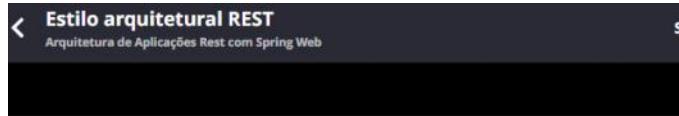


1 Arquitetura REST

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



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);
}
```

IntelliJ IDEA suggestion menu:

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

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 DEREM 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 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*/ 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 aplicação 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

```
GET http://localhost:8080/actuator/health
```

Params Authorization Headers (7) Body Pre-request

Query Params

KEY

Key

idy Cookies (1) Headers (11) Test Results

Pretty Raw Preview Visualize 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

The screenshot shows the Spring Boot Admin web application interface. The top navigation bar includes tabs for 'Wallboard', 'Applications', 'Journal', 'About', and language selection ('en'). The main left sidebar has a green header 'spring-boot-application 1b361bc86277' and a 'Metrics' tab selected. The main content area displays two tables of metrics:

Metric	Value	Count	Total Time	Max
http.server.requests	(no tags)	313	10.7344323	1.8560513
jvm.buffer.count	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

Deployment Descriptor: Sring-boot-admin
  - JAX-WS Web Services
  - src/main/java
    - br.com.dio.admin.Sring.boot.admin
      - ServletInitializer.java
      - SpringBootAdminApplication.java

3>import org.springframework.boot.SpringApplication;
4>
5>
6 @Configuration
7 @EnableAutoConfiguration
8 @EnableAdminServer
9 public class SpringBootAdminApplication {
10
11     public static void main(String[] args) {
12         SpringApplication.run(SpringBootAdminApplication.class, args)
13     }
14
15 }
16

```

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 according Spring Boot Starter. In this example we're using the servlet web starter.

1. 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>
```

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 lanza 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 é um CONTROLLER, só um controller

Quando esse autoconfig está injetado, ele pega também todas as classes necessárias 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 a Medium homepage displaying several articles. The visible titles include "The Standards Innovation Paradox" by Frank Islam & Ed Oregan, "The Tale of Three Americas: Revolution, Evolution, Devolution." by Frank Islam & Ed Oregan, "Beyond Fake Toughness" by Brad Stulberg, and "I Asked an AI Robot to Generate 200-Year-Old Woodcuts of Mount Diablo" by Thomas Smith in The Bolditalic. On the right side of the screen, the browser's developer tools are open, specifically the DOM inspector, showing the hierarchical structure of the HTML code for one of the articles. The DOM tree shows various elements like `<div>`, ``, and `` nested within each other.

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 code editor on the left and a browser window on the right. The code editor displays the following HTML 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>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>
```

The browser window shows the rendered HTML. The title is "PORTAL DE NOTICIAS GALACTICA". The content area has a dark blue background with a grid pattern. On the left, there is a card containing a photo of Andre Felipe, a heading "Darth Vader Contrata Devs", and a paragraph about Darth Vader looking for developers who know how to work with components. On the right side of the card, there is an image of a Star Wars TIE fighter cockpit.

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">
index.html # card.css 1 ...
index.html > html > head
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>
```

Por default o navegador acaba adicionando margens, borda, tamanho

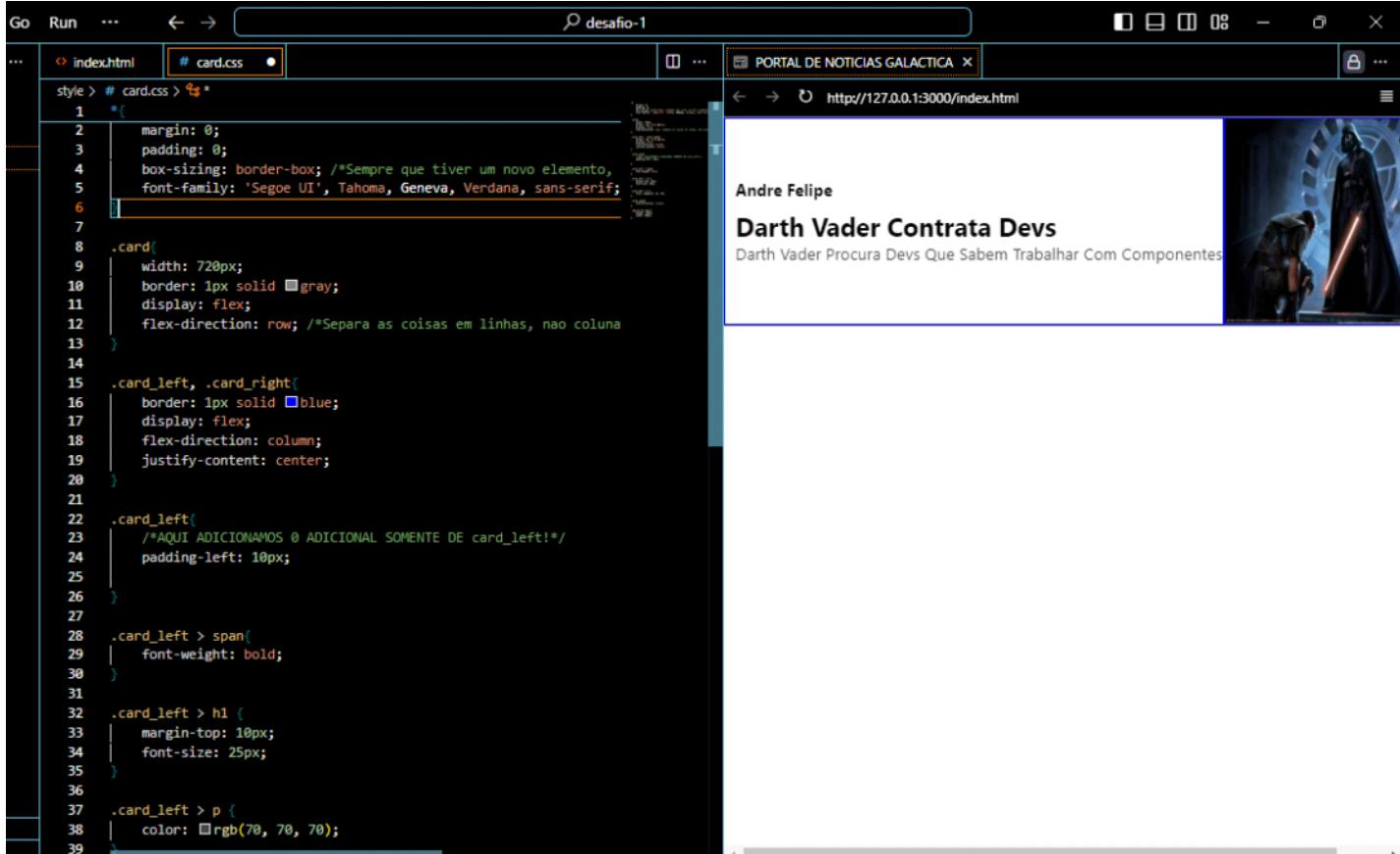
VAMOS RESETAR ISSO UTILIZANDO CSS

Em CSS

```
index.html # card.css ...
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 necessário, em blocos e direções

```
display: flex;
```



```
Go Run ... ← → desafio-1
index.html # card.css ...
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  }

7

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

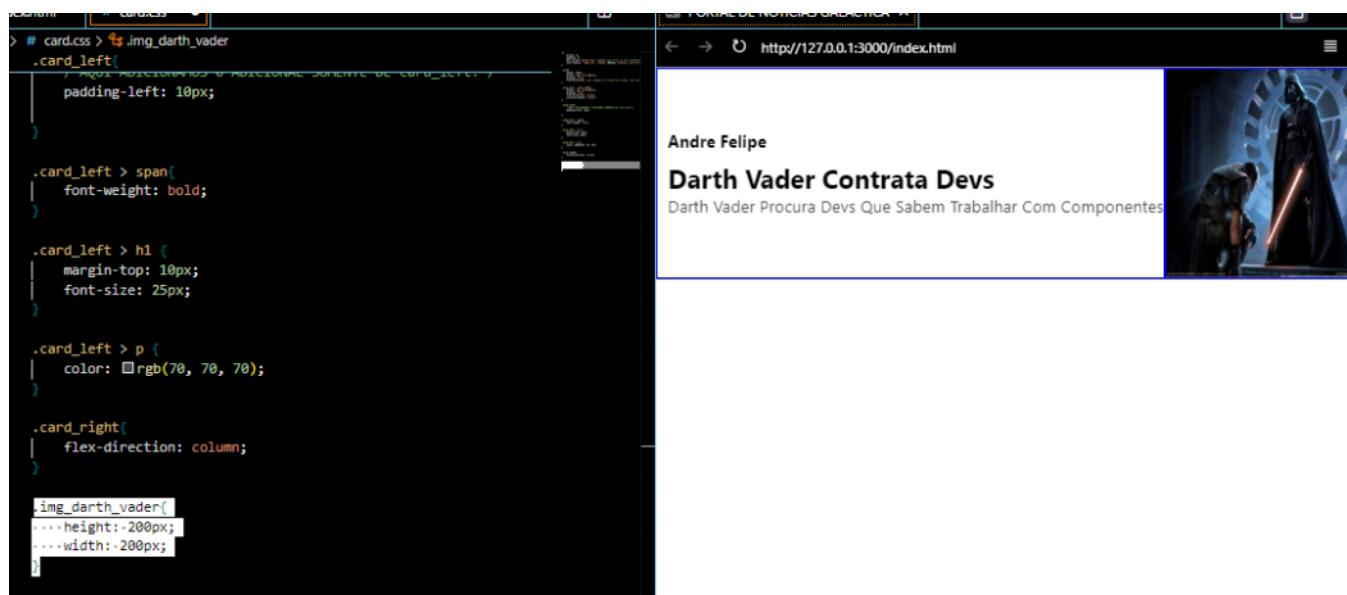
31

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 }
```

