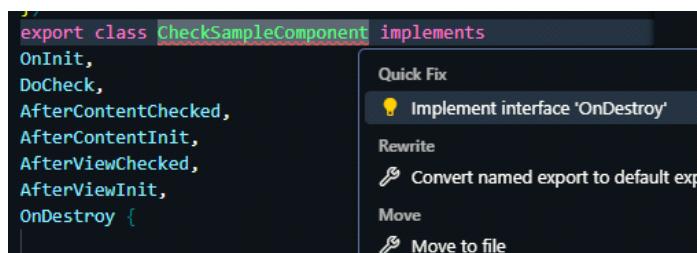
LEMBRANDO QUE
PRIMEIRO
A Checagem
SEGUNDO
O Conteúdo
TERCEIRO
A Visualização

17,12,2,2 Ciclo OnDestroy

terça-feira, 29 de outubro de 2024 23:07

Primeiramente quando chamado o mais um ciclo, necessário a import dele e sua implementação na classe

```
import {  
  Component,  
  OnInit,  
  DoCheck,  
  AfterContentChecked,  
  AfterContentInit,  
  AfterViewChecked,  
  AfterViewInit,  
  OnDestroy  
} from '@angular/core';
```



Implementar ele agora

```
ngOnDestroy(): void { /*QUANDO ELE MORRER, FAZER ISSO*/  
  console.log("Good Bye My Friend");  
}
```

Essa implementação define como o comp vai morrer, vai se encerrar

Como igual em um Scanner que da um scan.close();

É basicamente isso

Para definir o close/destroy

Em app.component.html , nosso HTML geral

Na chamada temos alguns métodos que futuramente iremos aprender, como NG IF

Esse nglf é por padrão == true,
então ter ele como true
ou não ter é a mesma coisa

ANTES DISSO

Vamos em geral comp... TS

Em AppComponent.TS

```
check-sample.component.ts app.component.ts AppComponent  
new-cicle > src > app > app.component.ts > AppComponent  
1 import { Component } from '@angular/core';  
2  
3 @Component({  
4   selector: 'app-root',  
5   templateUrl: './app.component.html',  
6   styleUrls: ['./app.component.css'  
7 })  
8 export class AppComponent {  
9   title = 'new-cicle';  
10  isAliveCheckSample: boolean = true;
```

Definir uma variável como true, enquanto ele estiver vivo

```
export class AppComponent {  
  title = 'new-cicle';  
  isAliveCheckSample: boolean = true;
```

Então... isAlive+NomeDoComponente === true/false;

Em AppComponent.HTML

```
<app-check-sample *ngIf="isAliveCheckSample">
```

Na chamada do componente, vamos atribuir as funções ng...

Nesse caso ngIf...

```
<app-check-sample *ngIf="isAliveCheckSample">
```

<app-check-sample *ngIf="isAliveCheckSample"/>

Lembrando que, para morrer, precisa estar vivo, entao, marcado o elemento como TRUE

Agora que implementado isso, vamos fazer ele ter um metodo que o matará

Em APP.COMPONENT.TS

O TS Geral do projeto

Vamos criar um metodo "*disposeCheckSample*"

```
1 import { Component } from '@angular/core'
2
3 @Component({
4   selector: 'app-root',
5   templateUrl: './app.component.html',
6   styleUrls: ['./app.component.css']
7 })
8 export class AppComponent {
9   title = 'new-cicle';
10  isAliveCheckSample:boolean = true;
11
12  disposeCheckSample():void{
13    this.isAliveCheckSample=false;
14  }
15 }
```

Esse metodo esta "matando o componente"

Para implementação dele precisamos chamar o que vai executar essa ação

Ou seja, o html geral

Em APP.COMPONENT.TS

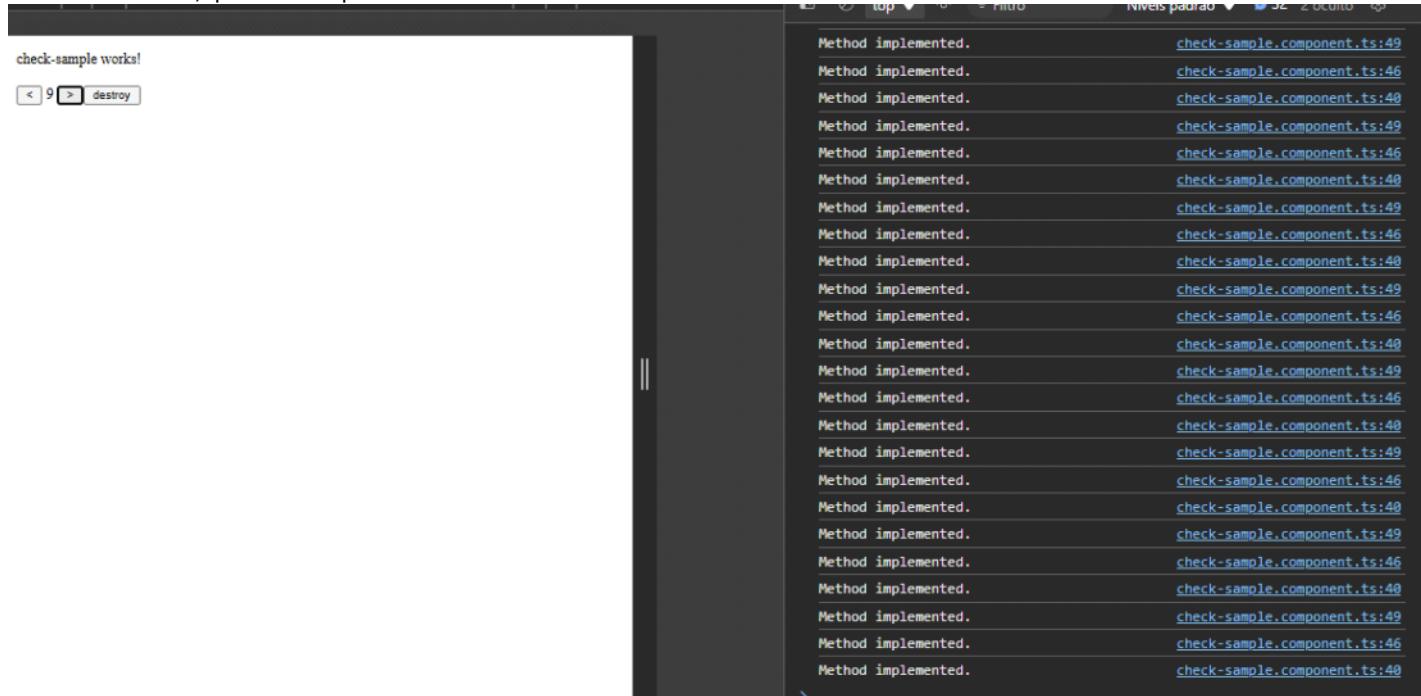
The screenshot shows a browser developer tools component editor interface. The top navigation bar has tabs for 'app.component.ts' and 'app.component.html'. Below the tabs, a breadcrumb navigation shows the path: 'new-cicle > src > app > app.component.html > button'. The main content area displays the following code:

```
1  <!-- <h1>MY APPLICATION</h1>
2  <app-title></app-title> -->
3  <app-check-sample *ngIf="isAliveCheckSample"/>
4  <button (click)="disposeCheckSample()">destroy</button>
5  <router-outlet/>
6
```

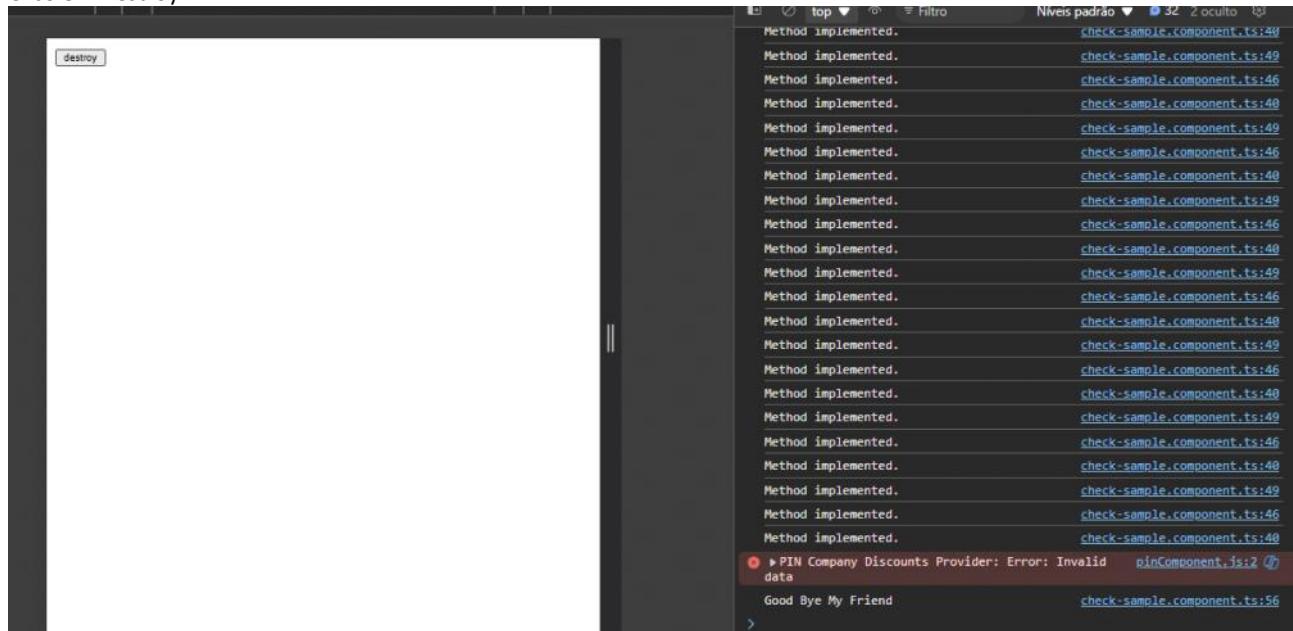
NÃO ESQUECA DOS PARENTESES "()", SEM ELE DA ERRO, EXP PROPRIA x_x

No Servidor

Dessa forma, quando manipulo...



Clico em Destroy



Ele simplesmente some a opção de manipulação e guarda as informações

Isso é super importante, pois pensa se voce estiver com varios componentes, sempre iniciando a

cada etapa do projeto, e nunca fecha

Vai sobreregar tudo,

ENTAO FECHEEEE OS COMPONENTES, MATE OS FILHOS DOS SEUS FILHOS KKKKKKKK

ENCERRE CICLOS QUE PRECISAM SER ENCERRADOS

17,13 DIRETIVAS DO ANGULAR

quarta-feira, 30 de outubro de 2024 22:09

Quando mexemos com componentes dinâmicos e DOM e Shadow DOM

Diretivas

Diretiva são a maneira como o angular manipula e altera a DOM dinamicamente

Diretivas de atributo

Alteram a aparência ou comportamento de um elemento, componente ou outra diretiva

- NgClass - Adiciona ou remove conjunto de classes CSS
- NgStyle - Adiciona ou remove um conjunto de estilos ao HTML
- NgModel - Adiciona vinculação de dados bidirecional a um elemento de um formulário

Diretivas estruturais

Moldam ou remodelam a estrutura da DOM, adicionando ou removendo elementos na tela

- Nglf - Condicional que verifica se modelo deve ser visualizado ou não
- NgFor - repete um elemento para cada item em uma lista
- NgSwitch - Utilizado para alternar entre comportamentos alternativos

Mostrar ocultar elementos

Temos 2 diretivas

Atributos -

Alteram informações de um componente específico ou de outro atributo

Tipo uma manipulação, como um modo escuro na tela, onde se o usuário clicar em

Modo Escuro ele fica dark senão,

e se clicar em desativar Ele fica White

Estruturais

Alteram a estrutura da DOM,

Com o nglf verificando se o modelo está sendo visualizado ou não

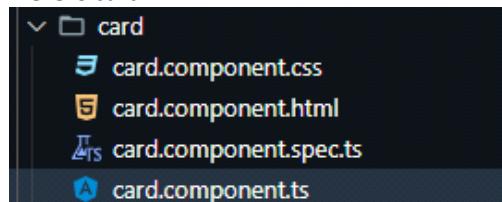
If - For - Switch em lógica de Angular ng...

17,13,1 Diretivas de Estruturais

quinta-feira, 31 de outubro de 2024 20:37

Criando um novo componente

NG G C card



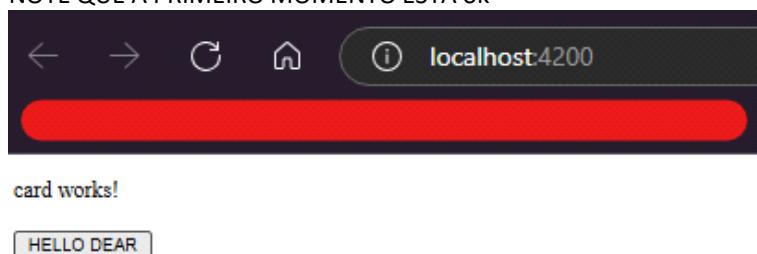
NG IF

Basicamente é um *ngIf

Ele é um valor booleano de verdadeiro e falso

Onde se ele estiver definido como false, ele não aparecerá na página do servidor

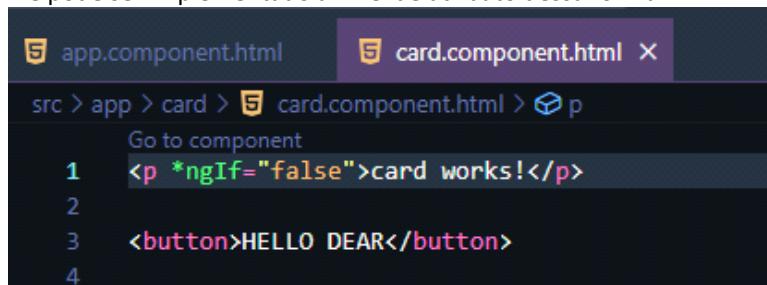
NOTE QUE A PRIMEIRO MOMENTO ESTA OK



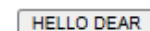
Todas as condições ou diretivas como chamado aqui,
Podem ser feitas a nível de componente ou atributo

A nível de atributo

Ele pode ser implementado a nível de atributo dessa forma



NOTE QUE FAZENDO ISSO...

