Angular é modular tem por padrão um módulo que é como um limitador de contexto cujo todos os componentes dentro dele compõem aquele contexto

Casa = módulo pai

Cada cômodo seria um módulo e dentro de cada módulo existem componentes específicos naquele módulo Ex módulo quarto possui cama, guarda-roupa

Package.json – Contém todas as dependências da nossa aplicação

Angular.json – Contém as informações globais da nossa aplicação e ele também vai definir o arquivo que vai inicializar nossa aplicação

Angular segue o padrão single page aplication spa – toda a informação está envelopada em uma única pagina

Main.ts – responsável por inicializar nossa aplicação (fazer o bootstrap) e ele espera receber o modulo raiz para que ele possa fazer toda leitura da aplicação

App.modulo.ts – é a partir dele que ele vai fazer a leitura do componente pai que é o componente responsável por envelopar todos os componentes nossos componentes criados e rodado no nossa index.html

Decoreitor – são propriedades que dão mais informações aos componentes para que o angular possa fazer umas funcionalidades a mais

Oneway databite – apenas exibe e não atualiza a propriedade, ou seja, uma ligação de uma única via caracterizado pelo uso dos []

Twoway databite – junção de 2 vias vai tanto ler quanto alterar o valor da propriedade independentemente de onde ela estiver no nosso componente caracterizado pelo [()]

Modo HTTP é assíncrono

Mesclado os atributos da tag HTML com informação da variável do angular

Colocar a tag entre []

<img [src]="course.ImageUrl">

OnChanges assim que receber o valor do HTTP ele vai passar o valor

**Injeção de dependência**

injeção de dependência é um padrão de projeto no qual uma classe solicita dependências de fontes externas ao invés de criá-las.

Tornando a classe disponível para Injeção de dependência

@Injectable({

provideIn: 'root'

})

Data binding

Data Binding permite com que as tags e atributos peguem o valor de um component.ts.

Set quando for digitar

Get quando for fazer a leitura

Pipes

Pipe nos ajuda a transformar dados que estão em um formato para outro. Eles são chamados assim pois usa o “|” para chamá-lo.

Pode-se utilizar mais de um pipeline ao mesmo tempo

Rotas

Import {RouterModule} from ""

Imports: [

RouterModule.forRoot([

{

Path: '', component/redirect/pathMatch

}

)]

]

Path: '' caminho para raiz

Path: '\*\*' caminho para quando não acha o caminho

Ativando Rotas

Acessando um componente entro de outro componente

[routerLink]= "[' url ', course.id]" class:" "

[ngClass]="{classe, condição}"

Passa classes dinamicamente para nossa propriedade

[Disabled]="variável.invalid" desabilita a opção/botão caso tenha algo(variável) inválido

(Click)=" save()"

Requisição HTTP

Metodos HTTP

Get

Put atualiza a requisição

O padrão de retorno de um HTTP cliente é um observable, que trabalha com contratos

Next: parâmetro => {

Retorno

}

Se a requisição http for efetuada sem problemas será retornado o que estiver no Next

Se der algo errado sera retonado o erro

Error: função => console.log()

Delete

deleteById(id: number): Observable<any> {

}

Declarativa - informa os Pipes e componentes

Import - declara as rotas que nosso módulo tem

Um componente não pode ser declarado em dois módulos diferentes

Shared

Contém componentes que são de nível pequeno/genérico a qual pode mos importar os módulos conforme a nossa necessidade

Core

Contém componentes pesados com regras de negócios mais complexos