Podemos alterar até o tamanho de uma imagem com CSS



```
> # card.css > .img_darth_vader
  .card_left{
    padding-left: 10px;
  }

  .card_left > span{
    font-weight: bold;
  }

  .card_left > h1 {
    margin-top: 10px;
    font-size: 25px;
  }

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

  .card_right{
    flex-direction: column;
  }

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

Para alguns efeitos especiais, temos algumas ferramentas a nosso dispor
CSS GENERATOR, através 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.

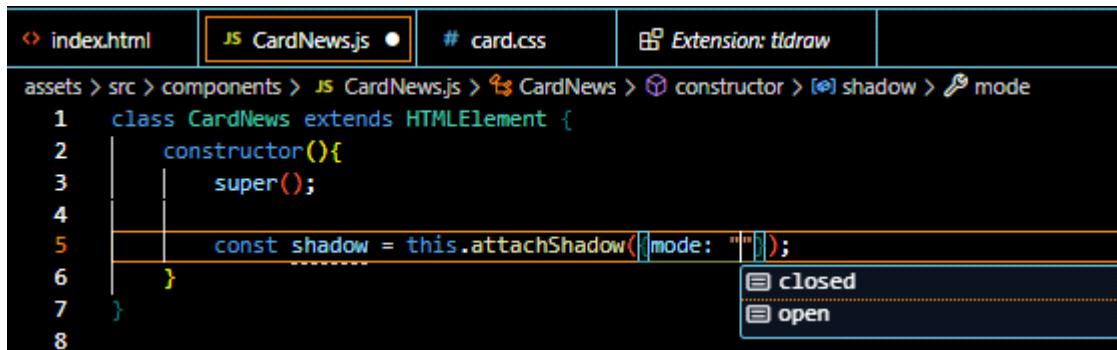
index.html:

```
<!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
</div>
<div class="card_right">
  
<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 'TituloDinamico.js'. The 'TituloDinamico.js' tab is active and displays 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
28 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 'TituloDinamico.js'. The 'index.html' tab is active and displays 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        <titolo-dinamico></titolo-dinamico>
13    </body>
14
15 </html>
```

13,8 Criando PROPS

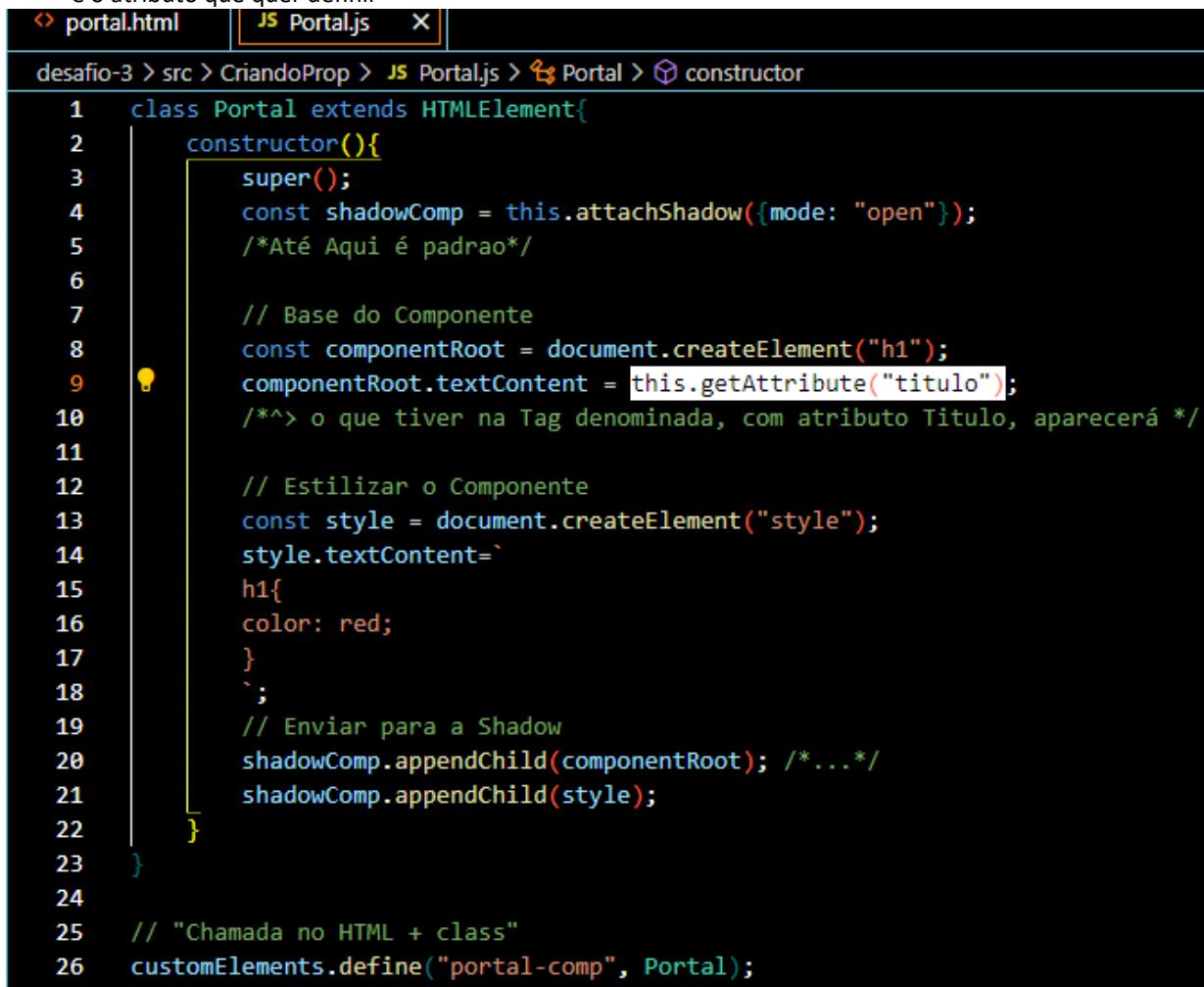
sábado, 14 de setembro de 2024 22:43

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

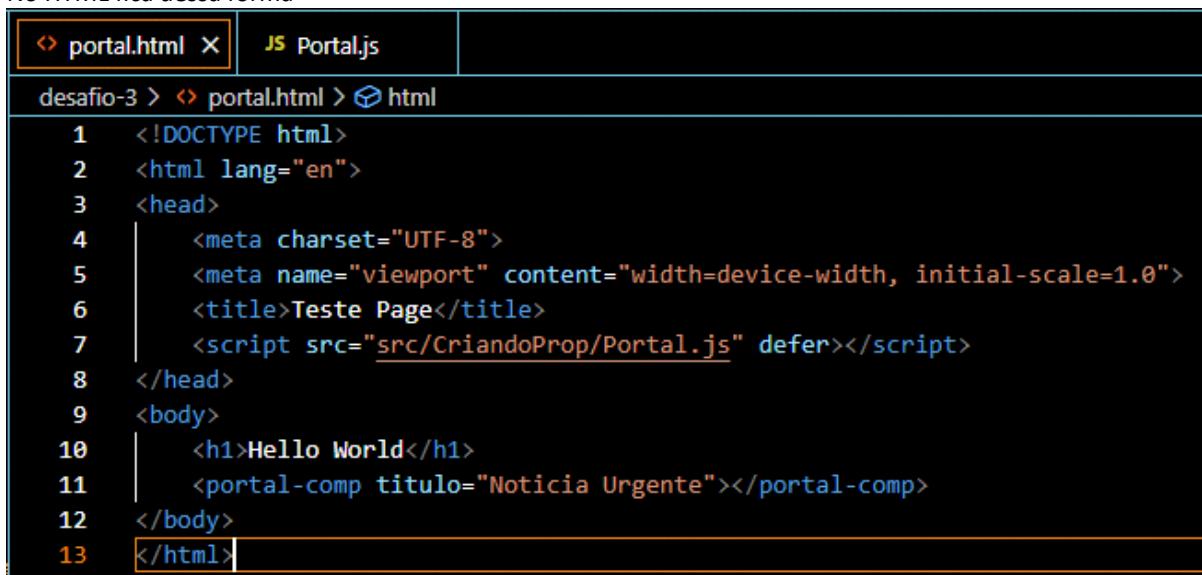
e o atributo que quer definir



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