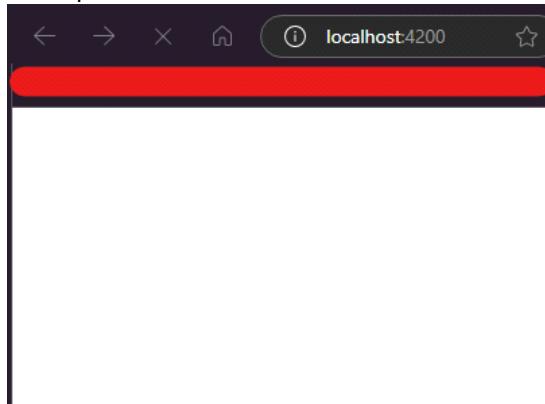


Ele ocultou o p com o dizer Card Works

A nível de componente, podemos implementá-lo dessa forma

The screenshot shows the Angular documentation for the `NgIf` directive. It defines it as a structural directive that conditionally includes a template based on the value of an expression. The code example shows the directive being used with the syntax `<app-card *ngIf=>`.

Note que dessa forma some todo conteúdo do serv



DESSA FORMA CONSEGUIMOS ENTENDER QUE O IF SERVE PARA DIZER QUE (COMO DEFAULT) O TRUE EXIBE O ELEMENTO NA TELA, COMO FALSE (TEM QUE IMPLEMENTAR) ELE OCULTA ELEMENTO OU COMPONENTE

NG IF COM DATA BINDING

Com data binding, podemos amarrar ele a uma classe

Combinando o ng if com a classe, conseguimos fazer dessa forma

```

c > app > A app.component.ts > AppComponent > isAliveCard
1 import { Component } from '@angular/core';
2
3 @Component({
4   selector: 'app-root',
5   templateUrl: './app.component.html',
6   styleUrls: ['./app.component.css']
7 })
8 export class AppComponent {
9   title = 'diretivas';
10  isAliveCard = true;
11 }
12

```

DENTRO DE APP COMPONENT HTML

Podemos implementá-lo dessa forma, mas só funciona isso a nível de componente, até onde eu consegui testar

```

src > app > app.component.html > app-card
      Go to component
1 <app-card *ngIf="isAliveCard"/>
2 <router-outlet />

```

NG IF NG TEMPLATE

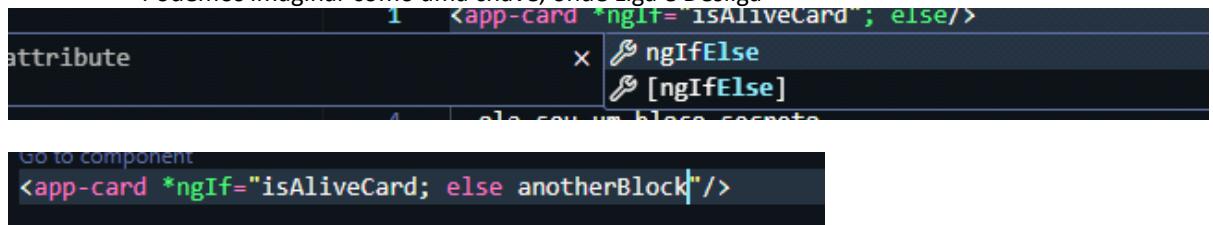
Podemos atrelar o `ngIf` com blocos condicionais

Esses blocos podem ser no componente html geral

COMO IF é uma condicional, podemos utilizar com ele o ELSE

Dessa forma, quando If e Else são implementados,

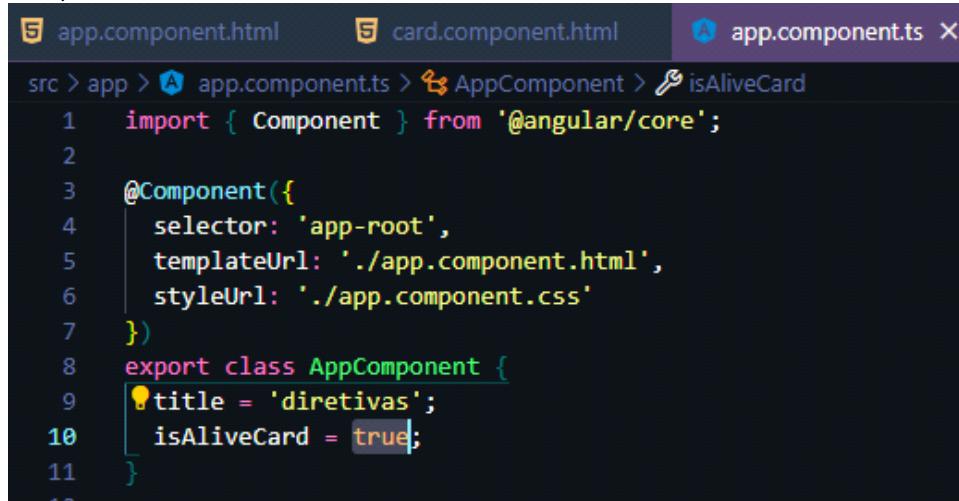
Podemos imaginar como uma chave, onde Liga e Desliga



The screenshot shows an IDE interface with three tabs: 'app.component.html', 'card.component.html', and 'app.component.ts'. The 'app.component.html' tab has the following code:
<app-card *ngIf="isAliveCard; else anotherBlock"/>
A tooltip from the IDE's code completion feature is displayed above the code, showing two options: 'ngIfElse' and '[ngIfElse]'. The 'ngIfElse' option is highlighted with a blue border.

NOTE QUE TODA ESSA ESTRUTURA ESTA DE4NTRO DOS PARENTESES

Se a primeira estrutura for TRUE,



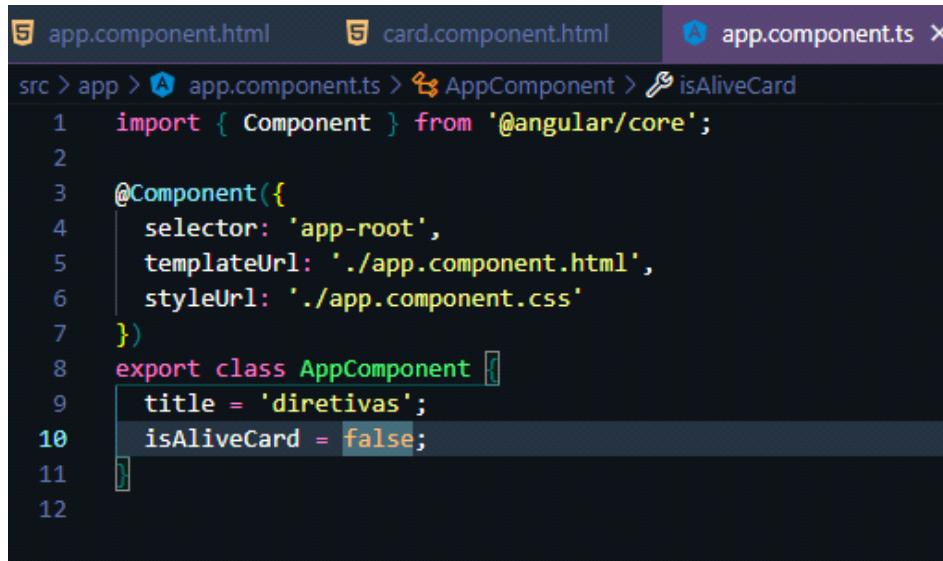
The screenshot shows the 'app.component.ts' file with the following code:
import { Component } from '@angular/core';

@Component({
 selector: 'app-root',
 templateUrl: './app.component.html',
 styleUrls: ['./app.component.css']
})
export class AppComponent {
 title = 'diretivas';
 isAliveCard = true;
}
The line 'isAliveCard = true;' is highlighted with a blue selection bar.

Aparece Isso:



Se for FALSE



The screenshot shows the 'app.component.ts' file with the following code:
import { Component } from '@angular/core';

@Component({
 selector: 'app-root',
 templateUrl: './app.component.html',
 styleUrls: ['./app.component.css']
})
export class AppComponent {
 title = 'diretivas';
 isAliveCard = false;
}
The line 'isAliveCard = false;' is highlighted with a blue selection bar.

Aparece Isso:

ola sou um bloco secreto

AGORA VAMOS VOLTAR PARA TRUE

NG FOR

Utilizado para gerar informações de forma autônoma

Vamos criar um array de produtos, onde ele "pegara de uma API imaginária" as seguintes informações

```
export class CardComponent implements OnInit {
    /*ARRAY DE PRODUTOS*/
    produtos:string[] = []

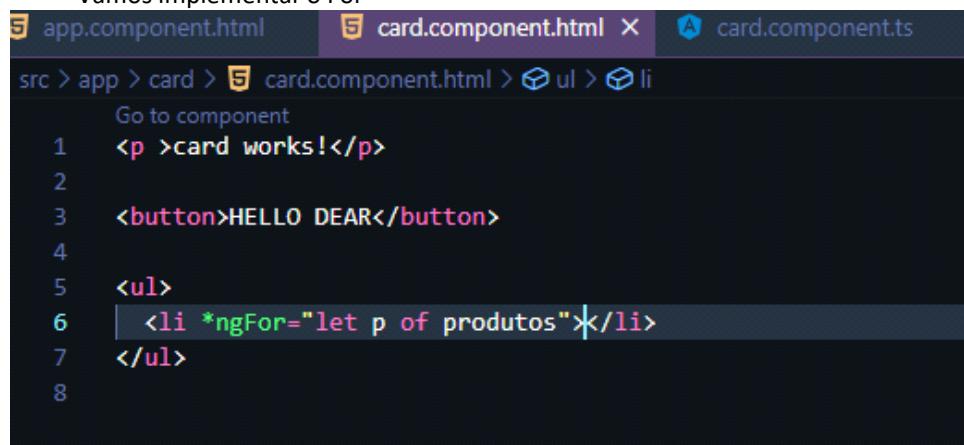
    constructor(){
        // ELE FARÁ A REQUISIÇÃO NA API,
        // E RETORNARÁ COM ESSAS INFORMAÇÕES
        this.produtos =[]
            ["Mouse", "Teclado", "Cabo", "Fonte"]
        ]
    }
    ngOnInit(): void {
    }

}
```

Já falamos para que serve o OnInit

Agora para fazer ele aparecer de maneira dinâmica,

Vamos implementar o For



```
src > app > card > card.component.html > ul > li
      Go to component
1   <p>card works!</p>
2
3   <button>HELLO DEAR</button>
4
5   <ul>
6     <li *ngFor="let p of produtos">{{ p }}</li>
7   </ul>
8
```

FAZENDO DESSA FORMA, ACREDITO QUE SE ENCAIXE MAIS COMO FOR EACH DO QUE
COMO FOR

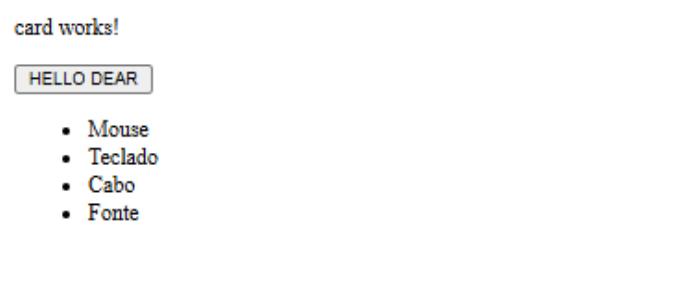
No entanto, dessa forma, ele está percorrendo a P dentro de PRODUTOS

Para melhorar ainda mais, podemos fazer uma interpolação,

Onde ele retornará um elemento p de cada um

```
<ul>
| <li *ngFor="let p of produtos">{{ p }}</li>
</ul>
```

NO SERVIDOR



REATIVIDADE

Reagir aquilo que ele está conectado

Dessa forma

LEMBRA QUE FOI IMPLEMENTADO O PRODUTO COM MOUSE, TECLADO...

card component HTML

Vamos criar um método dentro de card component HTML

```
<ul>
  <li *ngFor="let p of produtos">{{ p }}</li>
  <button (click)="Adicionar()"></button>
</ul>
```

