# **Angular 8**

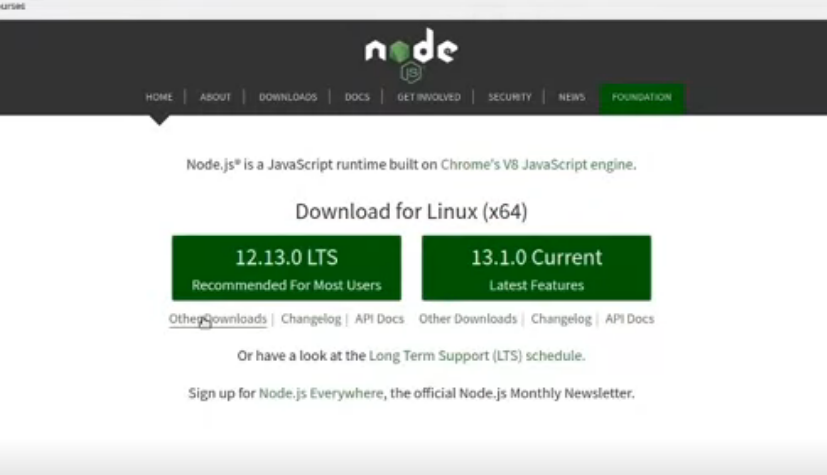
## **Primeiros passos para desenvolver com angular**

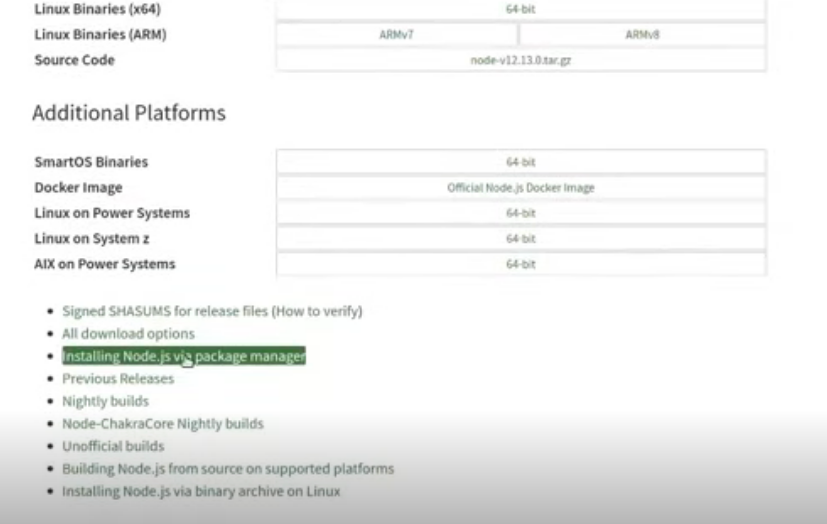
Conhecimento prévio em HTML e JS.

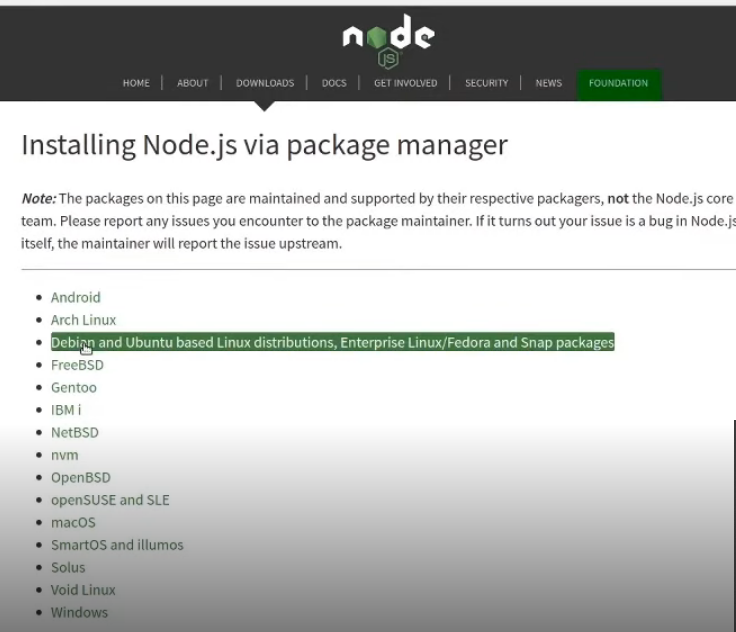
### Instalando e configurando o ambiente