```
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



The screenshot shows a code editor with two tabs: 'portal.html' and 'JS Portal.js'. The 'portal.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>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>
```

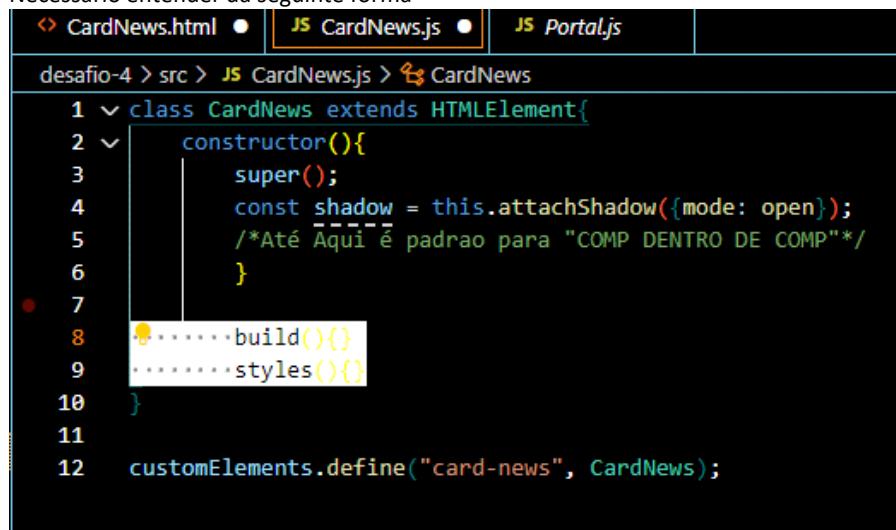
ENTAO AGORA COM UMA NOVA CLASSE

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

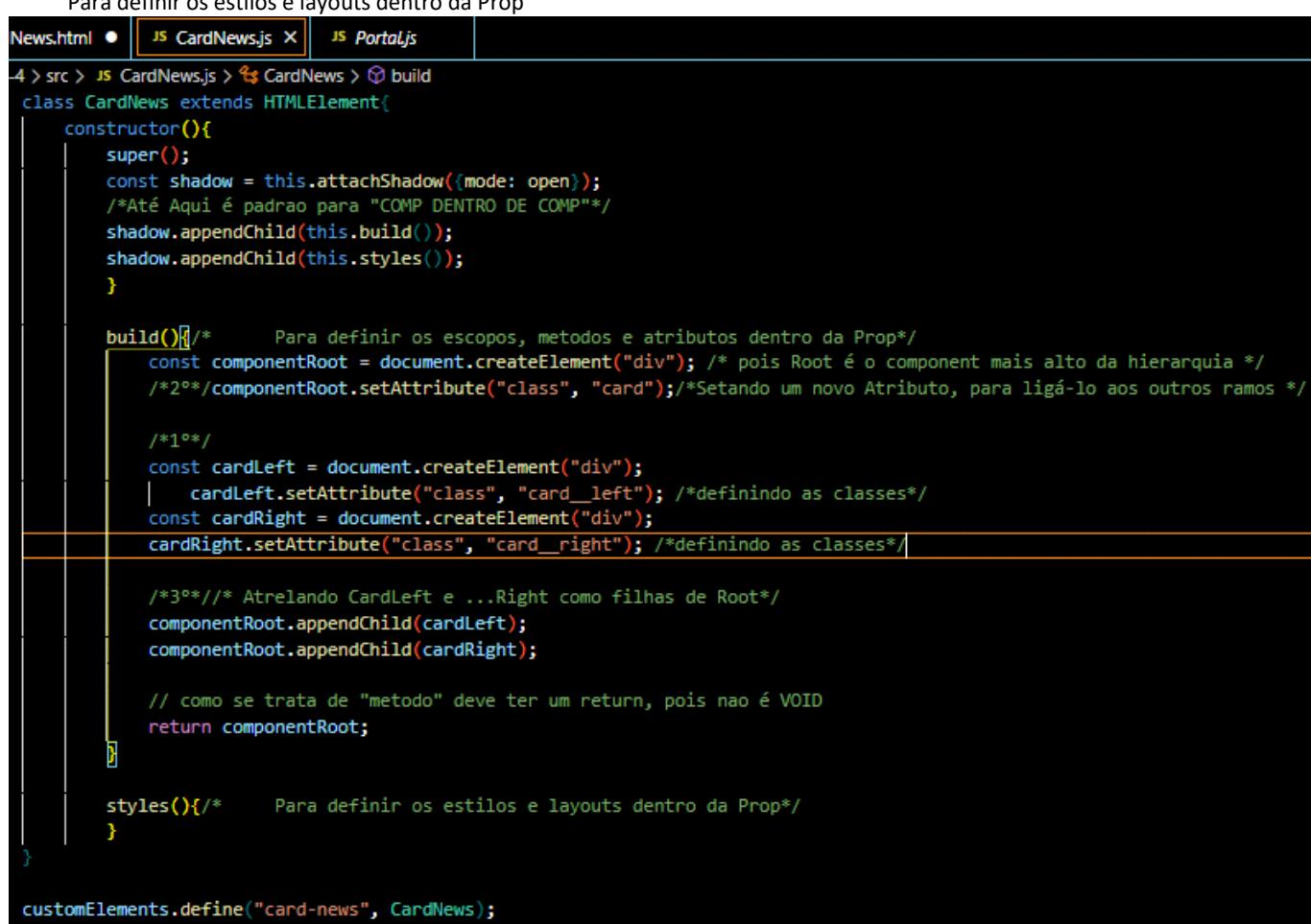


The screenshot shows a code editor with three tabs: 'CardNews.html' (disabled), 'JS CardNews.js' (selected), and 'JS Portal.js'. The 'CardNews.js' tab contains the following code:

```
1 < class CardNews extends HTMLElement{  
2   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



The screenshot shows a code editor with three tabs: 'News.html' (disabled), 'JS CardNews.js' (selected), and 'JS Portal.js'. The 'CardNews.js' tab contains the following code with annotations:

```
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 codigo dentro de Build, como já descrito

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

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

14,1 Aplicando Elementos em Props

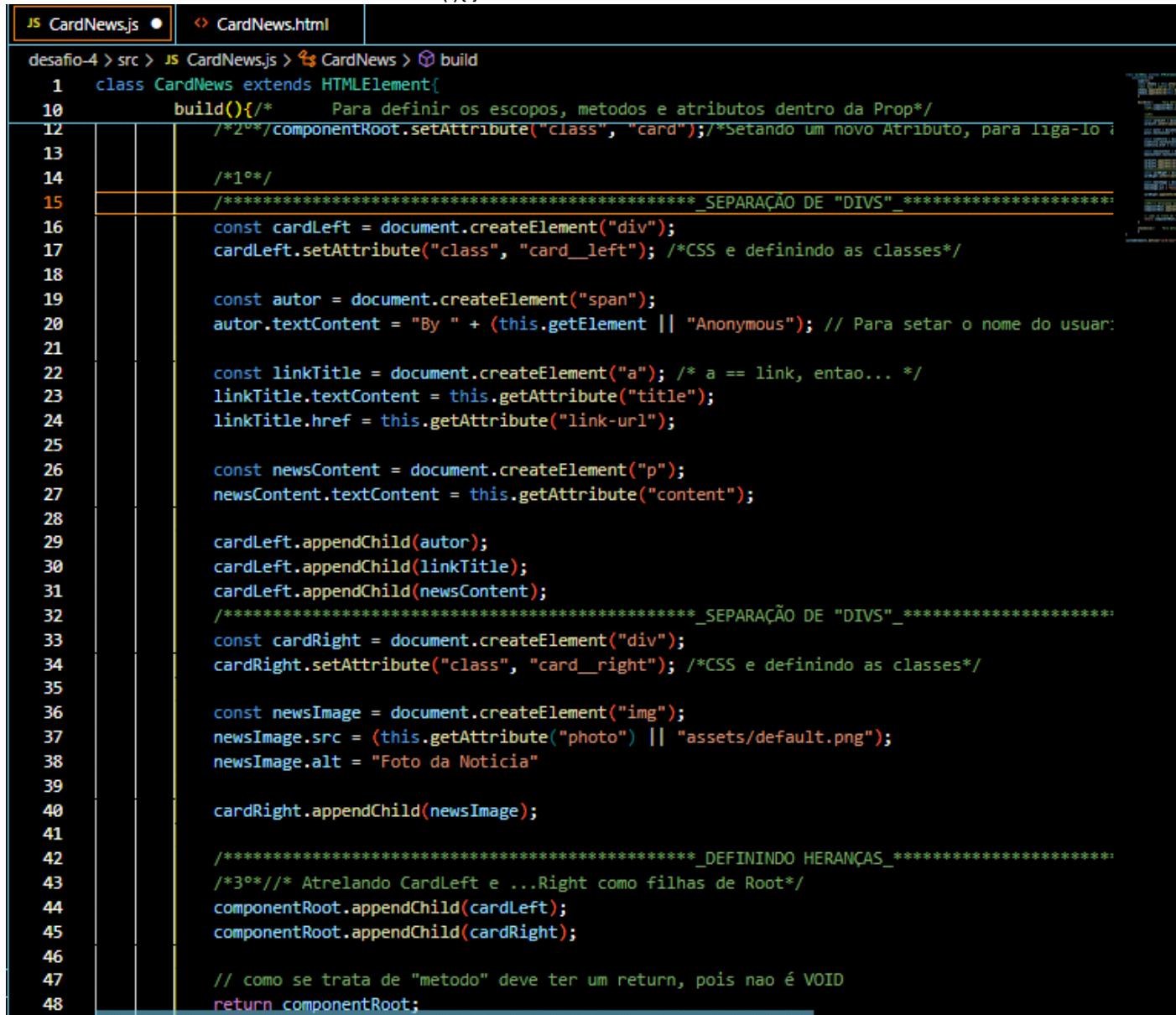
domingo, 15 de setembro de 2024 17:13

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 ATRAVÉS 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      /*2º*/componentRoot.setAttribute("class", "card");/*Setando um novo Atributo, para liga-lo */
13
14      /*1º*/
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, entao... */
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      /*3º// 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

Conseguirá ver tudo de maneira Clean

Então 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

JS CardNews.js	CardNews.html	# CardNews.css
----------------	---------------	----------------