Como ainda não implementado, está com erro

Vamos implementar esse método dentro do nosso componente utilizado nessa parte de DIRETIVAS == "CARD"

```
src > app > card > A card.component.ts > CardComponent
1
2   @Component({
3     selector: 'app-card',
4     templateUrl: './card.component.html',
5     styleUrls: ['./card.component.css']
6   })
7   export class CardComponent implements OnInit {
8     /*ARRAY DE PRODUTOS*/
9     produtos: string[] = []
10
11    constructor(){
12      // ELE FARÁ A REQUISIÇÃO NA API,
13      // E RETORNARÁ COM ESSAS INFORMAÇÕES
14      this.produtos = [
15        "Mouse", "Teclado", "Cabo", "Fonte"
16      ]
17    }
18    ngOnInit(): void {
19    }
20
21    adicionar(){
22      |
23    }
24  }
25
26
```

VAMOS IMPLEMENTAR de uma maneira Statica só para estudo

```
adicionar(){
  this.produtos.push("André"); /**ADICIONA UM ELEMENTO */
}
```

Em TS DO CARD

Implementando 2 métodos

```
src > app > card > A card.component.ts > CardComponent > constructor
  2
  3   @Component({
  4     selector: 'app-card',
  5     templateUrl: './card.component.html',
  6     styleUrls: ['./card.component.css']
  7   })
  8   export class CardComponent implements OnInit {
  9     /*ARRAY DE PRODUTOS*/
10     produtos:string[] = []
11
12     constructor(){
13       // ELE FARÁ A REQUISIÇÃO NA API,
14       // E RETORNARÁ COM ESSAS INFORMAÇÕES
15       this.produtos =[
16         "Mouse", "Teclado", "Cabo", "Fonte"
17       ]
18     }
19     ngOnInit(): void {
20     }
21
22     adicionar(){
23       this.produtos.push("André"); /**ADICIONA UM ELEMENTO */
24     }
25     remover(){
26       this.produtos.pop();/**REMOVE O ÚLTIMO ELEMENTO DO ARRAY
27     }
28   }
29
```

EM HTML DO CARD

```
src > app > card > S card.component.html > ul > button
  Go to component
  1   <p>card works!</p>
  2
  3   <button>HELLO DEAR</button>
  4
  5   <ul>
  6     <li *ngFor="let p of produtos">{{ p }}</li>
  7     <button (click)="adicionar()">Adicionar</button>
  8     <button (click)="remover()">Remover</button>
  9
10   </ul>
11
```

NO SERVER

Sempre que clicado no botão, ele adiciona aquele statico

card works!

HELLO DEAR

- Mouse
- Teclado
- Cabo
- Fonte
- André
- André
- André
- André

Adicionar
Remover

Se clicar em Remover, com o metodo POP ele remove o ultimo elemento
 card works!

HELLO DEAR

- Mouse
- Teclado
- Cabo

Adicionar
Remover

NG FOR COM INDEX

For tem algumas coisas a mais, no entanto vamos ver somente sobre INDEX

Trabalhando com Index, vamos aperfeiçoar o REMOVER

O Index consegue colocar Index dentro de elementos

```
<li *ngFor="let p of produtos; let i = index">
| {{i}} - {{ p }}
</li>
```

Dessa forma, dentro do servidor aparece assim

- 0 - Mouse
- 1 - Teclado
- 2 - Cabo
- 3 - Fonte

Quando é deletado, o Index é deletado junto, quando é inserido,
 um novo index é inserido junto

- 0 - Mouse
 - 1 - André
 - 2 - André
 - 3 - André
 - 4 - André
 - 5 - André
 - 6 - André
 - 7 - André
 - 8 - André
 - 9 - André
 - 10 - André
 - 11 - André
- Adicionar
Remover

Vamos implementar remove de forma que, se eu clicar em algum elemento, ele já o remova

```
<li *ngFor="let p of produtos; let i = index"
(click)="remover(i)">
```

card.component.html

```

src > app > card > card.component.html > div > div > ul
    Go to component
1   <p> card works! </p>
2
3   <button>HELLO DEAR</button>
4   <div style="display: flex; flex-direction: row;">
5       <div style="display: flex; flex-direction: column; padding-right: 100px">
6           <ul>
7               <li *ngFor="let p of produtos; let i = index"
8                   (click)="remover(i)">
9                   {{i}} - {{ p }}
10              </li>
11          </ul>
12      </div>
13      <div style="display: flex; flex-direction: column;">
14          <button (click)="adicionar()">Adicionar</button>
15          <!-- <button (click)="remover()">Remover</button> -->
16      </div>
17  </div>
18

```

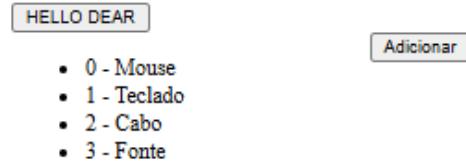
```

remover(index: number){
    // this.produtos.pop();/*REMOVE O ULTIMO ELEMENTO DO ARRAY */
    this.produtos.splice(index, 1);/*REMOVE O ELEMENTO COM INDEX DO ARRAY */
    /*POSIÇÃO QUE VAI REMOVER + QUANTIDADE DE ELEMENTOS PRA REMOVER */
}

```

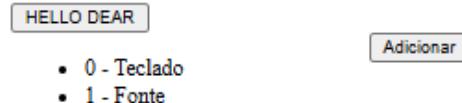
DESSA FORMA

card works!



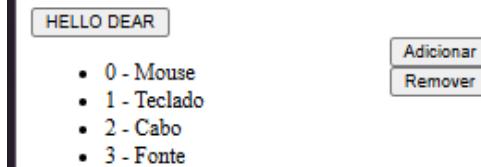
QUANDO CLICO EM UM ELE JÁ O REMOVE

card works!



VAMOS BRINCAR UM POUCO E DEIXAR O ADD E REMOVE DO OUTRO LADO

card works!



FIZ ISSO APLICANDO UM CSS INLINE

```

    HELLO DEAR</button>
    le="display: flex; flex-direction: row;">
      style="display: flex; flex-direction: column; padding-right: 100px">
    l>
    <li *ngFor="let p of produtos; let i = index">
      {{i}} - {{ p }}
    li>
    >

      style="display: flex; flex-direction: column;">
      ton (click)="adicionar()">Adicionar</button>
      ton (click)="remover()">Remover</button>

```

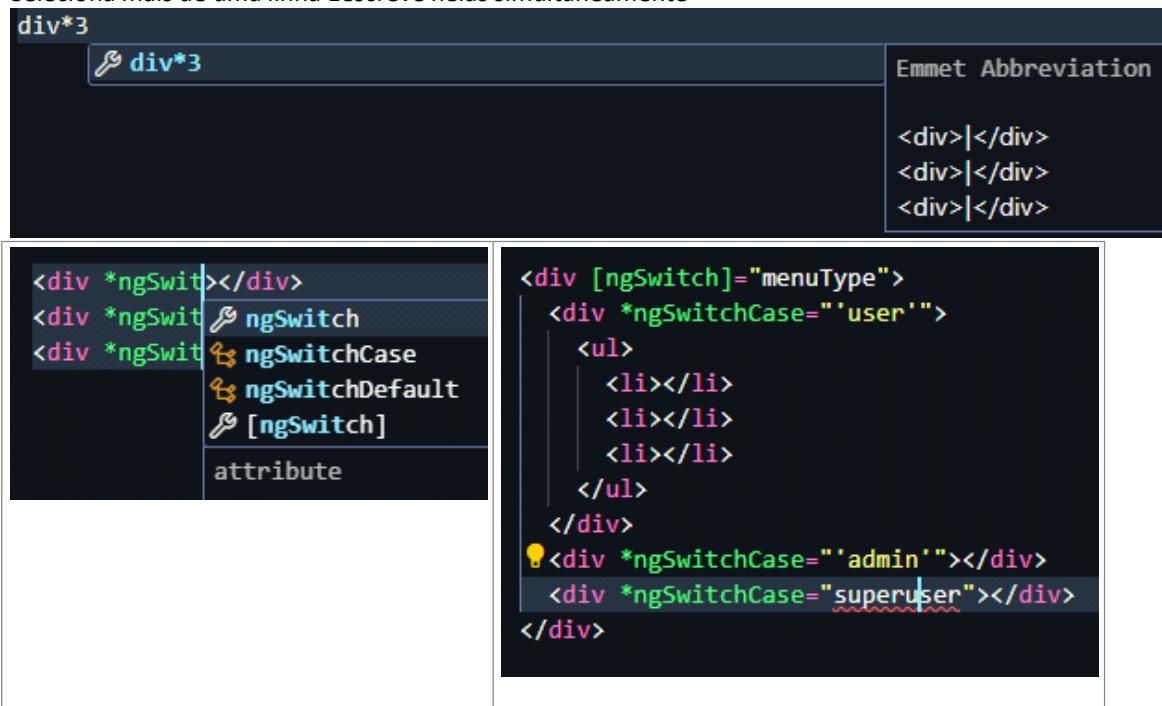
NG SWITCH

Um switch case dentro do Angular

!!! ATALHO VS CODE !!!

CTRL ALT SHIFT 'Seta para baixo'

Seleciona mais de uma linha E escreve nelas simultaneamente



SE NÃO TIVER COM ASPAS SIMPLES DEPOIS DE ASPAS DUPLAS DARA ERRO

Vamos agora com swich, ver niveis de acesso

[ngSwitch] -> Para definir que vamos mexer com switchcase

```

<div [ngSwitch]="_menuType">
  <div *ngSwitchCase="user">
    <ul>
      <li></li>
      <li></li>
      <li></li>
    </ul>
  </div>
  <div *ngSwitchCase="admin"></div>
  <div *ngSwitchCase="superuser"></div>
</div>

```

Note que esse MENUTYPE não está implementado ainda, por isso o erro

Para implementá-lo,

DENTRO DO TS

```
export class CardComponent implements OnInit {  
    /*ARRAY DE PRODUTOS*/  
    produtos:string[] = [];  
    menuType:string = "user";  
  
    constructor(){  
        // ELE FARÁ A REQUISIÇÃO NA API,  
        // E RETORNARÁ COM ESSAS INFORMAÇÕES  
        this.produtos =[
```

(property) CardComponent.menuType: string

menuType:string = "";

DENTRO DO HTML CARD

```
<div [ngSwitch]="menuType">  
    <div *ngSwitchCase="'user'">  
        <ul>  
            <li>Editar Perfil</li>  
            <li>Adicionar</li>  
        </ul>  
    </div>  
    <div *ngSwitchCase="'admin'">  
        <li>Editar Perfil</li>  
        <li>Adicionar</li>  
        <li>Gerenciar Usuários</li>  
    </div>  
    <div *ngSwitchCase="'superuser'">  
        <li>Editar Perfil</li>  
        <li>Adicionar</li>  
        <li>Gerenciar Usuários</li>  
        <li>Gerenciar Admins</li>  
    </div>  
</div>
```

Não está mais com erro

Dessa forma, foi determinado que o usuário tem um nível de acesso, o admin tem as mesmas que o user + mais acessos e o super acesso total

O Switch trabalha de forma que só um dos casos é verdadeiro

Então se no TS fizermos dessa forma...

TS

```
export class CardComponent implements OnInit {  
    /*ARRAY DE PRODUTOS*/  
    produtos:string[] = [];  
    menuType:string = "user";  
  
    constructor(){
```

SERVER

card works!

HELLO DEAR

- 0 - Mouse
- 1 - Teclado
- 2 - Cabo
- 3 - Fonte

Adicionar

- Editar Perfil
- Adicionar

Se no TS for ADMIN,

TS

```
export class CardComponent implements OnInit {  
    /*ARRAY DE PRODUTOS*/  
    produtos:string[] = [];  
    menuType:string = "admin";  
  
    constructor(){  
        // ELE FARÁ A REQUISIÇÃO NA API,  
        // E RETORNARÁ COM ESSAS INFORMAÇÕES  
        this.produtos =[
```

Server

card works!

HELLO DEAR

- 0 - Mouse
- 1 - Teclado
- 2 - Cabo
- 3 - Fonte

Adicionar

- Editar Perfil
- Adicionar
- Gerenciar Usuários

Se como SUPER USER

TS	Server
<pre>export class CardComponent implements OnInit { /*ARRAY DE PRODUTOS*/ produtos:string[] = []; menuType:string = "superuser"; constructor(){}</pre>	<p>card works!</p> <p>HELLO DEAR</p> <p>• 0 - Mouse • 1 - Teclado • 2 - Cabo • 3 - Fonte</p> <p>• Editar Perfil • Adicionar • Gerenciar Usuários • Gerenciar Admins</p> <p>Adicionar</p>

AGORA PARA NÃO PRECISAR DEFINIR O ACESSO BASICO COMO USER

Podemos deixá-lo como Default

Então, se não tiver definido como Admin ou SuperUser, ele exibirá Default

ANTES	DEPOIS
<pre><div [ngSwitch]="menuType"> <div *ngSwitchCase="'user'"> Editar Perfil Adicionar </div></pre>	<pre><div [ngSwitch]="menuType"> <div *ngSwitchDefault> Editar Perfil Adicionar </pre>

No TS

```
export class CardComponent implements OnInit {
  /*ARRAY DE PRODUTOS*/
  produtos:string[] = [];
  menuType:string = "";
```

No SERVER

card works!