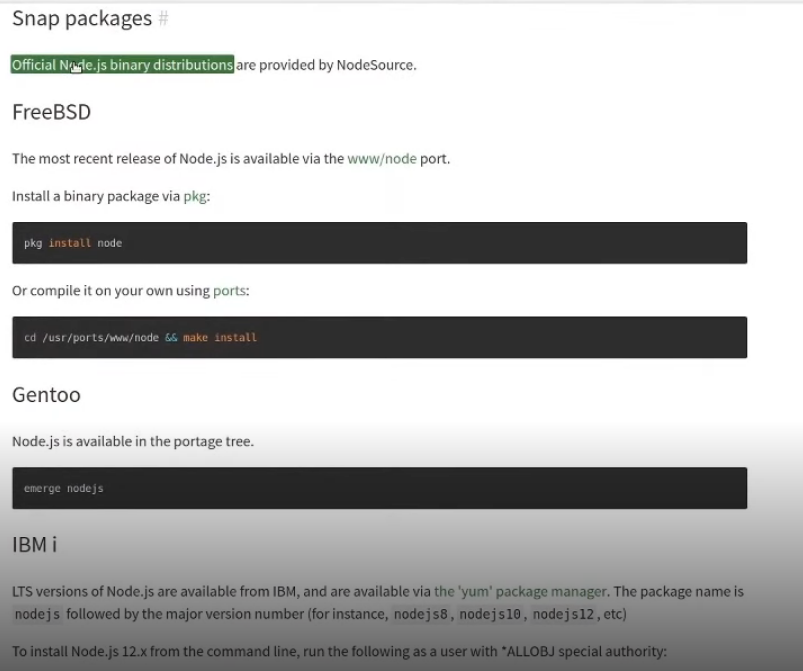
Será necessário a instalação do node.js, angular CLI e Visual Studio Code.

- Instalando o node.js

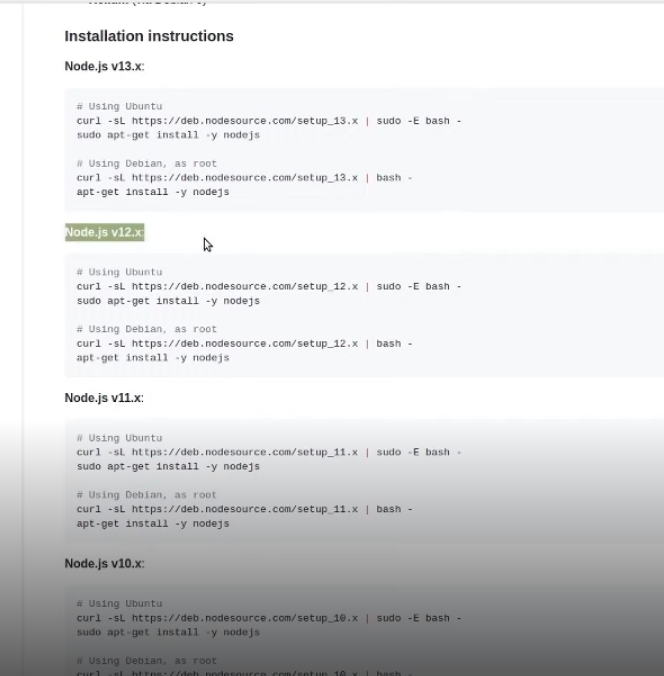








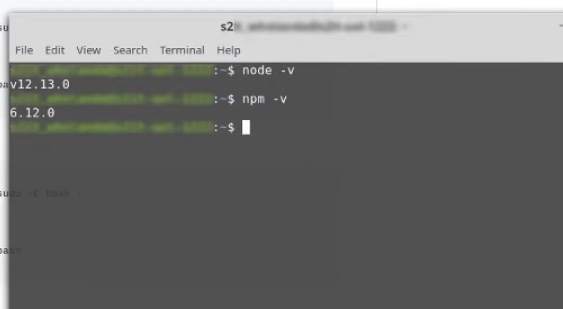
Esse link vai nos jogar dentro do github do node.js e então temos que procurar a versão que estaremos utilizando na lista de todos as versões. Nas aulas será usado o node.js vs12.