```
desafio-4 > src > CardNews.js > Cardnews > build
  1  class Cardnews extends HTMLElement{
  51    styles()/*      Para definir os estilos e layouts dentro da Prop*/
  54    .card{
  55      width: 80%; /*720px valor real*//*100% porcentagem independente tamanho da tela*/
  56      display: flex;
  57      flex-direction: row; /*Separa as coisas em linhas, nao colunas*/
  58      justify-content: space-between;
  59      box-shadow: 10px 8px 4px 6px rgba(0,0,0,0.28);
  60      -webkit-box-shadow: 10px 8px 4px 6px rgba(0,0,0,0.28);
  61      -moz-box-shadow: 10px 8px 4px 6px rgba(0,0,0,0.28);
  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    }
```

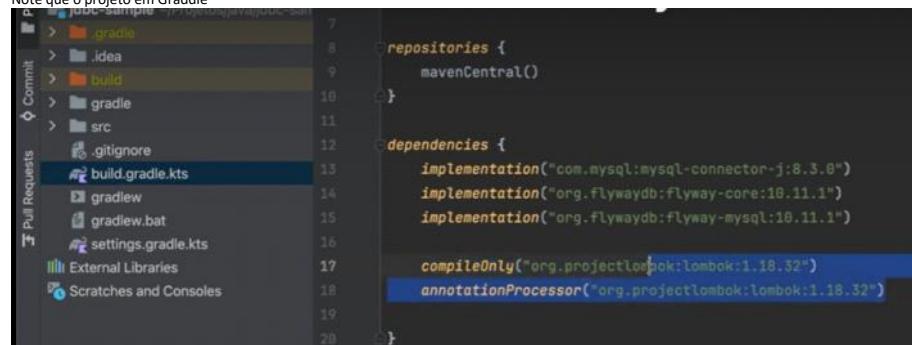
2 JDBC: Explorando Persistência de Dados com Java

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

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

Como

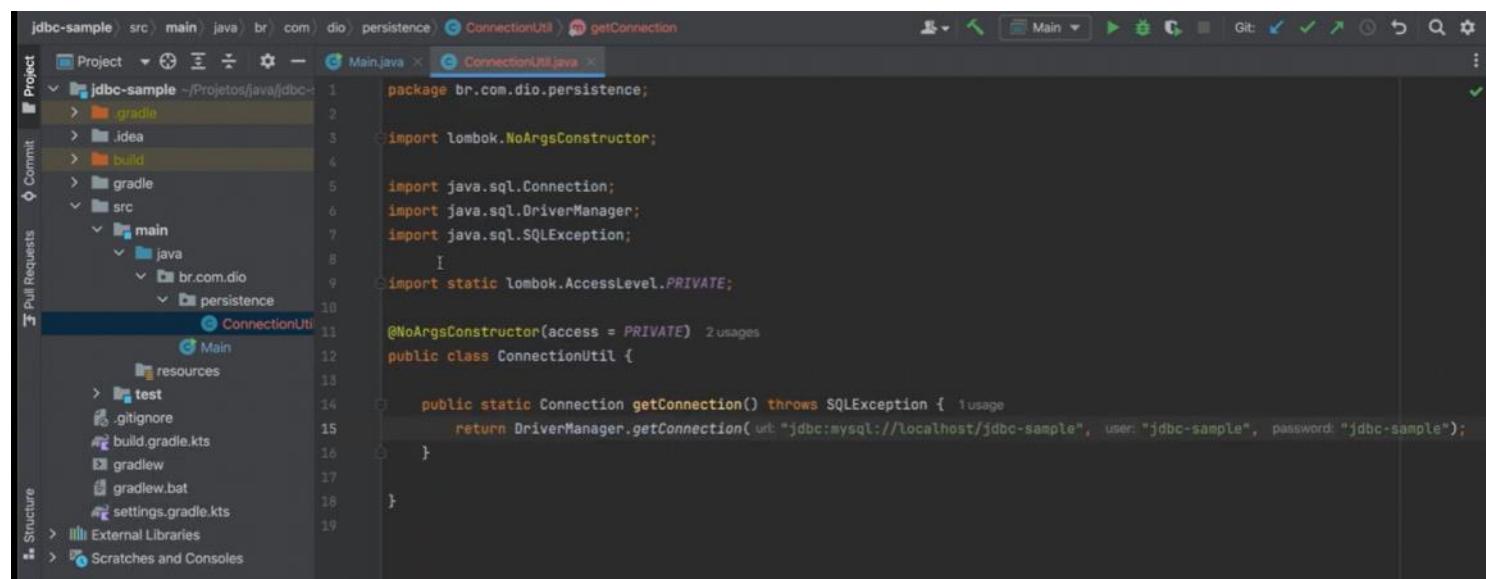
Note que o projeto em Graddle



```
repositories {
    mavenCentral()
}

dependencies {
    implementation("com.mysql:mysql-connector-j:8.0.3")
    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 Connectivity

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 Statement

Sql Injection - Injeta a consulta de forma segura

Statement

ResultSet - 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 (Postgres 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

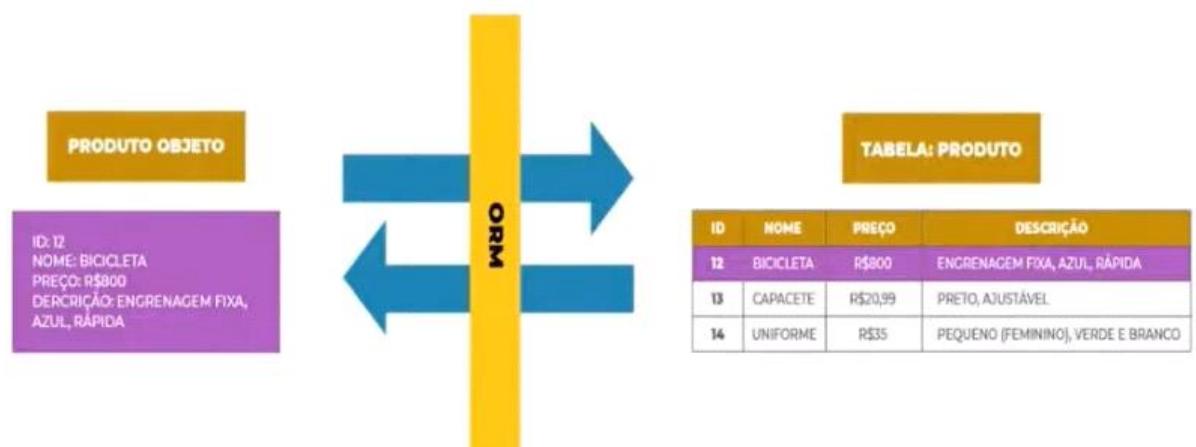
Em vez de ficar rodando While toda vez para ficar persistindo no banco e rodando a aplicação

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

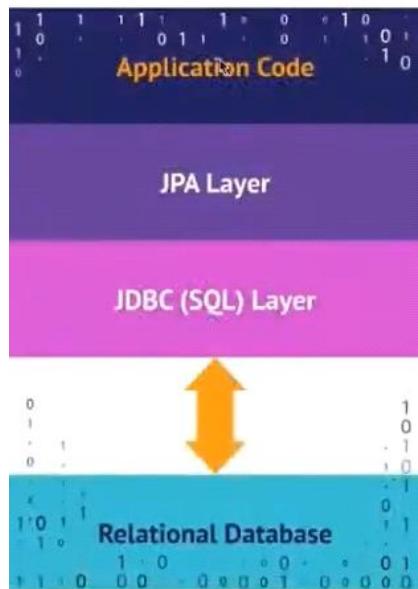
Descreve interface para persistencia em Banco de Dados

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

JPA



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, POSTGRESSQL OU OUTRO)

Dentro de RESPURCES, 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 code editor. A red box highlights the 'resources/META-INF/persistence.xml' path in the project tree. The code editor displays the XML configuration for a persistence unit named 'Clientes_PU'. The configuration includes properties for database connection (URL, user, password), JDBC driver ('com.mysql.cj.jdbc.Driver'), and Hibernate dialect ('org.hibernate.dialect.MySQL8Dialect'). It also includes settings for SQL output ('show_sql=true') and connection autocommit ('autocommit=true').

```

<?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/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>

```

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

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

The screenshot shows the 'persistence.xml' file with specific configurations for MySQL. The URL is set to 'jdbc:mysql://localhost/db_cadastro_cliente'. The user is 'root' and the password is 'root'. The JDBC driver is 'com.mysql.cj.jdbc.Driver' and the dialect is 'org.hibernate.dialect.MySQL8Dialect'. Other properties like 'show_sql=true' and 'autocommit=true' are also present.

```

<?xml version="1.0" encoding="UTF-8"?>
<persistence xmlns="http://xmlns.jcp.org/xml/ns/persistence" http://xmlns.jcp.org/xml/ns/persistence/pers... 30 ^ ...
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    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 Usuário e senha, um em seguida do outro, não mais como inserido na aula JDBC

O driver MySQL ou se utilizar outro, o driver dele

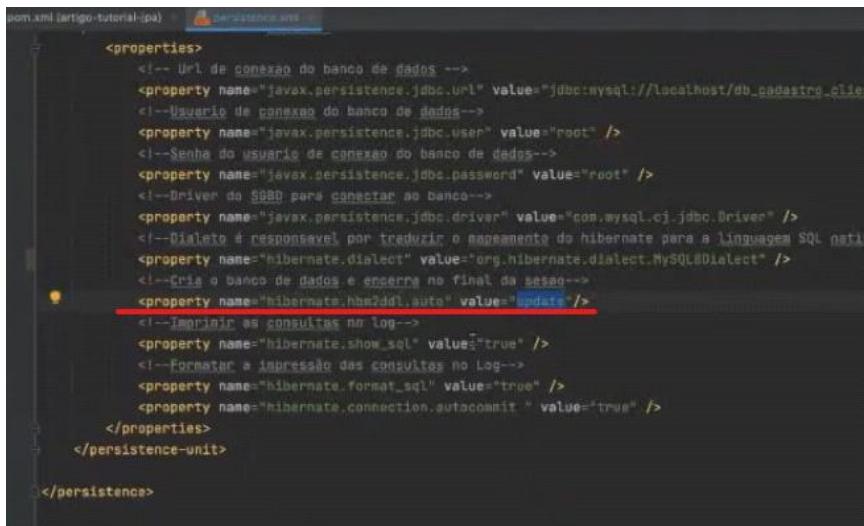
Entre outros

The screenshot shows the 'persistence.xml' file again, identical to the previous one. It specifies the MySQL URL, user 'root', password 'root', driver 'com.mysql.cj.jdbc.Driver', and dialect 'org.hibernate.dialect.MySQL8Dialect'. The properties section includes 'show_sql=true' and 'autocommit=true'.