HELLO DEAR

• 0 - Mouse
• 1 - Teclado
• 2 - Cabo
• 3 - Fonte

• Editar Perfil
• Adicionar

Adicionar

17,13,2 Diretivas de Atributo

sábado, 2 de novembro de 2024 13:01

Para essa aula CRIANDO UM NOVO COMPONENTE
NG G C COMP-ATRIBUTOS

Lembrando que sempre que criado um novo componente, jogar ele NO HTML geral do projeto app-comp-atributos

```
card.component.html card.component.ts app.component.html
src > app > app.component.html > app-comp-atributos
    Go to component
1   <app-card *ngIf="isAliveCard; else anotherBlock"/>
2
3   <ng-template #anotherBlock>
4       ola sou um bloco secreto
5   </ng-template>
6
7   <app-comp-atributos/>
8
9   <router-outlet />
10
```

OK

VAMOS COMEÇAR A MANIPULAR NO NOSSO CSS

```
comp-atributos.component.css
src > app > comp-atributos > comp-atributos.component.css
1   .enable{color: blue;}
2   .disable{color: red;}
3
```

```
comp-atributos.component.css comp-atributos.component.html
src > app > comp-atributos > comp-atributos.component.html > p.enable
    Go to component
1   <p class="enable">comp-atributos works!</p>
```

Se no HTML ficar dessa forma, não teria intuito, vamos fazer isso ficar dinâmico

Com BINDING

NO HTML

```
comp-atributos.component.html comp-atributos.component.ts
src > app > comp-atributos > comp-atributos.component.html > p
    Go to component
<p [ngClass]="estilo">comp-atributos works!</p>
```

NO TS

```
comp-atributos.component.html comp-atributos.component.ts
src > app > comp-atributos > comp-atributos.component.ts > CompAtributosComponent
import { Component } from '@angular/core';

@Component({
  selector: 'app-comp-atributos',
  templateUrl: './comp-atributos.component.html',
  styleUrls: ['./comp-atributos.component.css']
})
export class CompAtributosComponent {
  estilo: string = "enable"
}
```

NO SERVER

```
| comp-atributos works!
```

Para fazermos isso ter ainda mais autonomia
Podemos fazer isso com um botão liga/desliga
NO TS

```
export class CompAtributosComponent {  
  trocar() {  
    if(this.estilo == "disable"){  
      this.estilo= "enable";  
    }else {  
      this.estilo = "disable"  
    }  
  }  
}
```

NO HTML

A screenshot of a browser developer tools interface, specifically the Element tab. It shows the DOM structure of a component. At the top, there's a breadcrumb navigation: 'pp > comp-atributos > comp-atributos.component.html > ...'. Below it, under 'Go to component', is the component's template code:

```
<p [ngClass]="estilo">comp-atributos works!</p>  
<button (click)="trocar()"> Trocar </button>
```

NO SERVER

comp-atributos works!

NG STYLE

Nesse podemos trocar o estilo do elemento somente

Fazer o binding que seria deixar o nosso componente dinâmico, dessa forma...

EM TS

```
export class CompAtributosComponent implements OnInit {  
  estilo:string="disable"  
  corFundo:string ="blue"  
  corDafonte:string = "yellow"  
  
  constructor() { }  
}
```

EM HTML

```
<button (click)="trocar()">CHANGE</button>  
<h1 [ngStyle]="{{'background':corFundo, 'color':corDafonte}}>CURSO
```

NO SERVIDOR

A screenshot of a browser window. The address bar shows 'localhost:4200'. The page content is as follows:

```
comp-atributos works!  
  


# CURSO ANGULAR


```

NG MODEL

Uma diretiva que faz uma comunicação de 2 lados

Tendo um binding tanto da parte do HTML como do TS

Chamamos isso como caixinha de banana == [()]

Ele não é padrão no Angular

Por isso, é uma biblioteca,

Essa biblioteca é o FORMS MODULE

Alem disso quando importamos alguma biblioteca, não temos como fazer como em NG
GENERATE COMPONENT ...

Devemos fazer manual

EM DECLARATIONS

Entra os nossos components

EM IMPORTS

É que entra nossos imports externos...

```
imports: [  
  BrowserModule,  
  AppRoutingModule,  
  FormsModule  
,  
  ...]
```

AGORA, VAMOS IMPLEMENTAR ESSE NG MODEL

Dentro de HTML

```
comp-atributos.component.css comp-atributos.component.html 3  
src > app > comp-atributos > comp-atributos.component.html > ...  
  Go to component  
1   <p [ngClass] = "estilo" > comp-atributos works! </p>  
2   <button (click) = "trocar()" > Trocar </button>  
3  
4  
5   <input type = "text" [(ngModel)] = "item">
```

COMO NO MEU PROJETO, ALEM DELE NÃO ESTAR IMPLEMENTADO,

Ele não esta importado, lembre de importá-lo corretamente, como na imagem anterior

EM TS

```
atributos.component.html app.module.ts comp-atributos.component.ts  
src > app > comp-atributos > comp-atributos.component.ts > CompAtributosCom  
1   import { Component, OnInit } from '@angular/core';  
2  
3   @Component({  
4     selector: 'app-comp-atributos',  
5     templateUrl: './comp-atributos.component.html',  
6     styleUrls: ['./comp-atributos.component.css']  
7   })  
8   export class CompAtributosComponent implements OnInit {  
9     estilo: string = "disabled";  
10    item : string = "";  
11  
12  
13    ngOnInit(): void {  
14    }
```

Como ele pegou uma biblioteca não vista ainda nesse projeto, ele apareceu isso

```
5 |   templateUrl: './comp-atributos.component.html',  
|  
  
Initial chunk files | Names | Raw size  
polyfills.js | polyfills | 90.20 kB |  
main.js | main | 10.57 kB |  
styles.css | styles | 95 bytes |  
| Initial total | 100.86 kB  
  
Application bundle generation complete. [0.367 seconds]  
  
Page reload sent to client(s).  
15:04:05 [vite] ⚡ new dependencies optimized: @angular/forms  
15:04:05 [vite] ⚡ optimized dependencies changed. reloading
```

Agora já implementado

card works!



comp-atributos works!



Vamos fazer isso parecer mais um elemento do mundo real

Adcionando elements na lista

Pegando como aquela em FOR e fazendo uma comunicação ponta-a-ponta

EM TS

Implementado o item, list e o metodo adicionarLista

```
<h1>Lista de Compras</h1>
<ul>
| <li *ngFor="let elemento of lista">{{elemento}}
| </li>
</ul>

<input type="text" [(ngModel)]="item">

<button (click)="adicionarLista()"> Adicionar </button>

<p>{{item}}</p>
<hr>

adicionarLista() {
| this.lista.push(this.item);
| }
```

EM HTML

```
<h1>Lista de Compras</h1>
<ul>
| <li *ngFor="let elemento of lista">{{elemento}}
| </li>
</ul>

<input type="text" [(ngModel)]="item">

<button (click)="adicionarLista()"> Adicionar </button>

<p>{{item}}</p>
<hr>
```

NO SERVIDOR

Lista de Compras

- Carne



Carne

NG TEMPLATE

É um modelo

Um pedaço do HTML

Só é usado quand é chamado

POR PADRAO SE ELE NÃO FOR HABILITADO,

POR PADRAO ELE É DESATIVADO

```
<hr>

<ng-template>
  <a href="#">ADICIONAR AO CARRINHO</a>
</ng-template>
<br>
<ng-template>
  <a href="#">FINALIZAR COMPRA</a>
</ng-template>

<hr>
```

Ele está dentre essas 2 liunhas, mas não aparece pois está desativado

Agora, implementando de forma INBINDING

```
os.component.html      A app.module.ts      comp-atributos.component.ts

> app > comp-atributos > comp-atributos.component.ts > CompAtributosCo

4   selector: 'app-comp-atributos',
5   templateUrl: './comp-atributos.component.html',
6   styleUrls: ['./comp-atributos.component.css']
7 }
8 export class CompAtributosComponent implements OnInit {
9   estilo: string = "disabled";
10  item : string = "";
11  lista:string[] = [];
12  isEnableBlock:boolean = true;
```

NO HTML

```
<p>{{item}}</p>
<hr>
<ng-template [ngIf]="isEnableBlock">
| <a href="#">ADICIONAR AO CARRINHO</a>
</ng-template>
<br>
<ng-template [ngIf]="!isEnableBlock">
| <a href="#">FINALIZAR COMPRA</a>
</ng-template>

<hr>
```

No serve

Como só um deles está como true e o outro está com a negação de TRUE

ADICIONAR AO CARRINHO

NG CONTENT

Ele é um component pre pronto,

Onder ele permite que o component pai passa a informação para o filho e conseguimos utilizar o comp dinamicamente

Vamos entao para o PAI (APP COMPONENT)

HTML PAI

```
ributos.component.html A app.module.ts E app.component.html

src > app > E app.component.html > E app-comp-atributos > E p
    Go to component
1   <app-card *ngIf="isAliveCard; else anotherBlock"/>
2
3   <ng-template #anotherBlock>
4   | ola sou um bloco secreto
5   </ng-template>
6
7   <app-comp-atributos>
8   | <h2>Olá</h2>
9   | <p>Eu Vim De Fora</p>
10  </app-comp-atributos>
11
12  <router-outlet />
```

DENTRO DE UM DOS EII HO'S

[comp_attributes_component.html](#)

Podemos chamar uma dessas tags em qualquer lugar do código e mais de uma vez



The screenshot shows a code editor with three tabs at the top: 'comp-atributos.component.html', 'app.module.ts', and 'app.component'. The 'comp-atributos.component.html' tab is active and displays the following code:

```
src > app > comp-atributos > comp-atributos.component.html > ng-content
Go to component
1 <p [ngClass] = "estilo" > comp-atributos works! </p>
2 <button (click) = "trocar()" > Trocar </button>
3
4 <ng-content select = "h2"></ng-content>
5
6 <h1>Lista de Compras</h1>
7 <ul>
8 | <li *ngFor = "let elemento of lista" > {{elemento}}</li>
9 </ul>
10
11 <input type = "text" [(ngModel)] = "item">
12
13 <button (click) = "adicionarLista()" > Adicionar </button>
14
15 <p>{{item}}</p>
16 <hr>
17 <ng-template [ngIf] = "isEnabledBlock">
18 | <a href = "#" > ADICIONAR AO CARRINHO </a>
19 </ng-template>
20 <br>
21 <ng-template [ngIf] = "!isEnabledBlock">
22 | <a href = "#" > FINALIZAR COMPRA </a>
23 </ng-template>
24
25 <hr>
26 <ng-content selected = "p"></ng-content>
```

NO SERVER

Perceba que ele está sendo chamado em dois lugares diferentes,

Essa é uma maneira simples de utilizá-lo, mas podemos fazer isso de mais formas,
como utilizar um componente dentro de outro

Através de tags dentro de tags

17,14 MODULOS ANGULAR

sábado, 2 de novembro de 2024 15:55

LEVE EM CONSIDERAÇÃO COMO NO MUNDO REAL, ONDE NOS TEMOS ALGUMAS GAVERAS, CADA UMA PARA UMA PEÇA DE ROUPA

Nos temos uma para Calça,

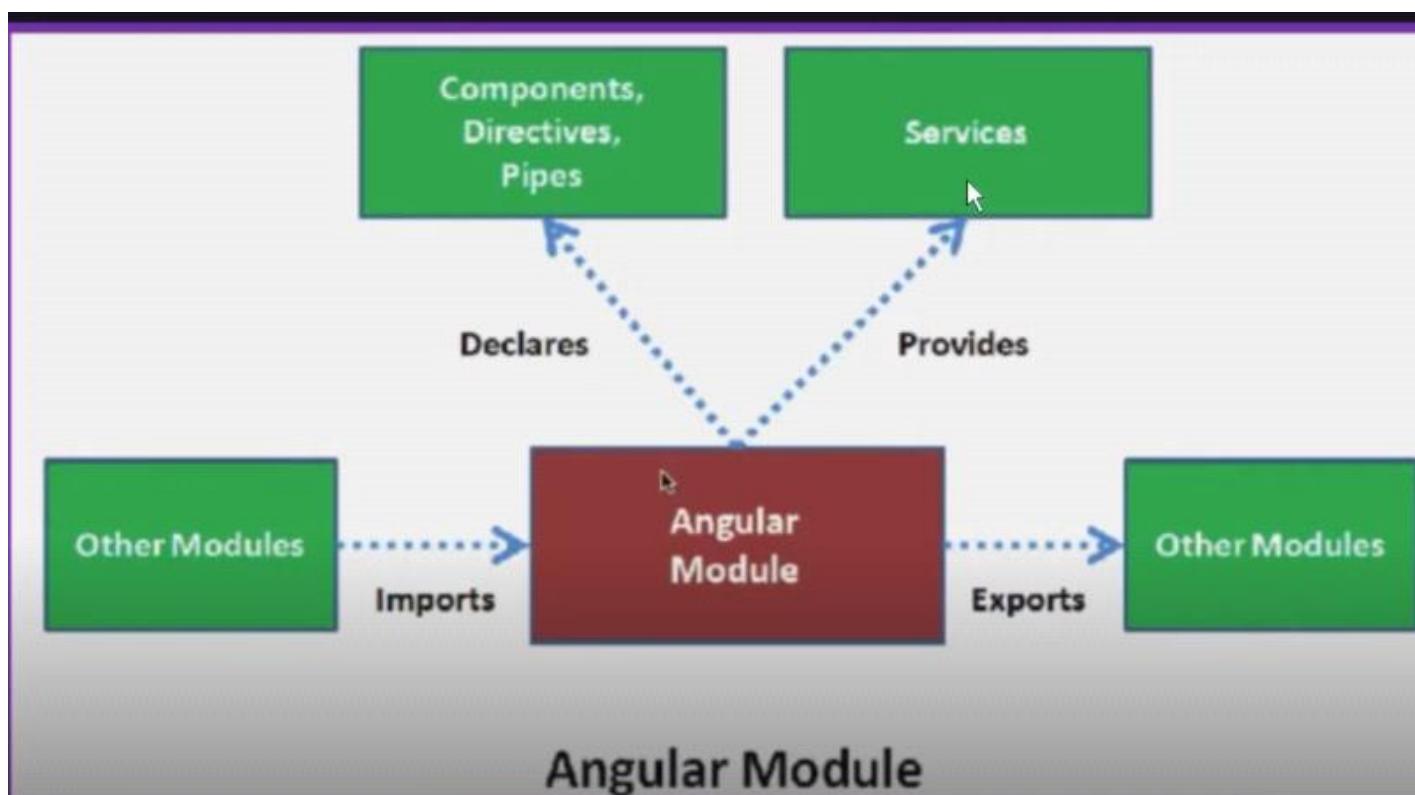
Camisa,

Intimos,

Shorts...