Para verificar se as versões que foram instaladas estão corretas digitar esses códigos no terminal

node –v (para saber a versão)

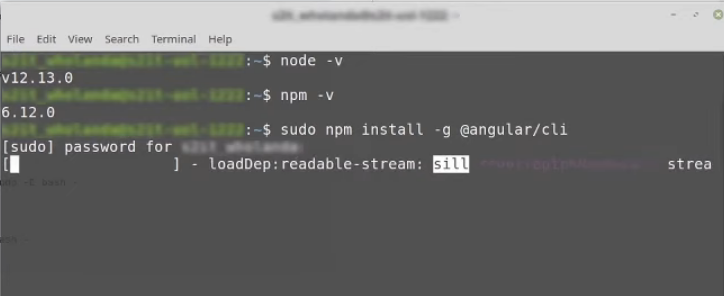
npm –v (para gerenciador de pacotes)



- Instalando o angular direto no terminal digitando o seguinte comando

sudo npm install –g @angular/cli

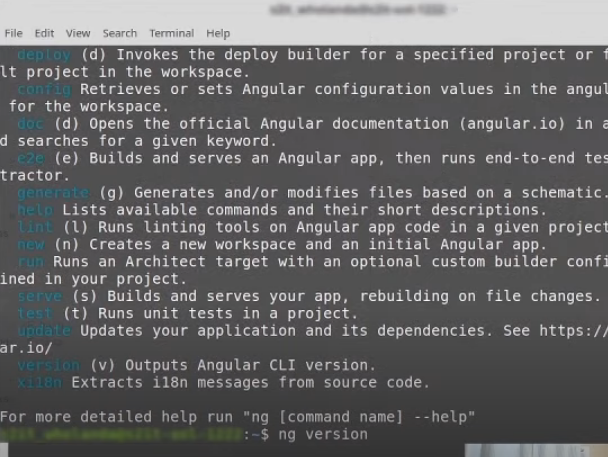
nota: o “g” nesse comando significa que estamos fazendo a instalação de forma global no sistema e não apenas em um diretório.



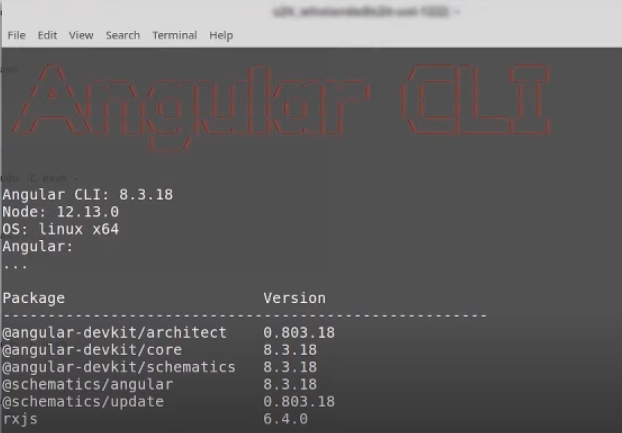
Irá requisitar a senha para começar a instalação.