```

<?xml version="1.0" encoding="UTF-8"?>
<persistence xmlns="http://xmlns.jcp.org/xml/ns/persistence" http://xmlns.jcp.org/xml/ns/persistence/pers... 30 ^ ...
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    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>

```



```
pom.xml.jartigo-tutorial-(pa)  persistence.xml

<!-- 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](https://refactoring.guru/pt-br/design-patterns)

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

Minico Dependências

Vantagens

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

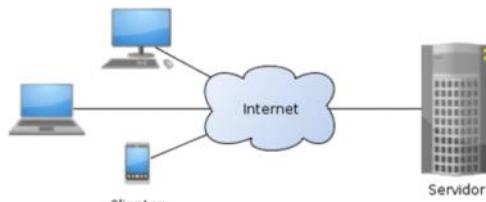
4 Cliente Servidor

sexta-feira, 27 de setembro de 2024

09:24

Modelo Cliente-Servidor

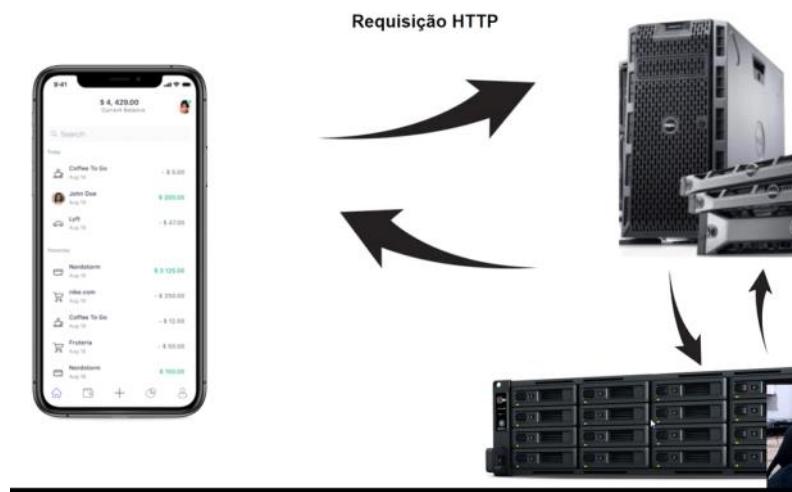
O modelo cliente-servidor é uma estrutura de aplicação que distribui as tarefas e cargas de trabalho entre os fornecedores de um recurso ou serviço, designados como servidores, e os requerentes dos serviços, designados como clientes.



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



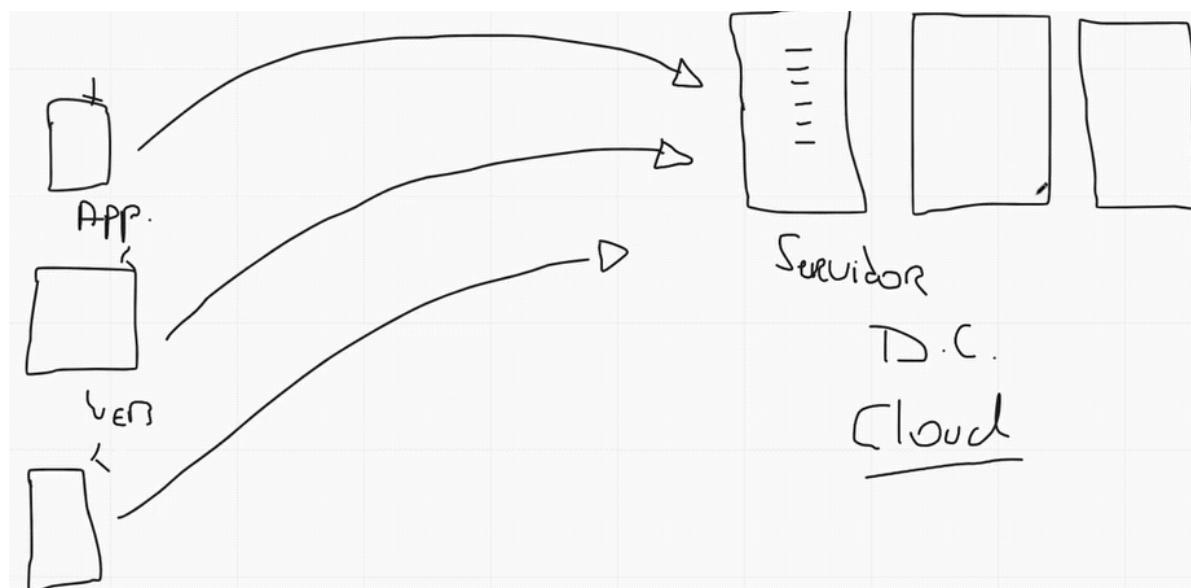
Quando é uma conversa/comunicação entre dispositivos

Os 2 precisam de uma identificação

Enviam a info p servid. o servid. 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

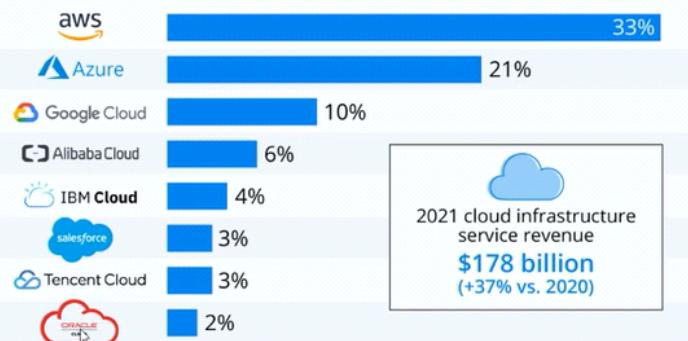
A cloud computing é o acesso sob demanda, via internet, a recursos de computação — aplicativos, servidores (físicos e virtuais), armazenamento de dados, ferramentas de desenvolvimento, recursos de rede e muito mais — hospedados em um data center remoto gerenciado por um provedor de serviços em cloud (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

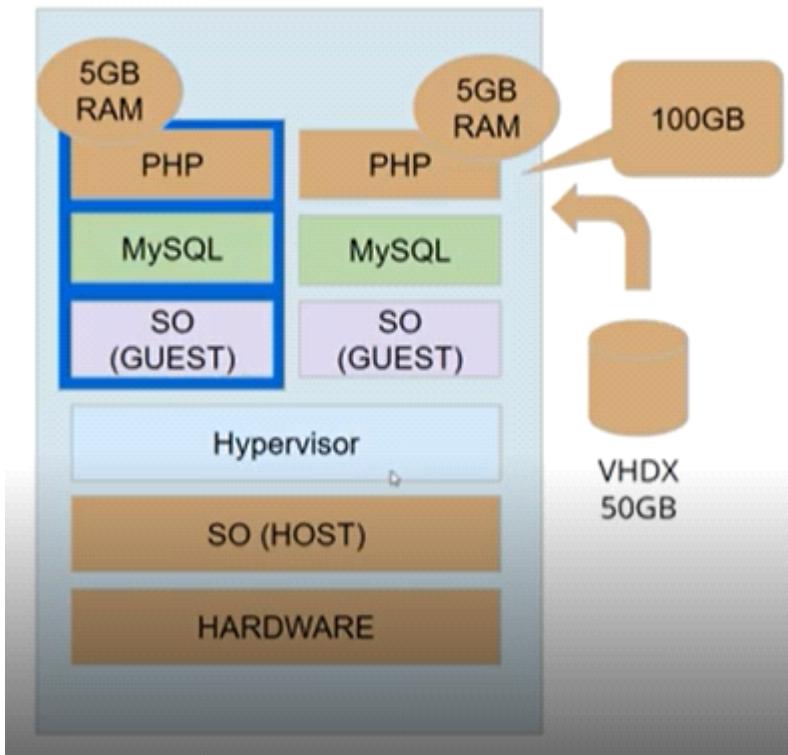
A virtualização utiliza software para criar uma camada de abstração sobre o hardware do computador, permitindo que os recursos de hardware de um único computador (processadores, memória, armazenamento, etc) sejam divididos em vários computadores virtuais.

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

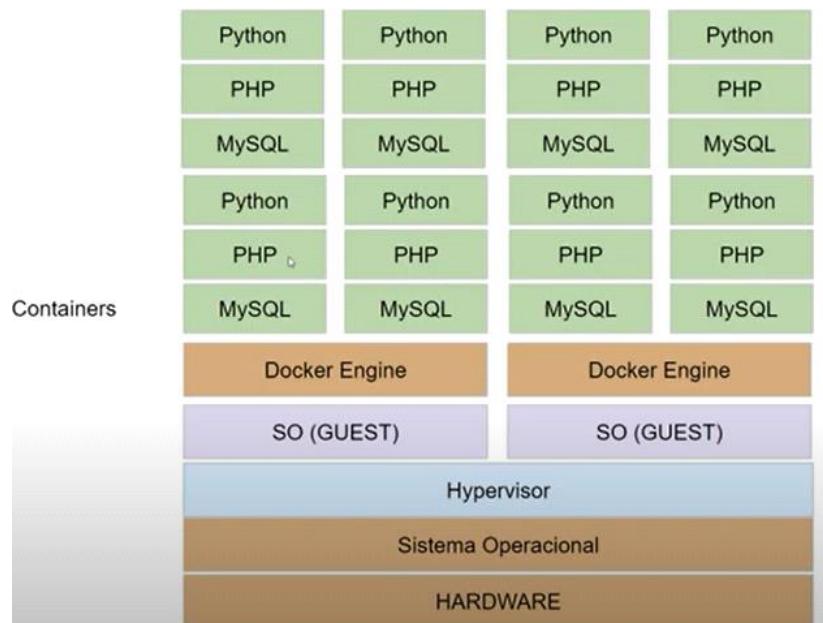
Eles podem ser feitos e montados em OVA

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 microservcos podem acontecer em coponentes



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

This section describes how to install Docker Engine on Linux, also known as Docker CE. Docker Engine is also available for Windows, macOS, and Linux, through Docker Desktop. For instructions on how to install Docker Desktop, see:

- [Docker Desktop for Linux](#)
- [Docker Desktop for Mac \(macOS\)](#)
- [Docker Desktop for Windows](#)

Supported platforms

Platform	x86_64/amd64	arm64/aarch64	arm(32-bit)	ppc64le	s390x
CeOS	✓	✓		✓	
Debian	✓	✓	✓	✓	
Fedora	✓	✓		✓	
Raspberry Pi OS (32-bit)			✓		
RHEL	✗	✗		✓	
SLES				✓	
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: 21.3M (peak: 24.0M)  
     CPU: 2.192s  
    CGroup: /system.slice/docker.service  
           └─2113 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd  
  
set 29 05:11:08 ubuntu-server dockerd[2113]: time="2024-09-29T05:11:08.4543110672"  
set 29 05:11:08 ubuntu-server dockerd[2113]: time="2024-09-29T05:11:08.4581997842"  
set 29 05:11:08 ubuntu-server dockerd[2113]: time="2024-09-29T05:11:08.48123617282"  
set 29 05:11:10 ubuntu-server dockerd[2113]: time="2024-09-29T05:11:10.4063429162"  
set 29 05:11:10 ubuntu-server dockerd[2113]: time="2024-09-29T05:11:10.5211607262"  
set 29 05:11:10 ubuntu-server dockerd[2113]: time="2024-09-29T05:11:10.5215889622"  
set 29 05:11:10 ubuntu-server dockerd[2113]: time="2024-09-29T05:11:10.5217038232"  
set 29 05:11:10 ubuntu-server dockerd[2113]: time="2024-09-29T05:11:10.5219961932"  
set 29 05:11:10 ubuntu-server systemd[1]: Started docker.service - Docker Application  
set 29 05:11:10 ubuntu-server dockerd[2113]: time="2024-09-29T05:11:10.7173347552"  
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/>

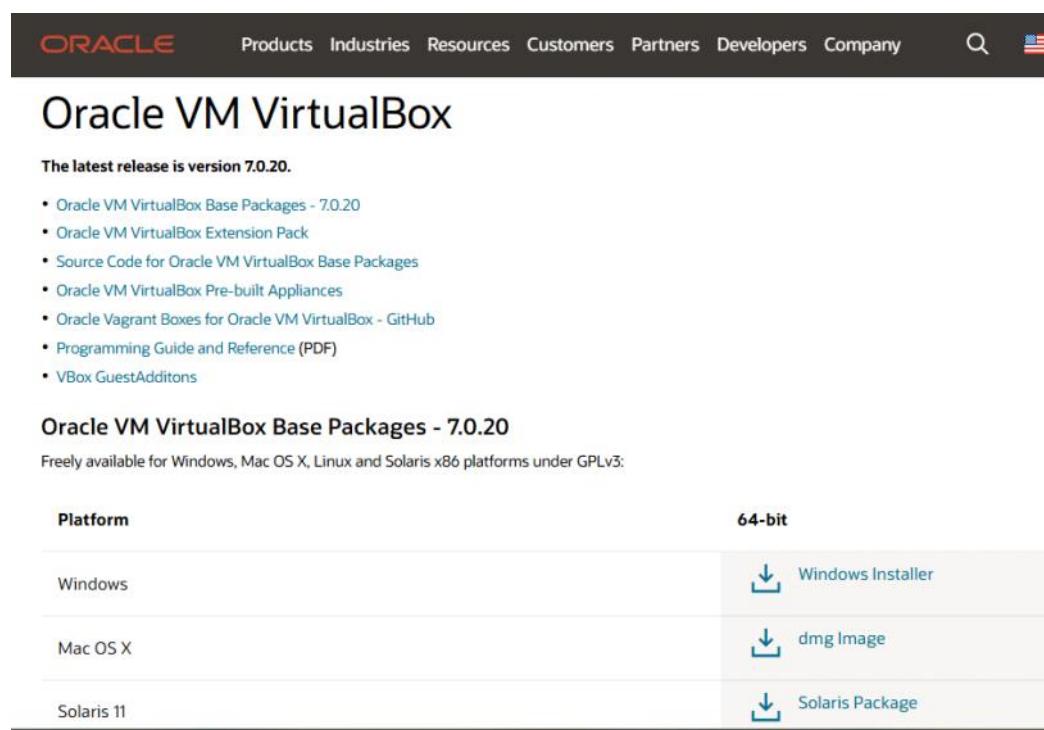
[VM VirtualBox | Virtualização | Oracle Brasil](#)

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 a US flag icon. Below the navigation is the title "Oracle VM VirtualBox". A note says "The latest release is version 7.0.20.". Below that is a 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)
- VBox GuestAdditions

Under "Oracle VM VirtualBox Base Packages - 7.0.20", it says "Freely available for Windows, Mac OS X, Linux and Solaris x86 platforms under GPLv3:". There's a table for "64-bit" packages:

| 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 Ubuntu Server download page. At the top, there's a navigation bar with links like 'Products', 'Use cases', 'Support', 'Community', 'Get Ubuntu', 'All Canonical', and 'Sign in'. Below the navigation is a menu with 'Downloads' (highlighted in orange), 'Desktop', 'Server', 'Core', and 'Cloud'. The main content area has a heading 'Get Ubuntu Server' and three tabs: 'Manual installation', 'Instant VMs', and 'Automated provisioning' (underlined). Below this is a section for 'Ubuntu 24.04.1 LTS' featuring a red crown icon. To the right, there's a brief description of LTS, a 'Download 24.04.1 LTS' button (2.6GB), and links for 'Alternative downloads' and 'Alternative architectures'. Further down are sections for 'What's new', 'System requirements', and 'How to install', each with several bullet points.

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

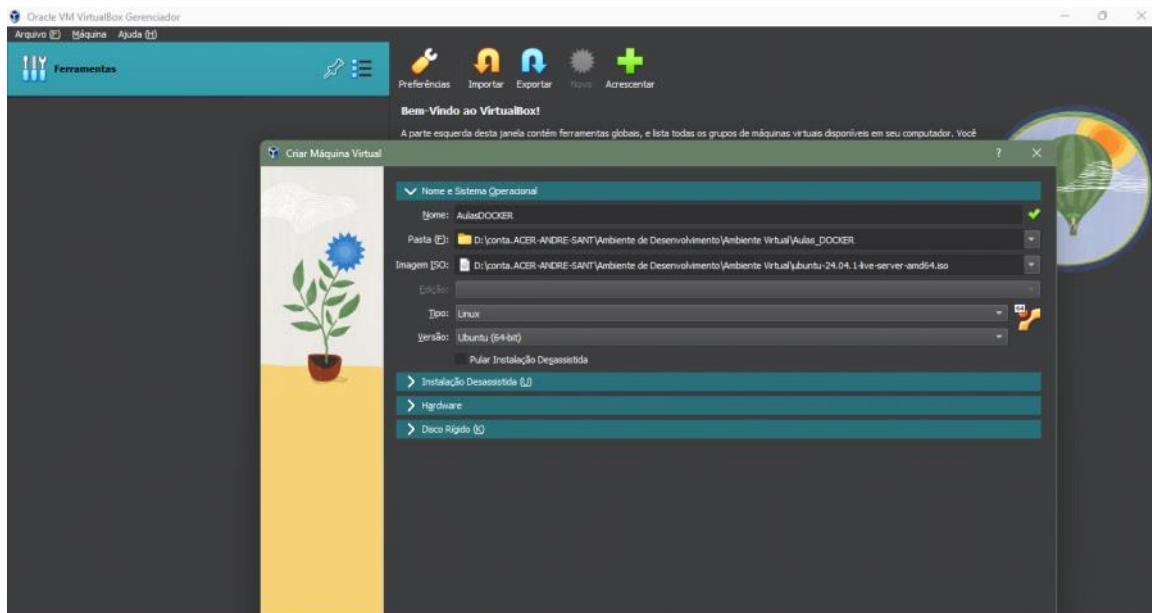
Rust 1.75, .NET 8 and OpenJDK 21 with TCK certification in addition to other toolchain upgrades