SE COLOCARMOS TUDO JUNTO, FICARA UMA BAGUNÇA

Em NG é a mesma coisa, não vamos colocar tudo em App Module, vamos criar demais modulos e esses modulos nos vamos atrelar tudo ao mesmo GuardaRoupas/AppModule



Como já visto

Os modulos podem declarar componentes, diretivas, pipelines

Podem Prover dados de outras APIs ou servicos de um BD

Podem conter outros modulos

Como criar um modulo

O que ele precisa ter

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';

import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';

@NgModule({
  declarations: [
    AppComponent,
  ],
  imports: [
    BrowserModule,
    AppRoutingModule,
    FormsModule
  ]
})
```

```
  ],
  exports:[],
  providers: [],
  bootstrap: [AppComponent]
})
```

Exports é para onde esse modulo ira ser exportado

Como o AppModule é o principal, ele não contem, mas os modulos secundarios deverao ter para ser importados pelo Principal

Providers é para injeção de dependente com Services, comunicação com o Banco

Bootstrap, é o componente de inicialização

17,14,1 Criando Modulo com NG

sábado, 2 de novembro de 2024 16:19

Para criação de MODULO COM

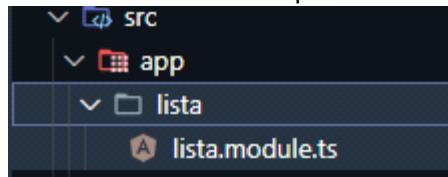
NG MODULE

Ng g m "nome componente"

```
ng g m lista
```

```
PS D:\conta.ACER-ANDRE-SANT\Ambiente de Desenvolvimento\Angular\Angular-17\src\app\lista\lista.module.ts (203 bytes)
```

Foi criado um novo component



Agora para criarmos algum componente dentro dessa pasta e dentro desse modulo, temos que fazer dessa forma

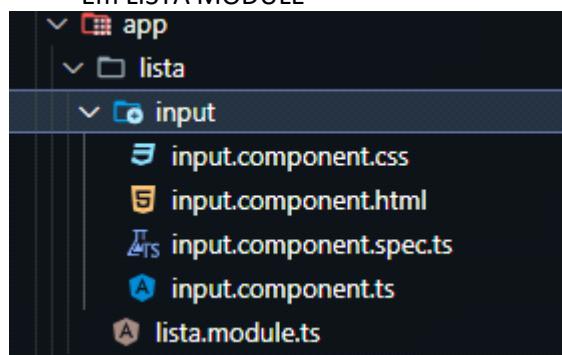
```
ng g m lista
```

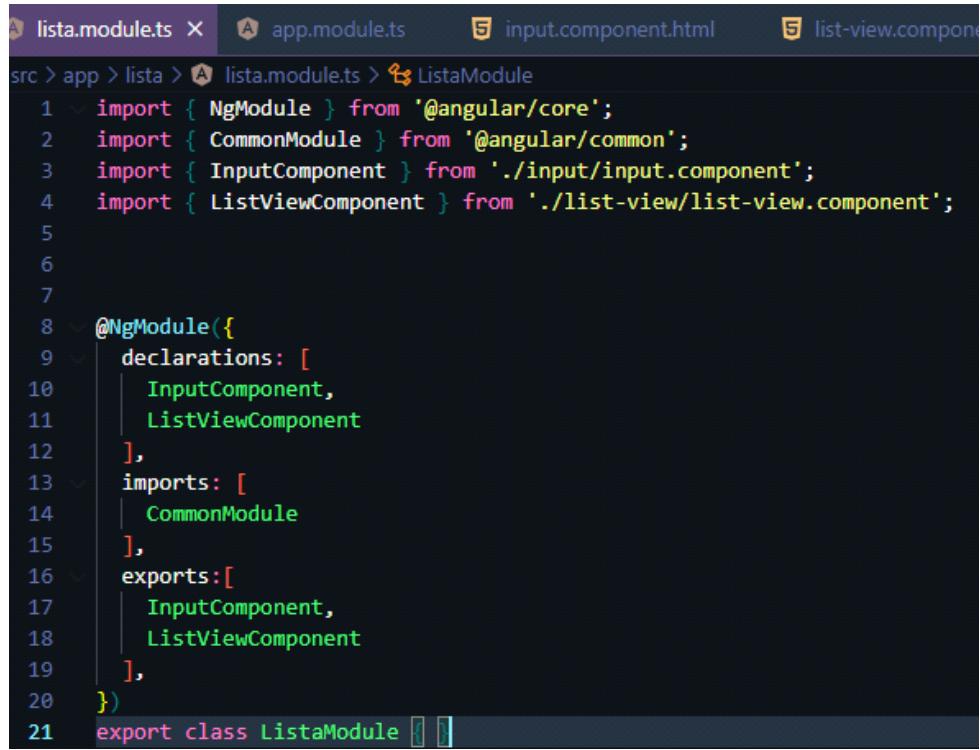
```
ng g c lista/input
```

Dessa forma foi criado os arquivos correspondentes e já pre configurado

Ao inves de em APP MODULE

Em LISTA MODULE





```

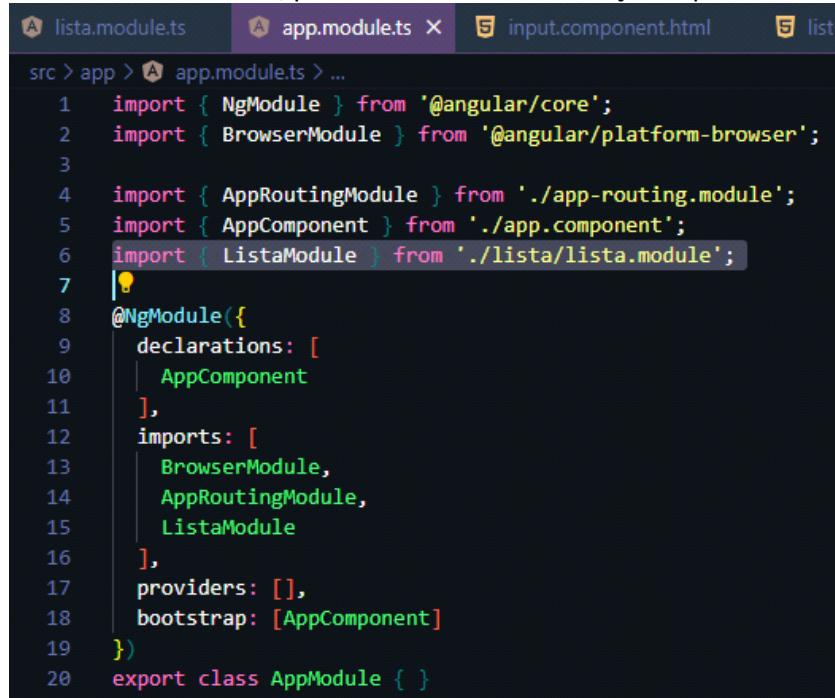
src > app > lista > lista.module.ts > ListaModule
1 import { NgModule } from '@angular/core';
2 import { CommonModule } from '@angular/common';
3 import { InputComponent } from './input/input.component';
4 import { ListViewComponent } from './list-view/list-view.component';
5
6
7
8 @NgModule({
9   declarations: [
10     InputComponent,
11     ListViewComponent
12   ],
13   imports: [
14     CommonModule
15   ],
16   exports:[
17     InputComponent,
18     ListViewComponent
19   ],
20 })
21 export class ListaModule {} []

```

Para eu conseguir exportar meu componente para utilizar em outro lugar, vamos inclui-lo em Export(ele não vem por padrao, entao temos que criar a solução)

Dentro de APP MODULE

Vamos chama-lo, para dessa forma ele começar a aparecer



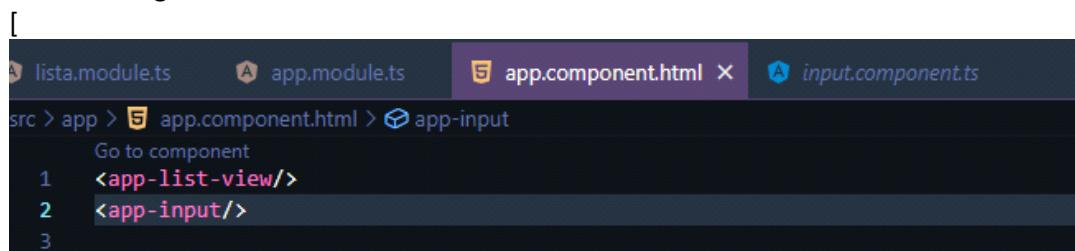
```

src > app > app.module.ts > ...
1 import { NgModule } from '@angular/core';
2 import { BrowserModule } from '@angular/platform-browser';
3
4 import { AppRoutingModule } from './app-routing.module';
5 import { AppComponent } from './app.component';
6 import { ListaModule } from './lista/lista.module';
7
8 @NgModule({
9   declarations: [
10     AppComponent
11   ],
12   imports: [
13     BrowserModule,
14     AppRoutingModule,
15     ListaModule
16   ],
17   providers: [],
18   bootstrap: [AppComponent]
19 })
20 export class AppModule {}

```

Como já implementado nos arquivos Main

Conseguimos chama-los normalmente



```

src > app > app.component.html > app-input
1 <app-list-view>
2 <app-input/>
3

```

The screenshot shows a code editor with four tabs at the top: lista.module.ts, app.module.ts, app.component.html (which is the active tab), and input.component.ts. Below the tabs, there is a breadcrumb navigation bar: src > app > app.component.html > app-input. The main content area contains the following code:

```
src > app > app.component.html > app-input
    Go to component
1   <app-list-view/>
2   <app-input/>
3
4
5   <router-outlet />
6
```

Como já exportado tudo no Modulo da Lista, conseguimos só chamar aquele modulo e aquele modulo vai exibir tudo o que lá foi implementado

list-view works!

- 1
- 2
- 3

input works!

18 Criando um Blog com Angular

sábado, 2 de novembro de 2024 19:33

Primeiramente, Obrigado por estar aqui

Seguindo as aulas e Passo a Passo do Felipao da DIO

Fazendo esse projeto, anotando as melhores práticas e fundamentos para esse projeto

Chegando também ao fim do Bootcamp onde vi e/ou revi sobre

SPRING BOOT

HTML

CSS

JS

TYPESCRIPT

NODE

ANGULAR

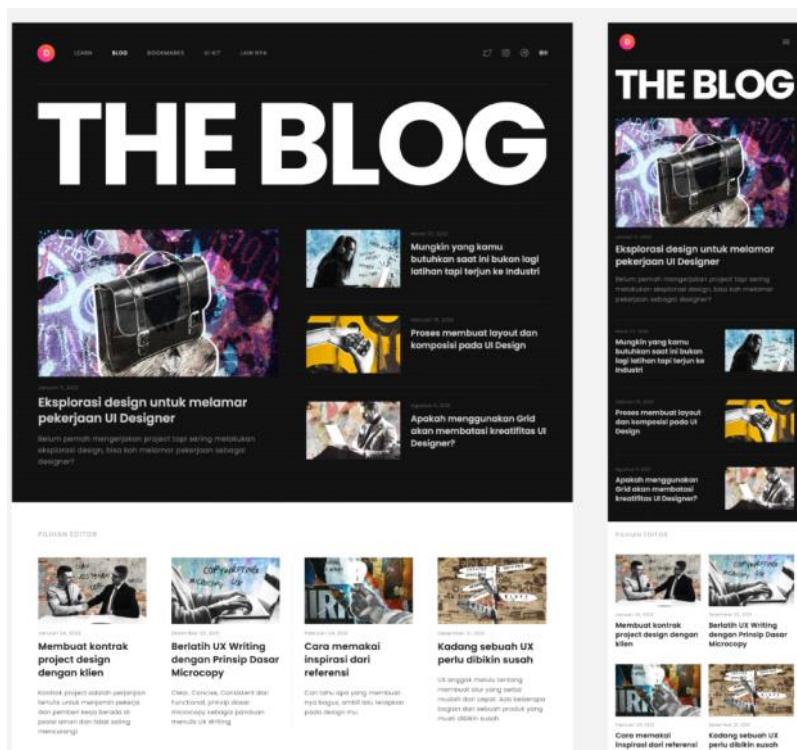
Vamos a algumas atenção antes do projeto

Primeiramente, sempre que quiser praticar mas não tem nenhuma inspiração ou base

Podemos visualizar através de um site: <https://dribbble.com/>

Vamos pegar esse como exemplo

<https://cdn.dribbble.com/userupload/2623158/file/original-ff44a9f2238eef9a1e0b5e21d27e7f94.png?resize=1600x1200>



Logo após, para pegar as cores exatas de cada parte, podemos utilizar o JUST COLOR PICKER > ANNY STUDIO

<https://annystudio.com/software/colorpicker/>



Para termos uma base de como criar uma representação gráfica de algo,

Vamos definir primeiro o que precisamos modelar

Vamos pegar essa imagem e como exemplo, jogar no PowerPoint/Paint/VSCode...

Essa BAR pode ser um component sendo reutilizado dentro de mais paginas

Esse card pode ser reutilizado

Esse Card Destaque ou BigCard

Tem uma forma de deixar o site de forma responsiva, para que em mobile consiga ver normal sem ter que ficar colocando como em Modo Computador

Atraves do MEDIA QUERIES CSS

The screenshot shows a browser window displaying the MDN Web Docs page on media queries. The URL is https://developer.mozilla.org/pt-BR/docs/Web/CSS/CSS_media_queries. The page content discusses the `and` keyword in media queries.

A palavra-chave `and` é usada para combinar múltiplas `media features`, bem como combinar `media features` com `media types`.

Uma `media query` básica, uma `media feature` simples com a `media type` `all`, pode parecer com isso:

```
css
@media (min-width: 700px) { ... }
```

Se, no entanto, você desejar que isso se aplique apenas para telas em `landscape`, você pode usar o operador `and` para deixar todas as `media features` juntas.