Para saber a versão que o angular está, digitar o seguinte comando terminal

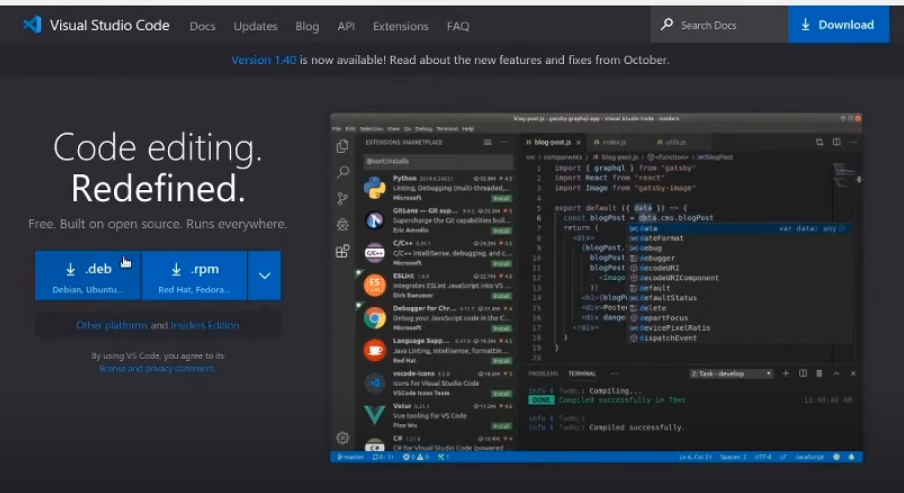
ng version



Vai aparecer a versão instalada na maquina



- Instalar o VS Code

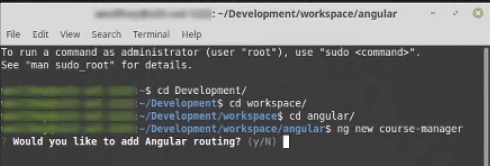


### Criando a primeira aplicação com Angular

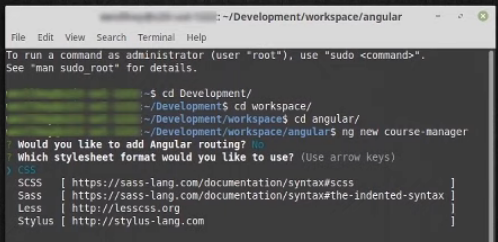
Primeiro passo é escolher a pasta no terminal, depois digitar o seguinte código no terminal

ng new course-manager

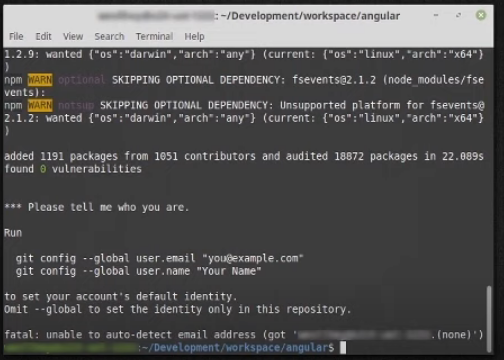
E clicar na opção N para não criar um angular routing



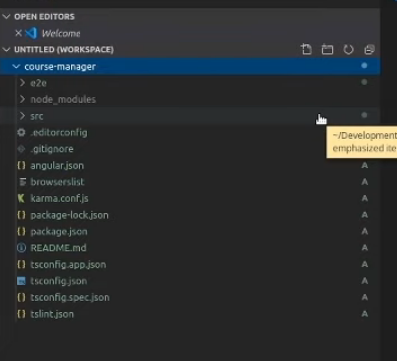
Escolher o CSS para o estilo



Vai realizar a instalação dos pacotes



Quando iniciar o VS Code dentro da pasta que foi criada, vai existir vários arquivos pré-definidos

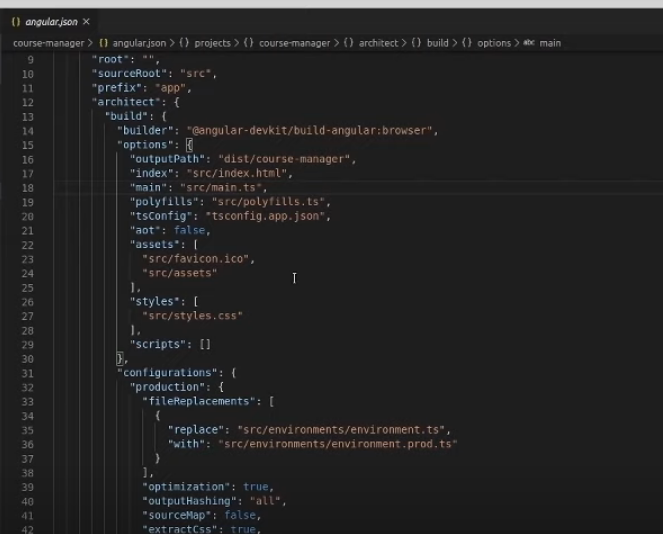


Entro do arquivo angular.json esta discriminado alguns ‘arquivos’