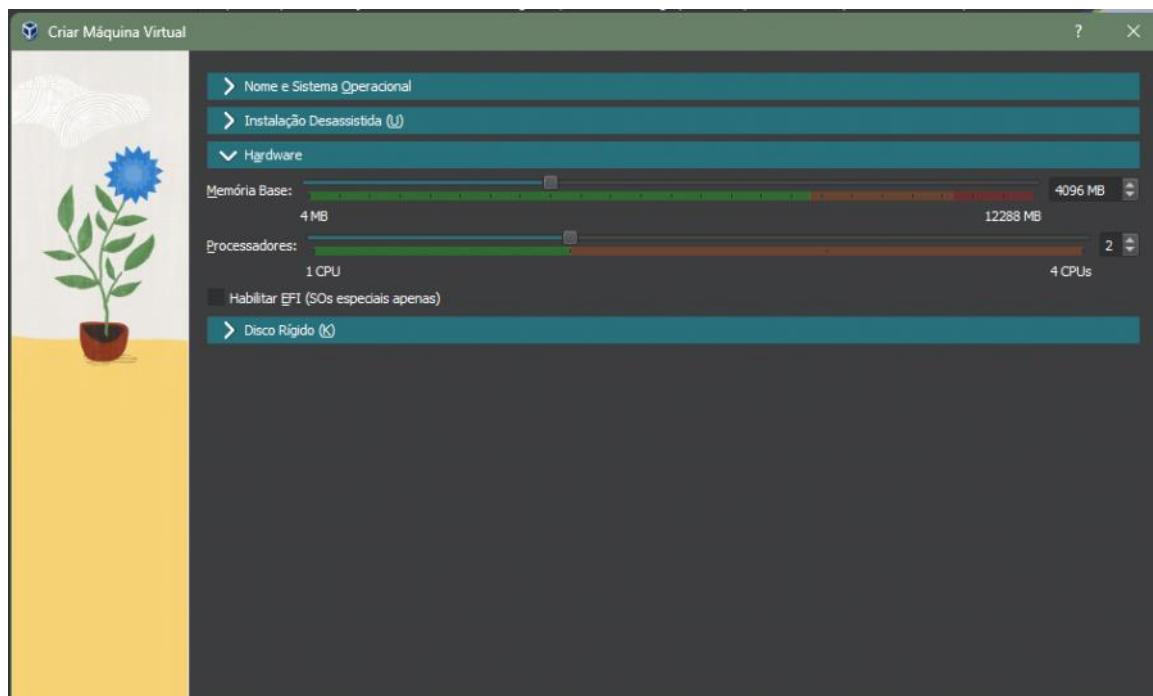
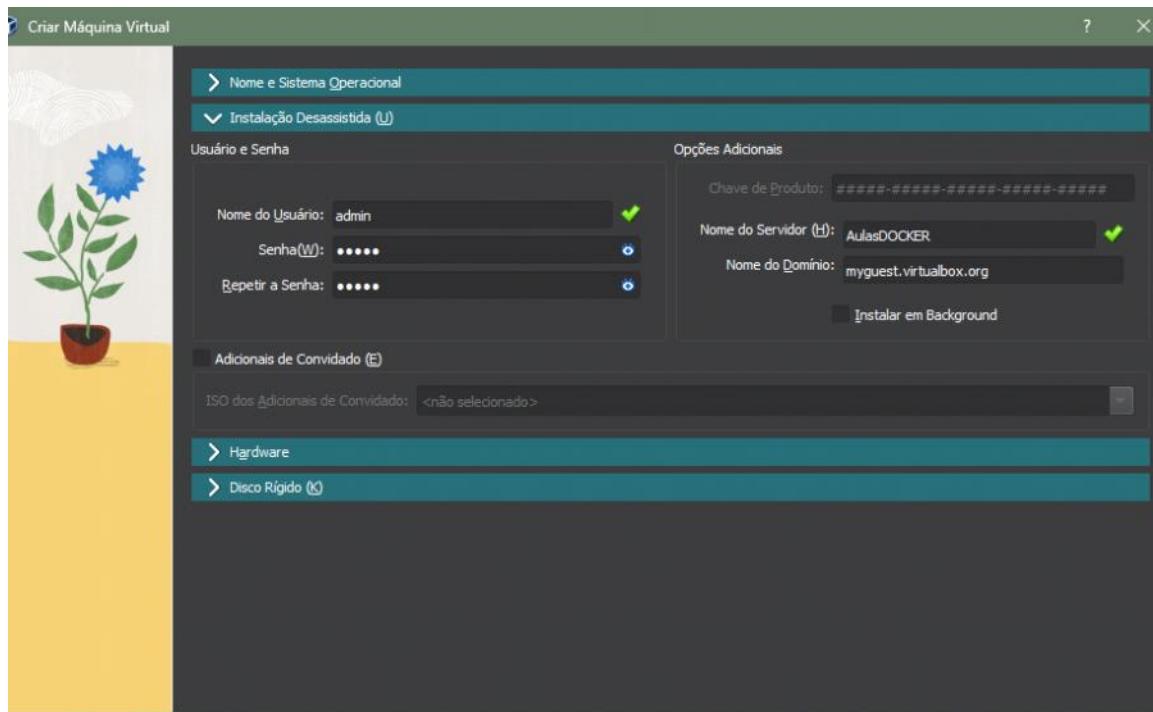
64-bit time_t by default on armhf to address the year 2038 issue

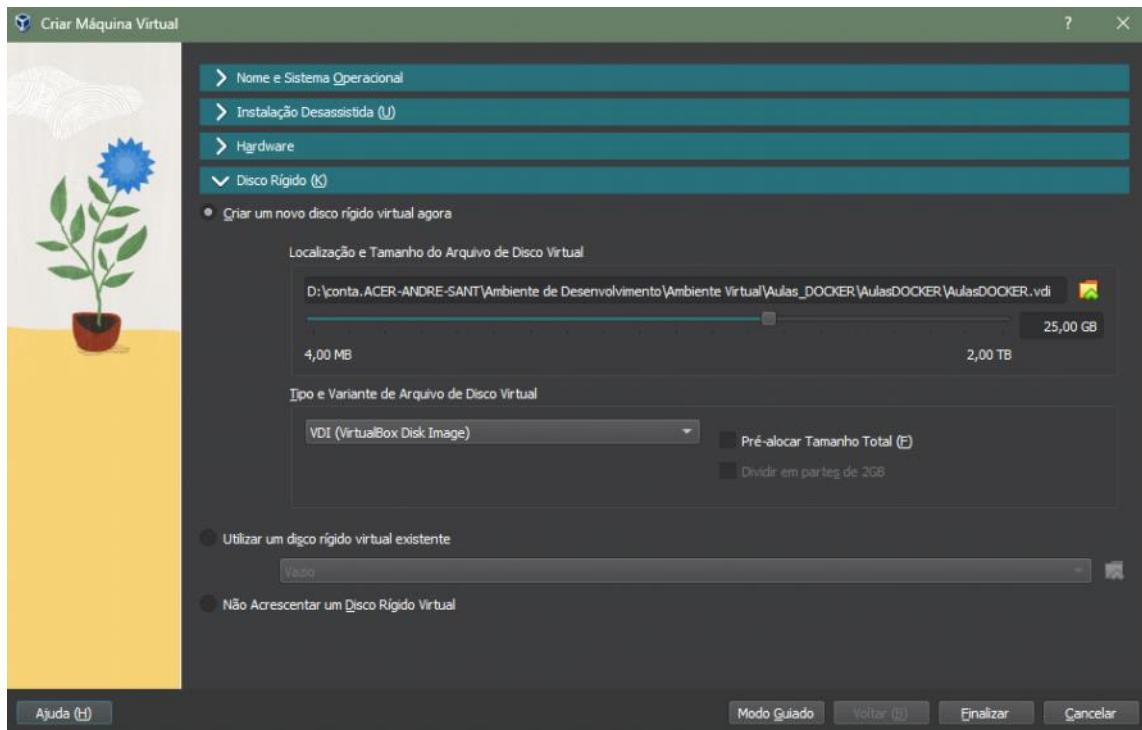
[server/thank-you?version=24.04.1&architecture=amd64<s=true](#)

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

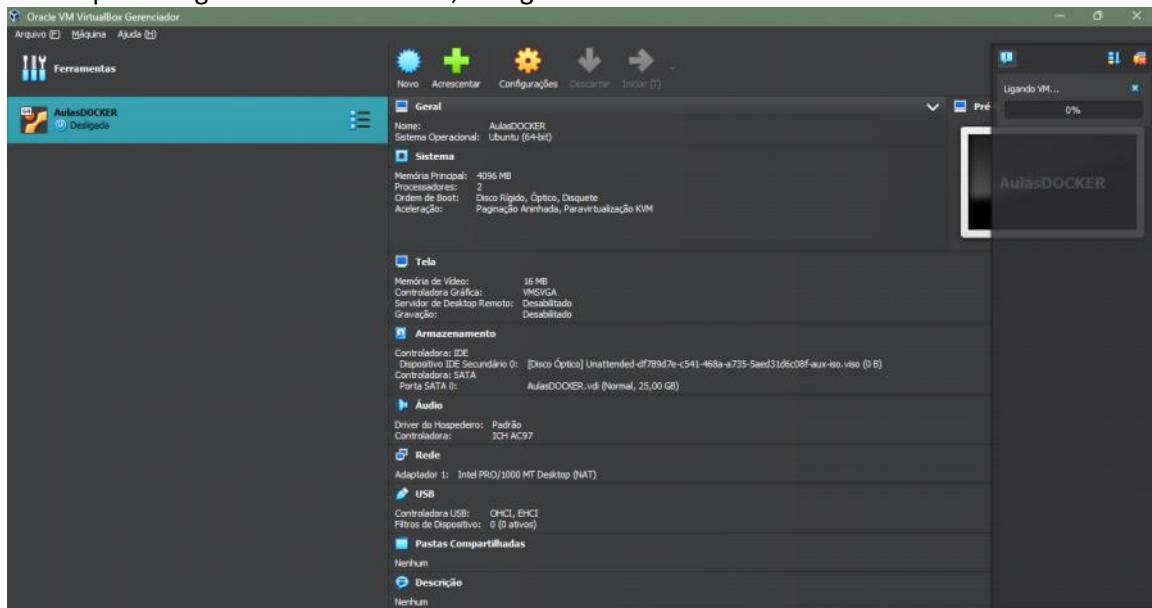
Clicando em NEW







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

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 tambem 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 page title is "15,2 - Instalando e Configurando o TS". Below it, the date and time are shown as "segunda-feira, 7 de outubro de 2024 17:49". A heading "Para instalar o TS, podemos entrar no site do TS" is followed by a link "Clicando em Download". The main content area has three columns:

- Handbook**: "Learn the language" with a right-pointing arrow.
- Playground**: "Try in your browser" with a right-pointing arrow.
- Download**: "Install TypeScript" with a right-pointing arrow.

Each column contains descriptive text and a screenshot of the TypeScript playground interface.

Get Started

Three cards are displayed side-by-side:

- Handbook**: "Learn the language" with a right-pointing arrow. It features a screenshot of the TypeScript handbook page.
- Playground**: "Try in your browser" with a right-pointing arrow. It features a screenshot of the TypeScript playground interface.
- Download**: "Install TypeScript" with a right-pointing arrow. It features a screenshot of the terminal showing the command `$ npm install + typescript`.