```
css
@media (min-width: 700px) and (orientation: landscape) { ... }
```

Agora, a `media query` acima vai apenas retornar verdadeira se o `viewport` for `700px`, `wide` ou `wider` e a tela estiver em `landscape`. Se, no entanto, você deseja apenas que isso seja aplicado se a tela em questão for `media type TV`, você pode encadear essas `features` com a `media type` usando o operador `and`.

```
css
@media tv and (min-width: 700px) and (orientation: landscape) { ... }
```

Agora, a `media query` acima vai ser aplicada apenas se a `media type` for `TV`, o `viewport` for `700px wide` ou `wider`, e a tela estiver

Através dessa regras, conseguimos deixar o código de forma responsiva

18,1 Etapa por Etapa do que foi feito do BLOG

domingo, 3 de novembro de 2024 14:42

Através das aulas da DIO começamos um site/blog com HTML CSS e Angular
começamos a colocar elementos de forma estáticos

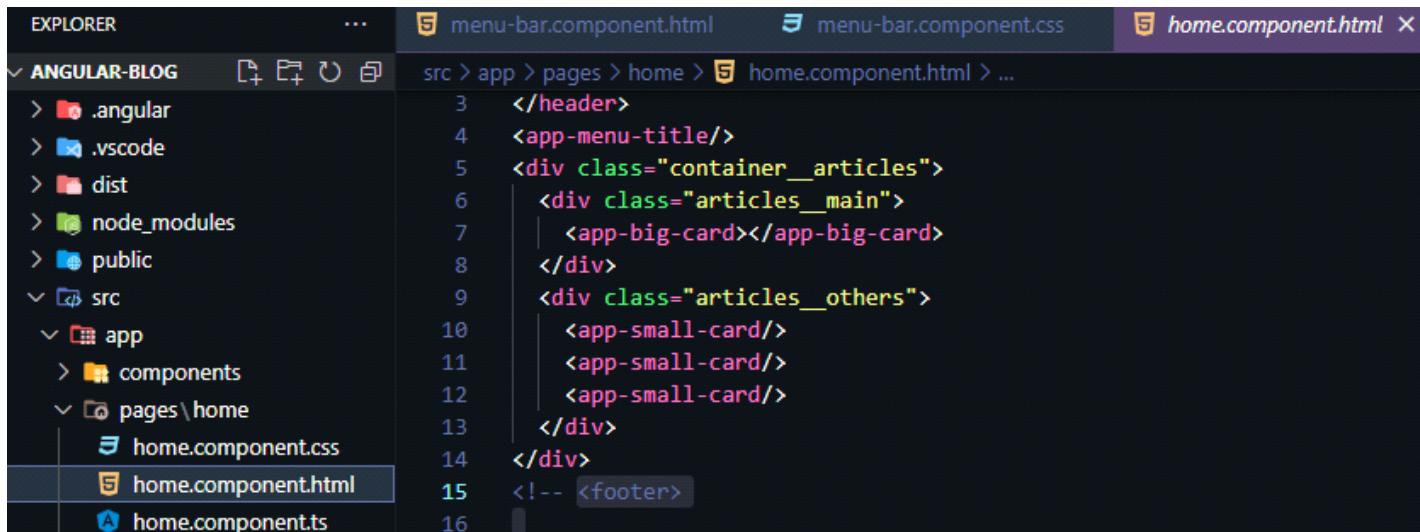
Agora vamos definir de forma dinâmica

Mas antes disso vou colocar como que está agora

Explicando por cima o que faz cada coisa

Primeiro, vamos ter a pasta PAGINAS

Nela estara a Home/1°/2°... Paginas



The screenshot shows the VS Code interface with the following details:

- EXPLORER**: Shows the project structure: ANGULAR-BLOG, .angular, .vscode, dist, node_modules, public, and src. Under src, there is an app folder containing components and pages. The pages folder has a home folder which contains home.component.css, home.component.html, and home.component.ts.
- menu-bar.component.html**, **menu-bar.component.css**, and **home.component.html** tabs are visible at the top.
- home.component.html** content (lines 1-16):

```
</header>
<app-menu-title/>
<div class="container__articles">
  <div class="articles_main">
    <app-big-card></app-big-card>
  </div>
  <div class="articles_others">
    <app-small-card/>
    <app-small-card/>
    <app-small-card/>
  </div>
</div>
<!-- <footer> -->
```

Lembrando que a página home é um componente como qualquer outro
então para criá-lo é só jogar o comando

NG G C pages/home

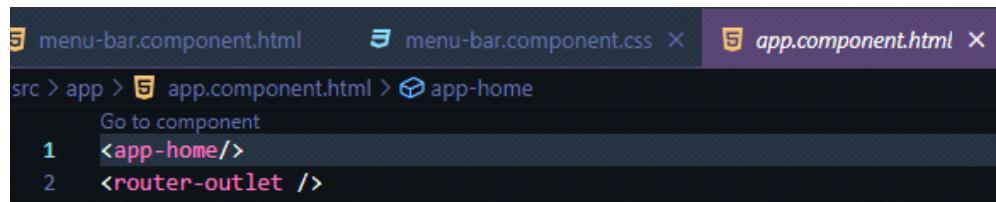
Isso fará com que ele jogue esse componente dentro de PAGES

Aqui estarão todos os componentes que precisarão ser colocados dentro de home

O home por sua vez será jogado dentro do app componente
dessa forma ele pegará tudo o que está dentro dos outros componentes só que ficará
dentro de home
deixando mais fácil de visualização do código

App component

Ficará assim



The screenshot shows the VS Code interface with the following details:

- menu-bar.component.html**, **menu-bar.component.css**, and **app.component.html** tabs are visible at the top.
- app.component.html** content (lines 1-2):

```
<app-home/>
<router-outlet />
```

Em styles.css (CSS GERAL DO PROJETO)

Antes de mais nada

nós conseguimos setar o CSS geral

dessa forma tudo que tiver dentro do projeto, se não tiver nenhuma regra dentro
de algum componente

Tudo e Qualquer elemento que tiver no site terá que obedecer essas regras

```
menu-bar.component.html      menu-bar.component.css      styles.css X
> styles.css > ...
1  /* You can add global styles to this file, and also import other ...
2  *
3  | margin: 0;
4  | padding: 0;
5  | box-sizing: border-box;
6  | background-color: black;
7  | color: white;
8  | font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
9  |
10 |
11 a{
12   text-decoration: none;
13 }
14 |
```

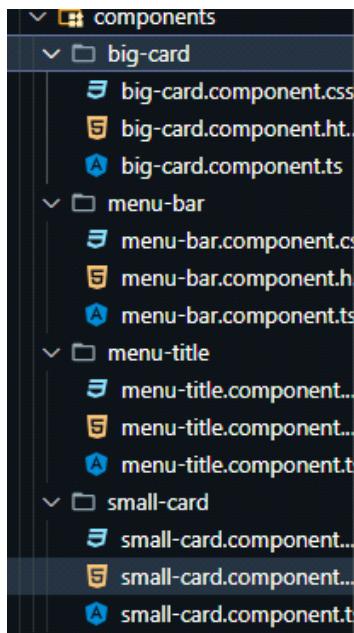
Em home.css

```
menu-bar.component.html      menu-bar.component.css      home.component.css X
src > app > pages > home > home.component.css > ...
1  .container__articles{
2    margin-top: 30px;
3    display: flex;
4    flex-direction: row;
5    justify-content: center;
6  }
7
8  header{
9    border: 1px solid red;
10   display: flex;
11   justify-content: center;
12 }
13
14 /* footer{} */
15 |
```

Vamos aos demais componentes



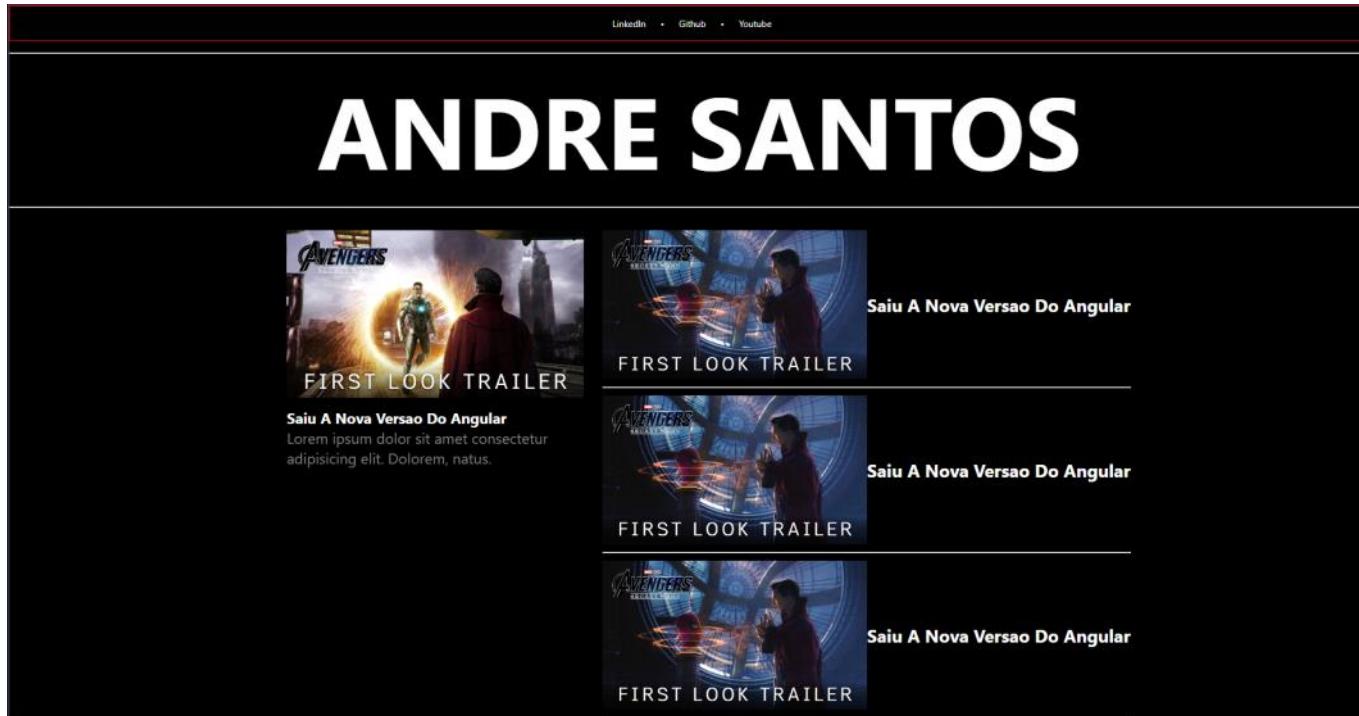
Componentes de HOME



Não vou passar os prints de todos os componentes porque se não existem muito e hoje não tem muito tempo

Nessa aba de componentes temos
o menu bar que fará parte bar do nosso home
Temos um Title que fará aquela parte grandona de título
Temos a parte do big card que fará aquele card spam maiorzão
Ao lado dele temos os small cards que são aqueles cardzinhos menores de destaque

No SERVE



Por fim, a epresentação grafica de como está ate agora

Claro, é so uma brincadeira, mas essa bar poderia ser uma Footer com essas informações e o Bar deixar as pages HOME|DESTAQUE|CATEGORIA... como feito em JS

Proximo passo?

Agora vamos deixar isso de forma autonoma, podendo manipular os componentes
VAMOS INBAINDAR

18,2 Refatorando, deixando comps dinamicos

domingo, 3 de novembro de 2024 15:07

BIG CARD

EM HTML BIG CARD

```
src > app > components > big-card > big-card.component.html > div
  Go to component
  1 <div class="container__big-card">
  2   <div class="big-card__photo">
  3     
  7   </div>
  8   <div cl (element) a: HTMLAnchorElement
  9     <h1><a href="#">{{ cardTitle }}</a></h1>
 10   </div>
 11   <div class="big-card__description">
 12     <p><a href="#">{{ cardDescription }}</a></p>
 13   </div>
 14 </div>
 15
```

Uma Annotation muito legal para se utilizar em TS é o @Input

Dessa forma, conseguimos agilizar e setar direto na home do nosso projeto o que queremos deixar na "home" de fato

EM TS BIG CARD