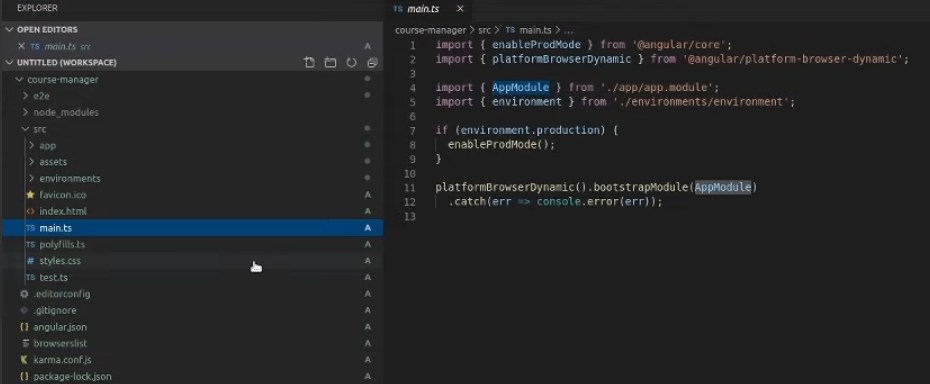
O index será nossa pagina da nossa aplicação

O main é responsável por inicializar a nossa aplicação

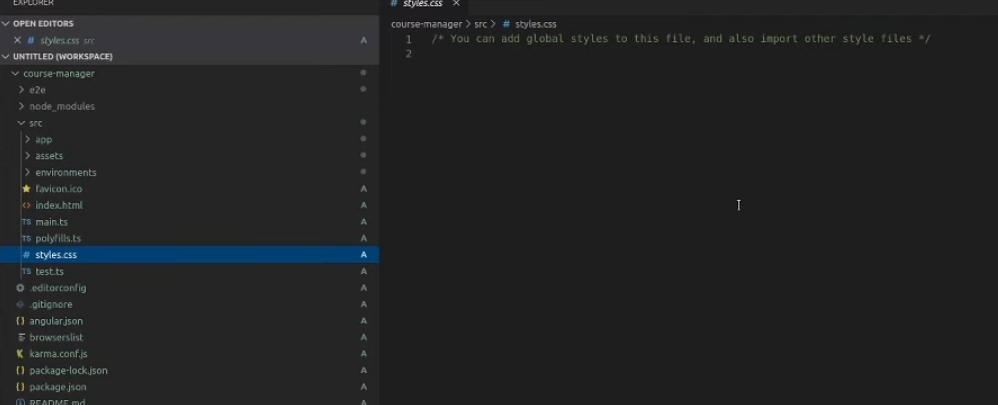
O style é o estilo que será utilizado na nossa página de aplicação



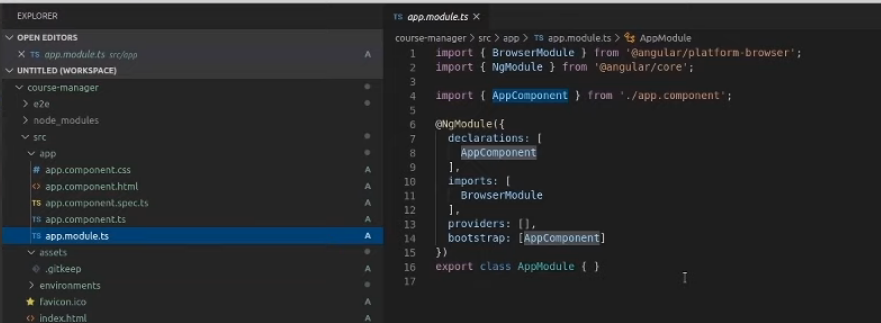
Dentro do main.ts é responsável por toda nossa aplicação, ele recebe um modulo raiz dentro da nossa aplicação, fazendo o processo de bootstrap. Nesse caso ele chama um arquivo chamado AppModule que se encontra dentro da pasta de app.



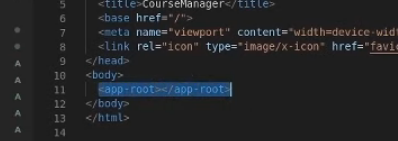
O arquivo style.css será responsável pelo estilo de toda nossa aplicação, se quisermos um estilo atua de forma global, aplicado para todos os componentes da nossa aplicação, precisamos fazer alteração dentro do nosso arquivo angular.json.



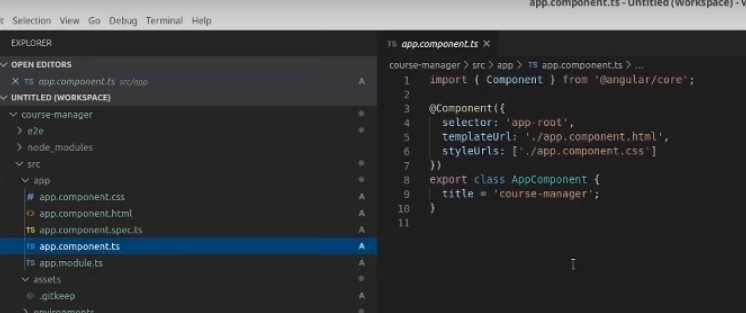
É a partir desse arquivo que vai fazer a leitura do componente pai. O componente pai seria o componente responsável por envelopar todos os componentes que a gente for criar de maneira direta ou indireta, ou seja, esse componente vai ser lido e carregado do index.html.