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

The screenshot shows the "TypeScript em seu projeto" section of the documentation. It provides instructions for integrating TypeScript into existing projects:

- via npm**: "O TypeScript está disponível como um pacote no registro npm disponível como `typescript`". It includes a command: `npm install typescript --save-dev`.
- 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`

For projects that don't support NuGet, the extension `TypeScript` is available for Visual Studio. The extension can be installed via the Visual Studio Extensions Manager.

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

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

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\conta.ACER-ANDRE-SANT\Ambiente de Desenvolvimento\DI - 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\DI - Bootcamps\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\DI - Bootcamps\Bootcamp_Springboot_Angular DEAL\Bootcamp_Springboot_Angular DEAL\Aulas-JS>

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

```
caseComJS.js
caseComTS.ts x
Extension: Dracula At Night

Aulas-JS > 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

JS caseComJSs U TS caseComTS.ts I, U ● Extension: Dracula At Night

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.  
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 vulgo: "Capitão América".  
14 );
```

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

```
✓ Aulas-TS \ Aula1-InstallAndTest
  > node_modules
  > src
  {} package-lock.json
  {} package.json
```

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 tabs, there are four buttons: PROBLEMS, OUTPUT, DEBUG CONSOLE, and TERMINAL. The TERMINAL tab is active, showing the following 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

Da 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, código traduzido e depois executado, será o JS

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

Created a new tsconfig.json with: TS

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\DIo - 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 browser tabs. The left tab is titled 'TypeScript Playground' and displays a sample TypeScript code snippet that prints "Hello World". The right tab is titled 'TSConfig Reference' and provides documentation for the '# Files - files' section, which describes how to specify a list of files to include in a build.

Quando gerado esse --init

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

The screenshot shows the 'tsconfig.json' file in a code editor. The file contains numerous comments explaining various compiler options. These comments are highlighted in yellow, indicating they have been removed from the final configuration. The file includes sections for projects, language and environment, modules, and compiler options.

```
2  "compilerOptions": {
3
4    /* Projects */
5    // "incremental": true,          /* Save .tsbuildinfo files to allow for incremental compilation of projects. */
6    // "composite": true,           /* Enable constraints that allow a TypeScript project to be used with projects. */
7    // "tsBuildInfoFile": "./tsbuildInfo", /* Specify the path to .tsbuildInfo incremental compilation file. */
8    // "disableSourceOfProjectReferenceRedirect": true, /* Disable preferring source files instead of declaration files when referring to a project. */
9    // "disableSolutionSearching": true, /* Opt a project out of multi-project reference checking when editing. */
10   // "disableReferencedProjectLoad": true, /* Reduce the number of projects loaded automatically by TypeScript. */
11
12   /* Language and Environment */
13   // "target": "es2016",           /* Set the JavaScript language version for emitted JavaScript and include features specific to that version. */
14   // "lib": [],                  /* Specify a set of bundled library declaration files that describe the target environment. */
15   // "jsx": "preserve",          /* Specify what JSX code is generated. */
16   // "experimentalDecorators": true, /* Enable experimental support for legacy experimental decorators. */
17   // "emitDecoratorMetadata": true, /* Emit design-type metadata for decorated declarations in source files. */
18   // "jsxFactory": "",           /* Specify the JSX Factory Function used when targeting React JSX emit, e.g. 'React.createElement'. */
19   // "jsxFragmentFactory": "",    /* Specify the JSX Fragment reference used for fragments when targeting React. */
20   // "jsxImportSource": "",       /* Specify module specifier used to import the JSX Factory Functions when targeting React. */
21   // "noEmit": 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 declaration. */
36   // "allowImdGlobalAccess": true, /* Allow accessing IMD global access 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 '--module'. */
39   // "resolvePackageJsonExports": true, /* Use the package.json 'exports' field when resolving package imports. */
40   // "removeBackslashRootPrefix": true, /* Remove the backslash from 'imports' field when resolving imports. */

4  "compilerOptions": {
5    // "declarationMap": true,      /* Create sourcemaps. */
6    // "emitDeclarationOnly": true,  /* Only output d.ts files and not JavaScript files. */
7    // "sourceMap": true,           /* Create source map files for emitted JavaScript files. */
8    // "inlineSourceMap": true,     /* Include sourcemap files inside the emitted JavaScript files. */
9    // "noEmit": true,              /* Disable emitting files from a compilation. */
10   // "outFile": "./",            /* Specify a file that bundles all outputs into one. */
11   // "outDir": "./build",         /* Specify an output folder for all emitted files. */
12   // "removeComments": true,      /* Disable emitting comments. */
13   // "importHelpers": true,       /* Allow importing helper functions from tslib once. */
14 }
```

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

ROOT DIR

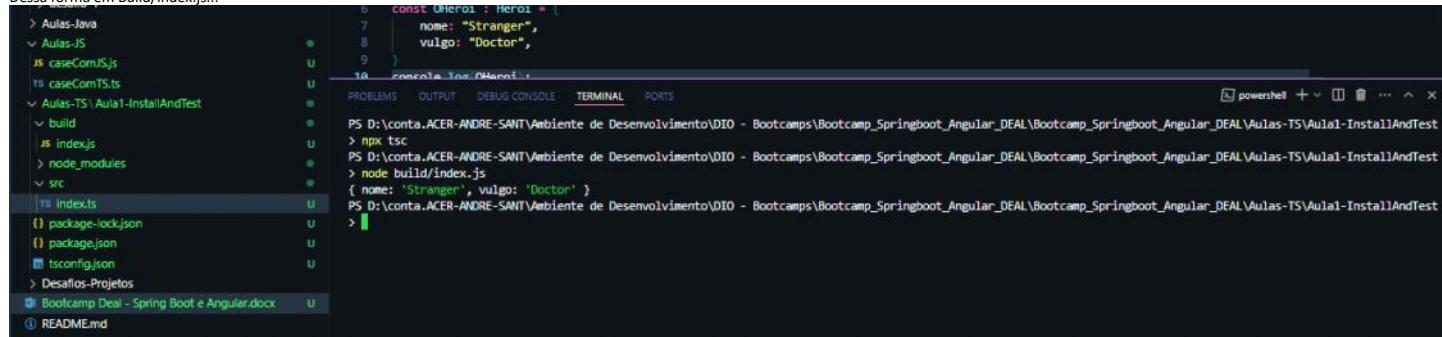
Dessa forma, quando npx tsc src... Se eu não especificar o caminho, por padrão ele vai traduzir tudo de src

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...



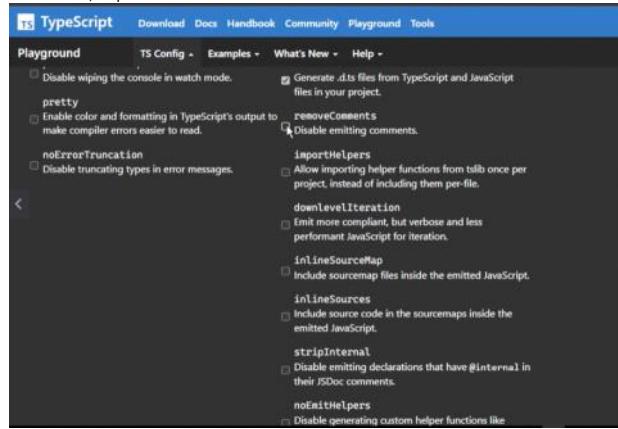
```
const Heroi : Heroi = {
  name: "Stranger",
  vulgo: "Doctor",
}
console.log(Heroi)
```

```
npx tsc
PS D:\conta.ACER-ANDRE-SANT\Ambiente de Desenvolvimento\DI - 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\DI - 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 também, mas é mais informativo para o futuro, da primeira forma



15,2,4 Configurando Node

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

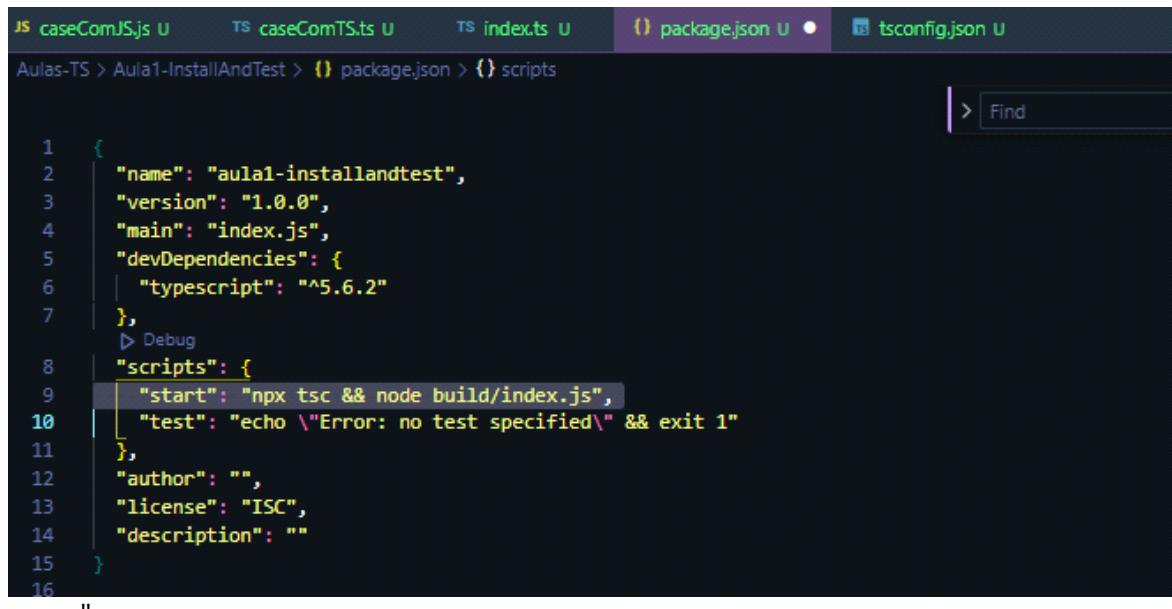
Para deixar ainda mais facil de rodar um projeto

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
```