```
src > app > components > big-card > big-card.component.ts > BigCardComponent > component
  1 import { Component, Input, OnInit } from '@angular/core';
  2
  3 @Component({
  4   selector: 'app-big-card',
  5   templateUrl: './big-card.component.html',
  6   styleUrls: ['./big-card.component.css']
  7 })
  8 export class BigCardComponent implements OnInit {
  9   @Input()
 10   photoCover:string="";
 11   @Input()
 12   cardTitle:string="";
 13   @Input()
 14   cardDescription:string="";
 15
 16
 17
 18   ngOnInit(): void {}
 19
 20 }
 21
```

EM HTML PAGES/HOME

```
<app-big-card
  photoCover="https://i.pinimg.com/originals/89/91/24/899124bbdac51df737db3bd012eeaaa9.jpg"
  cardTitle="Vingadores Guerras Secretas"
  cardDescription="maiores heróis e vilões do Universo Marvel são transportados até um mundo alternativo criado por Thanos.">
</app-big-card>
</div>
```

Quando chamando a tag, agora setar os valores que queira deixar

Podemos presumir que o Bigcard esta OK por hora

SMALL CARD

Mesma coisa,
Em TS

small-card.component.ts X small-card.component.css big-card.com

```
src > app > components > small-card > small-card.component.ts > SmallCardComponent
1 import { Component, Input, OnInit } from '@angular/core';
2
3 @Component({
4   selector: 'app-small-card',
5   templateUrl: './small-card.component.html',
6   styleUrls: ['./small-card.component.css'
7 })
8 export class SmallCardComponent implements OnInit {
9   @Input()
10  photoCover:string = "";
11  @Input()
12  cardTitle:string = "";
13  @Input()
14  cardLink:string = "";
15
16  ngOnInit(): void {}
17 }
```

Em HTML

Go to component

```
<div class="container__small-card">
  <div class="container__small-card-content">
    <div class="small-card__photo">
      
    </div>
    <div class="small-card__title">
      <h1><a href="{{cardLink}}>{{cardTitle}}</a></h1>
    </div>
    <hr>
  </div>
```

EM CSS

small-card.component.ts small-card.component.css

```

src > app > components > small-card > small-card.component.css
1  .container__small-card-content{
2    /* border: solid white 2px; */
3    display: flex;
4    flex-direction: row;
5    width: 800px;
6    height: 200px;
7  }
8  .small-card__photo > img{
9    display: flex;
10   justify-content: center;
11   align-items: center;
12   height: 180px;
13   width: 300px;
14   margin-right: 15px;
15 }
16
17 .small-card__title{
18   display: flex;
19   justify-content: center;
20   align-items: center;
21 }
22
23 .container__small-card > hr{
24   margin-top: 10px;
25   margin-bottom: 10px;
26 }
27
28

```

EM PAGES/HOME HTML

```

<div class="articles__others">
  <app-small-card
    photoCover="https://www.oliberale.com/image/contentid/policy:1.879759:1729888031/Zendaya-como-MJ-e-Tom-Holland"
    cardLink="#"
    cardTitle="Homem-Aranha 4': Marvel divulga data de estreia do filme com Tom Holland e Zendaya"
  ></app-small-card>
  <app-small-card
    photoCover="https://www.papodecinema.com.br/wp-content/uploads/2013/07/20180516-123778-aleni_800x600.jpg"
    cardLink="#"
    cardTitle="novo filme de X-Men em produção"
  ></app-small-card>
  <app-small-card
    photoCover="https://i.pinimg.com/originals/89/91/24/899124bbdac51df737db3bd012eeaaa9.jpg"
    cardLink="#"
    cardTitle="Nova Serie Anunciada"
  ></app-small-card>
</div>

```

NO SERVER

Dessa forma, conseguimos deixar o essencial de forma dinâmica

[Linkedin](#) • [Github](#) • [Youtube](#)

ANDRE SANTOS



FIRST LOOK TRAILER



Homem-Aranha 4': Marvel divulga data de estreia do filme com Tom Holland e Zendaya

ANDRE SANTOS



FIRST LOOK TRAILER



Homem-Aranha 4': Marvel divulga data de estreia do filme com Tom Holland e Zendaya



novo filme de X-Men em produção



Nova Serie Anunciada

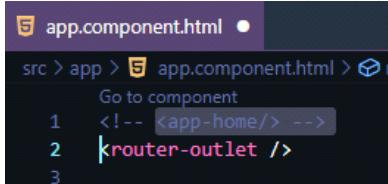
Da mesma forma que mesmo quando clicado num link, podemos futuramente redirecionar para o link aquele destaque

18,3 Adicionando paginas

domingo, 3 de novembro de 2024 16:00

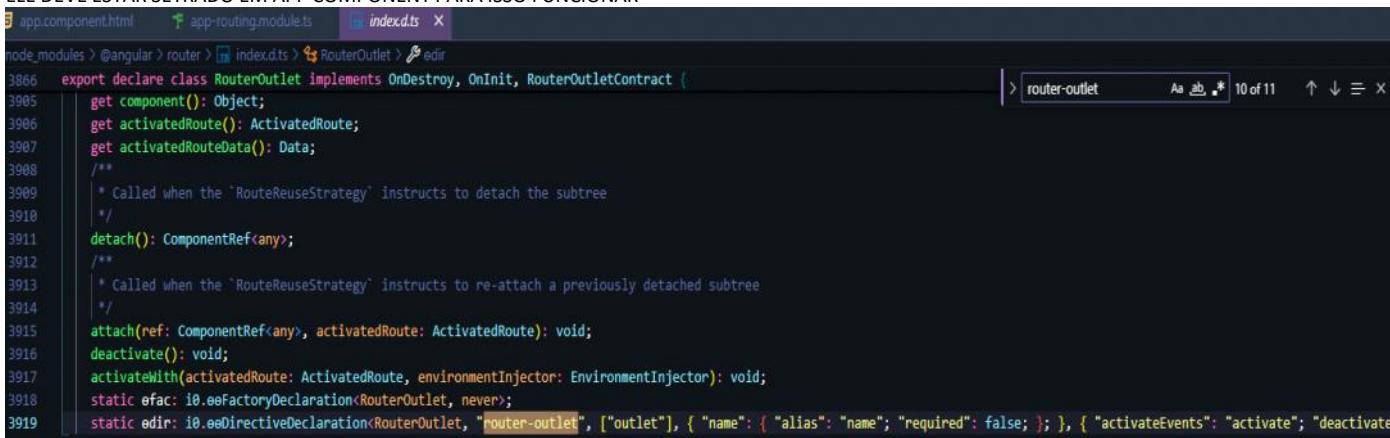
Para navegar em paginas

Agora não devemos mais ter setado a tag das paginas, quando mais de pagina



```
<!-- <app-home/> -->
<router-outlet>
```

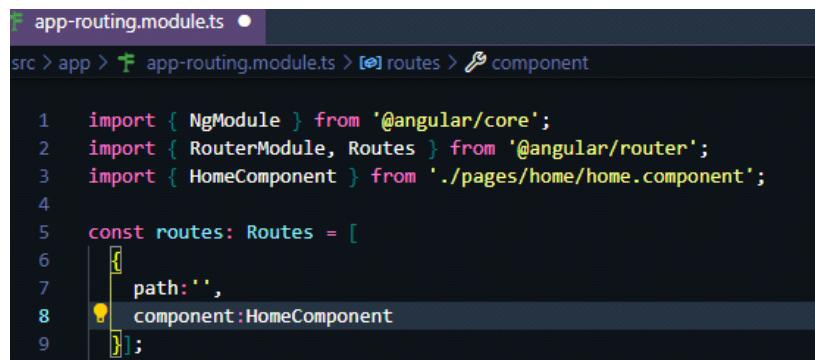
SIM, ESSE ROUTER-OUTLET É DO GERENCIAMENTO DE ROTAS,
ELE DEVE ESTAR SETRADO EM APP COMPONENT PARA ISSO FUNCIONAR



```
export declare class RouterOutlet implements OnDestroy, OnInit, RouterOutletContract {
  get component(): Object;
  get activatedRoute(): ActivatedRoute;
  get activatedRouteData(): Data;
  /**
   * Called when the `RouteReuseStrategy` instructs to detach the subtree
   */
  detach(): ComponentRef<any>;
  /**
   * Called when the `RouteReuseStrategy` instructs to re-attach a previously detached subtree
   */
  attach(ref: ComponentRef<any>, activatedRoute: ActivatedRoute): void;
  deactivate(): void;
  activateWith(activatedRoute: ActivatedRoute, environmentInjector: EnvironmentInjector): void;
  static efac: i0.ɵɵFactoryDeclaration<RouterOutlet, never>;
  static edir: i0.ɵɵDirectiveDeclaration<RouterOutlet, "router-outlet", [ "outlet" ], { "name": { "alias": "name"; "required": false; } }, { "activateEvents": "activate"; "deactivateEvents": "deactivate" }>;
}
```

Podemos fazer pelo App Routing

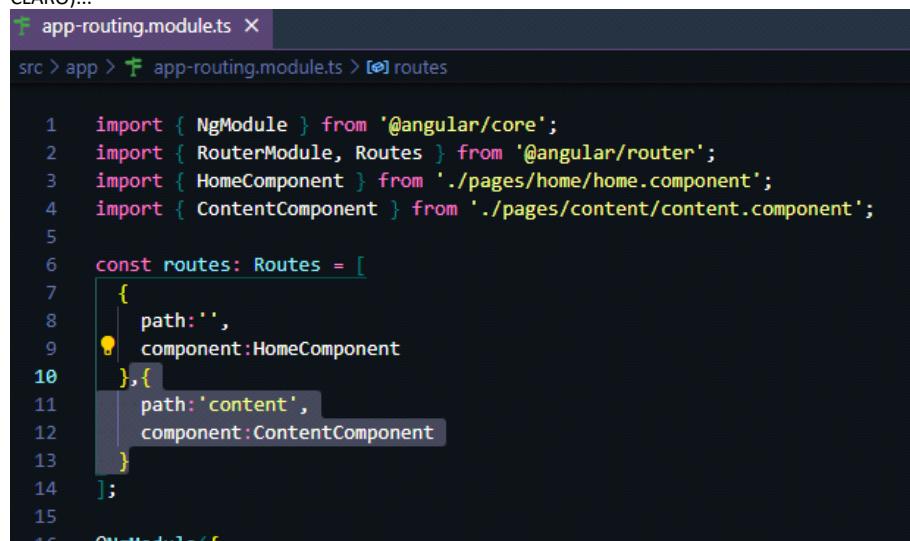
Setando dentro do Array, path e o caminho das paginas



```
const routes: Routes = [
  {
    path: '',
    component: HomeComponent
  }
];
```

Quando clicando em OK em setar esse HOME COMPONENT, automaticamente ele insere la em cima, como na foto (em import)

Para inserir mais de uma pagina... (DEPOIS DE CRIADA A PAGINA/COMPONENTE CLARO)...

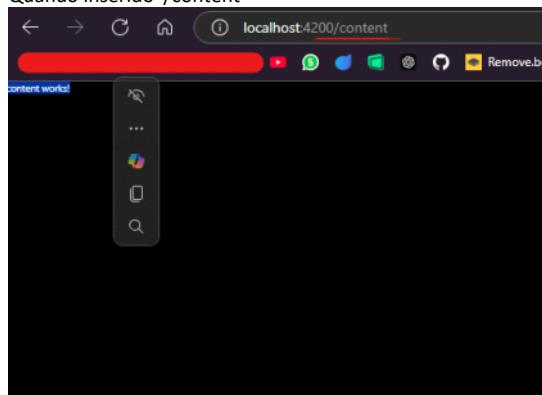


```
const routes: Routes = [
  {
    path: '',
    component: HomeComponent
  },
  {
    path: 'content',
    component: ContentComponent
  }
];
```

NO SERVER

The screenshot shows a web page with a large title "ANDRE SANTOS" at the top. Below it is a thumbnail for "FIRST LOOK TRAILER" featuring Iron Man and Doctor Strange. To the right is a thumbnail for "Homem-Aranha 4: Marvel divulga trailer do filme com Tom Holland e Zendaya". Further down is a thumbnail for "novo filme de X-Men em produção" featuring the X-Men team. At the bottom is another thumbnail for "FIRST LOOK TRAILER" featuring Iron Man and Doctor Strange. The page has a dark background.

Quando inserido '/content'



Em HTML CONTENT

The screenshot shows a code editor with three tabs: "content.component.css", "content.component.ts", and "content.component.html". The "content.component.html" tab is active and displays the following HTML code:

```
src > app > pages > content > content.component.html > div.container_content > div.content
  Go to component
  1 <div class="container_content">
  2   <div class="content_cover">
  3     
  4   </div>
  5   <div class="content_title">
  6     <h1>{{contentTitle}}</h1>
  7     <hr>
  8   </div>
  9   <div class="content_description">
 10     <p>{{contentDescription}}</p>
 11   </div>
 12 </div>
```

EM TS

The screenshot shows a code editor with two tabs open: 'content.component.ts' and 'content.component.css'. The 'content.component.ts' tab contains the following code:

```
src > app > pages > content > A content.component.ts > ContentComponent > contentDescription
1 import { Component, OnInit } from '@angular/core';
2
3 @Component({
4   selector: 'app-content',
5   templateUrl: './content.component.html',
6   styleUrls: ['./content.component.css']
7 })
8 export class ContentComponent implements OnInit {
9   photoCover:string = "https://th.bing.com/th/id/OIP.KSa-BsHmZPI4GobXZIKWMwHaEK?w=3840&h=2160&rs=1&pid=ImgDetMain";
10  contentTitle:string="Implementando";
11  contentDescription:string = "Implementando";
12
13  ngOnInit(): void { }
14
15 }
16
```

EM CSS

The screenshot shows the 'content.component.css' file with the following CSS rules:

```
src > app > pages > content > content.component.css > .container_content
1 .container_content{
2   justify-content: center;
3   align-items: center;
4   margin-top: 20px;
5
6 }
7
8 .content_cover{
9   border: 1px solid black;
10
11 }
12
13 .content_cover > img{
14   border: 5px solid red;
15   border-radius: 3px;
16   min-width: 500px;
17   width: 500px;
18   height: 300px;
19
20 }
21
22 .content_title{
23   margin-top: 1px;
24
25 }
26
27 .content_description{
28   margin-top: 15px;
29
30 }
31
32 .content_description > p{
33   color: #rgb(141, 138, 138);
34
35 }
```

Próximo passo é fazer uma chamada de elementos de forma dinâmica, sendo que pegando um id ele exibira um determinado elemento, outro ID outro, e assim por diante...

18,4 Dados de forma Dinamica

domingo, 3 de novembro de 2024 17:19

Lembra que os links já estão meio que de forma dinâmica com link certo?

Mas agora, vamos levar em consideração que temos imagem e informação que vem direto de um banco "fake"

Vamos imaginar que veio de um banco que no momento está em DATA

```
src > app > data > datafake.ts
1  export const dataFake = [
2    {
3      "id":1,
4      "photo": "aaa",
5      "title": "Salu Novo Filme",
6      "description": "Olá"
7    },
8    {
9      "id":2,
10     "photo": "aaa",
11     "title": "Salu Novo Filme",
12     "description": "Olá"
13   }
14 ]
```

Dentro de Rotas, vamos dizer que o Content recebe um id, que seria esse mesmo do Data Fake

```
src > app > app-routing.module.ts
1  import { NgModule } from '@angular/core';
2  import { RouterModule, Routes } from '@angular/router';
3  import { HomeComponent } from './pages/home/home.component';
4  import { ContentComponent } from './pages/content/content.component';
5
6  const routes: Routes = [
7    {
8      path:'',
9      component: HomeComponent
10    },
11    {
12      path:'content/:id',
13      component:ContentComponent
14    }
15  ];
16
17 @NgModule({
18   imports: [RouterModule.forRoot(routes)],
19   exports: [RouterModule]
20 })
21 export class AppRoutingModule {}
```

Agora sempre que for visitado essa página, ele pedirá um ID

localhost...//content/1

Vamos deixar agora Dinâmico

Para isso, dentro de TS Big Card e em seguida em Small Card

Vamos fazer um atributo Id iniciando em 0

```
src > app > components > big-card > big-card.component.ts
1  import { Component, Input, OnInit } from '@angular/core';
2
3  @Component({
4    selector: 'app-big-card',
5    templateUrl: './big-card.component.html',
6    styleUrls: ['./big-card.component.css']
7  })
8  export class BigCardComponent implements OnInit {
9    @Input()
10   Id:string = "0";
```

Agora dentro de Pages/Home

Setar um ID

```
src > app > pages > home > home.component.html
1  <app-menu-title>
2  <div class="container__articles">
3    <div class="articles__main">
4      <app-big-card
5        Id="1"
6        photoCover="https://i.pinimg.com/or...
7        cardLink="#"
8        cardTitle="Vingadores Guerras Secretas"
9        cardDescription=" Maiores heróis e vilões da história do cinema" >
```

Dentro de Big Card HTML

Agora dentro dessa pasta, fazer uma chamada com rotas,

Chamar a rota definida como content, e o ID com as informações correspondentes

```
src > app > components > big-card > big-card.component.html
1  <div class="container__big-card">
2    <div class="big-card__photo">
3      
7    </div>
8    <div class="big-card__title">
9      <p><a [routerLink]="'content', Id">{{ cardTitle }}</a></p>
10     | <-- <h1><a [routerLink]="{{ cardLink }}'">{{ cardTitle }}</a></h1> -->
11   </div>
```

Em Routing

```
}, {
  path: 'content/:id',
  component: ContentComponent
}
```

Dessa forma, como eles estão configurados com ID, cada um deverá ir para uma tela diferente

```
home.component.html
src > app > pages > home > home.component.html > div.container__articles > div.articles_main > app-big-card
  Go to component!
  1 <app-menu-title/>
  2 <div class="container__articles">
  3   <div class="articles_main">
  4     <app-big-card
  5       Id="1"
  6       photoCover="https://i.pinimg.com/originals/89/91/24/899124bbdac51df737db3bd012eaaa9.jpg"
  7       cardTitle="Vingadores: Guerras Secretas"
  8       cardDescription=" Maiores heróis e vilões do Universo Marvel são transportados até um mundo novo."
  9     </app-big-card>
 10   </div>
 11   <div class="articles_others">
 12     <app-small-card
 13       Id="2"
 14       photoCover="https://www.oliberal.com/image/contentid/policy:1.879759:1729888031/Zendaya-como-Homen-Aranha-4"
 15       cardTitle="Homem-Aranha 4": Marvel divulga data de estreia do filme com Tom Holland e Zendaya
 16     </app-small-card>
 17     <app-small-card
 18       Id="3"
 19       photoCover="https://www.papodecinema.com.br/wp-content/uploads/2013/07/20180516-123778-alice-in-wonderland-2010-movie-stills-1024x768.jpg"
 20       cardTitle="Novo filme de X-Men em produção"
 21     </app-small-card>
 22     <app-small-card
 23       Id="4"
 24       photoCover="https://i.pinimg.com/originals/89/91/24/899124bbdac51df737db3bd012eaaa9.jpg"
 25       cardTitle="Nova Série Anunciada"
 26     </app-small-card>
 27   </div>
 28 </div>
```

NOTE OS IDs

Como eles estão um em cada ID, de forma inbinding eles serão chamados cada um contendo seus respectivos IDs e informações

Em Content ts

Para isso de certo, vamos implementar esse método dentro do construtor

```
content.component.ts
src > app > pages > content > content.components > ContentComponent > constructor
  Go to component!
  1 import { Component, OnInit } from '@angular/core';
  2 import { ActivatedRoute } from '@angular/router';
  3
  4 @Component({
  5   selector: 'app-content',
  6   templateUrl: './content.component.html',
  7   styleUrls: ['./content.component.css']
  8 })
 9 export class ContentComponent implements OnInit {
10   photoCover: string = "https://th.bing.com/th/id/OIP.K5a-BshMzPI4GobXZIKW#hEKE?w=3840&h=2160&r=1&pid=ImgDetMain";
11   contentTitle: string = "Implementando";
12   contentDescription: string = "Implementando";
13
14   constructor(private route: ActivatedRoute) {}
15   ngOnInit(): void { }
16 }
```

Através desse Activated...

Ele tem alguns métodos que podemos implementar

```
private id:string|null="0";
constructor(private route: ActivatedRoute){ }

ngOnInit(): void {
  this.route.paramMap.subscribe(value => this.id = value.get("id"));
}
```

Note que com id dentro dessa classe, conseguimos manipular dentro do OnInit this.id

LEMBRANDO QUE

Como ele irá puxar de uma requisição HTML, o serve retorna um JSON, dessa forma quando chamamos, ele não retornará um ID mas sim um id, pois ele converte tudo em minúsculo (JSON NÃO É CASE SENSITIVE)

Indo em Inspecionar, ele ainda não está implementado dinamicamente os sites ou páginas, no entanto, o ID está puxando OK, então essa parte está OK



Para agora pegar esses dados de forma dinâmica

Necessário implementar algumas classes e entender como elas funcionam

Vamos lá

```

ngOnInit(): void {
  this.route.paramMap.subscribe(value => this.id = value.get("id"));
}

setValuesToComponent(id:string | null){
  const result = dataFake.filter(
    article => article.id == id)[0];
  // this.contentTitle = result.title;
  console.log(result.);
}

```

O que esta dentro de ON INIT faz o id buscar algum valor (id)

Esse setValues... Ele faz o Id ser do tipo string ou nulo, dessa forma, quando quisermos buscar tais informações

Ele deve fazer um filtro em dataFake colocando o que quer filtrar e a posição do array

Dessa forma, eliminaremos o que foi setado de forma statica e criamos uma consulta ao "banco"

No banco ficará dessa forma

```

1 export const dataFake = [
2   {
3     "id": "1",
4     "photo": "PHOTO ID 1",
5     "title": "Salu Novo Filme",
6     "description": "Essa é a descrição para o Id 1",
7     "photoCover": "https://i.pinimg.com/originals/89/91/24/8991248bdc51df737db03bd812eaaad9.jpg"
8   },
9   {
10     "id": "2",
11     "photo": "PHOTO ID 2",
12     "title": "Salu Filme 2",
13     "description": "Essa é a descrição para o Id 2",
14     "photoCover": "https://th.bing.com/th/id/OIP-KSa-8sim0Pf4ghx2TIdw4ufCw-3840h-2160rs-1&pid=ImgGetMain"
15   },
16   {
17     "id": "3",
18     "photo": "PHOTO ID 3",
19     "title": "Salu Filme 3",
20     "description": "Essa é a descrição para o Id 3",
21     "photoCover": "https://www.olibreria.com/images/contentid/policy:1.879759:172988831/Zendaya-como-MU-e-Tom-Holland-como-Homen-Aranha.PNG?f=2x&s=f-5ca74d7de-750&h=21569bb"
22   },
23   {
24     "id": "4",
25     "photo": "PHOTO ID 4",
26     "title": "Salu Filme 4",
27     "description": "Essa é a descrição para o Id 4",
28     "photoCover": "https://th.bing.com/th/id/OIP-KSa-8sim0Pf4ghx2TIdw4ufCw-3840h-2160rs-1&pid=ImgGetMain"
29   }
30 ]
31 
```

Agora, implementando isso dessa forma no content component

```

18 export class ContentComponent implements OnInit {
19   constructor(private route : ActivatedRoute){}
20
21   ngOnInit(): void {
22     this.route.paramMap.subscribe(value => this.id = value.get("id"));
23   }
24
25   setValuesToComponent(id:string | null){
26     const result = dataFake.filter(
27       article => article.id == id)[0];
28     this.contentTitle = result.title;
29     this.contentDescription = result.description;
30     this.photoCover = result.photo;
31   }
32 
```

Note que a parte que eswtava marcado, com informações staticas agora esta respondendo de forma dinamica

```

export class ContentComponent implements OnInit {
  photoCover:string = "";
  contentTitle:string="";
  contentDescription:string = "";
  private id:string|null="#";

```

NO SERVER

Vendo agora como ele ira responder no servidor, se tudo aparece como esperado
NÃO FUNCIONOU, MAS POR QUE EU TAVA COM SONO E ESQUECI DE ADICIONAR ONTEM O

METODO CRIADO

Mas so percebi pois estava procurando onde o metodo de filtro estava sendo chamado
E não encontrei em lugar nenhum

```

ngOnInit(): void {
  this.route.paramMap.subscribe(value => this.id = value.get("id"));

  this.setValuesToComponent(this.id);
}

```

LinkedIn • Github • Youtube

<< VOLTAR



Finalizando, agora deu tudo certo, ate adicionei uma Footer

MARVEL BLOG

Vingadores Guerras Secretas
Maisres heróis e vilões do Universo Marvel são transportados até um mundo alternativo criado por uma entidade de poder inconcebivelmente conhecida como Beyonder para se digladiarem em combates letais. Ao vencedor, será concedido tudo aquilo que desejar.

Homem-Aranha 4: Marvel divulga data de estreia do filme com Tom Holland e Zendaya

novo filme de X-Men em produção

Nova Serie Anunciada

Quando clicando em uma das notícias,

Homem-Aranha 4: Marvel divulga data de estreia do filme com Tom Holland e Zendaya

Essa é a descrição para o id 2

Cada um dos ids estão puxando uma informação

Pois imagine vc ter 30000 informações dentro de um site, ter um Alibaba ou Shopee, ele organiza tudo de forma dinâmica, não exatamente dessa forma, mas com algum tipo de identificador para puxar de forma dinâmica

Última coisa, para adicionar um novo título em vez de AngularBlog



Podemos trocar através de INDEX.HTML

Para um novo ícone em vez do ícone do angular

Podemos trocar dentro de FAVICON.ICO

Como ele está no padrão Angular, não tem nada setado lá, mas é lá o ícone que podemos usar como da página



A screenshot of a code editor showing the `index.html` file. The file contains the following code:

```
1  <!doctype html>
2  <html lang="en">
3  <head>
4  |   <meta charset="utf-8">
5  |   <title>Marvel Blog</title>
6  |   <base href="/">
7  |   <meta name="viewport" content="width=device-width, initial-scale=1">
8  |   <link rel="icon" type="image/x-icon" href="favicon.ico">
9  </head>
10 <body>
11 |   <app-root></app-root>
12 </body>
13 </html>
```

A tooltip is visible over the `href="favicon.ico"` line, reading "Follow link (ctrl + click)".