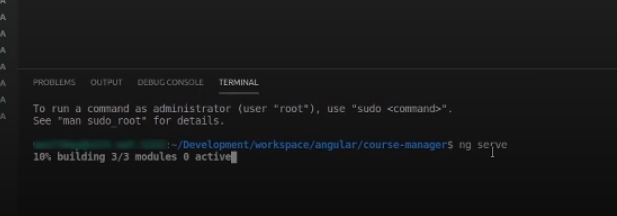
Na nossa aplicação toda teremos apenas um index.html, ou seja, html raiz. Os outros HTML que forem criados ao longo dos nossos componentes, eles serão criados dentro do nosso HTML raiz através do



O app-root faz referência ao nosso componente pai, ou seja, nosso app.component.ts, que esta sendo lido dentro do app.module.ts



Digitar o comando ‘ng serve’ para inicializar o angular no VS Code



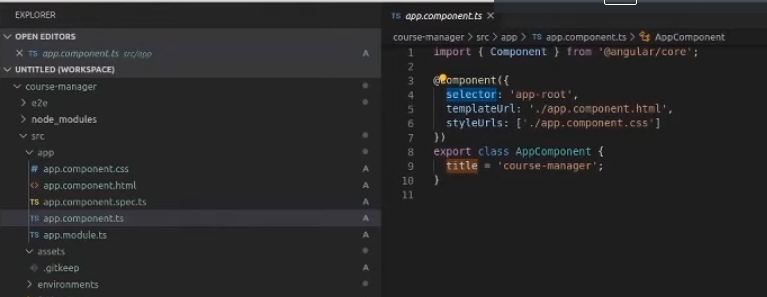
Ele vai adicionar todos os componentes no main.ts

### Trabalhando com módulos e componentes

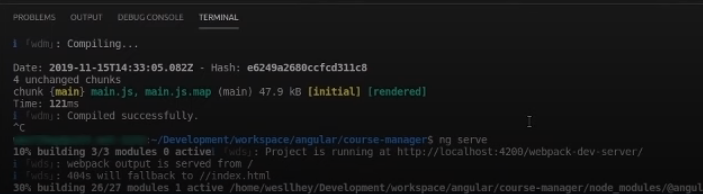
O angular é modular.

O selector serve para criar uma tag chamada “app-root”, que se encontrará no nosso index.html.

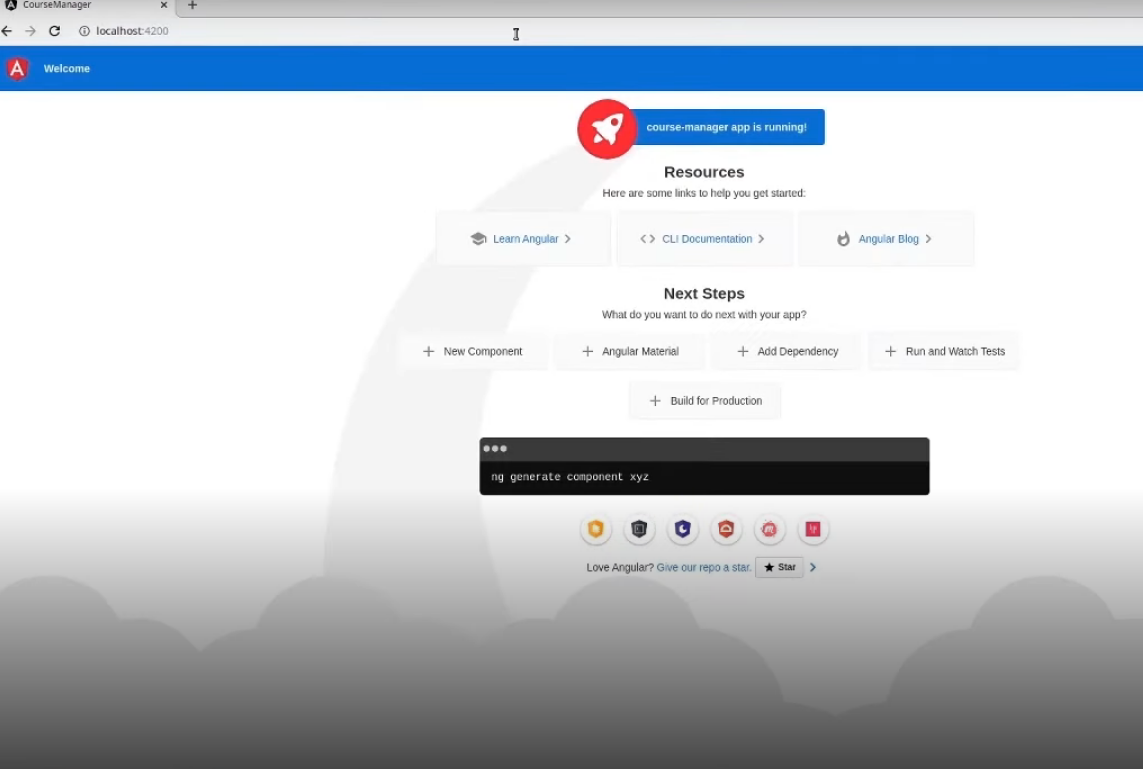
Ou seja, o selector irá criar uma tag que poderá ser usado em qualquer lugar que for necessário.



O componente title vai estar refletindo no app.component.html e poderá ser visualizado através da inicialização do “ng serve”, digitado no terminal.

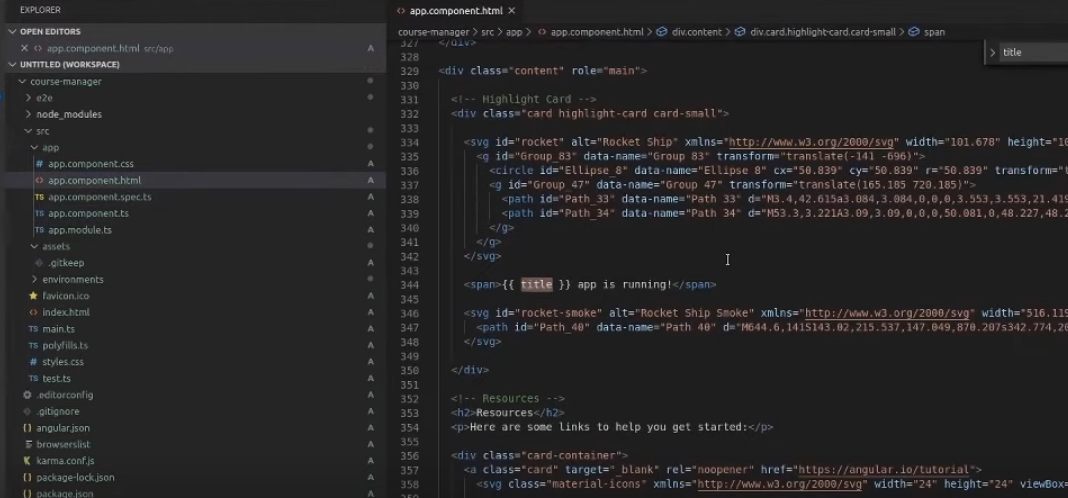


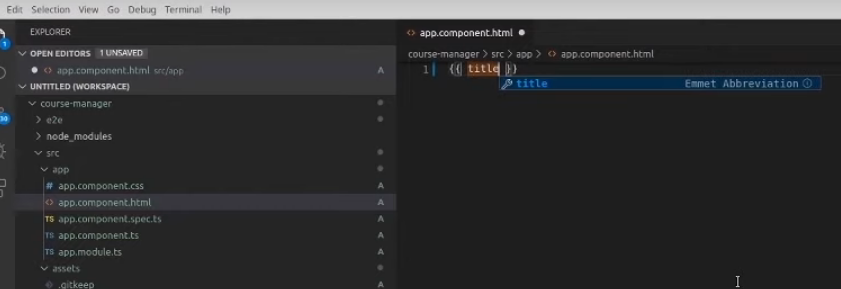
O localhost:4200 é onde conseguimos visualizar como está ficando nossa aplicação.



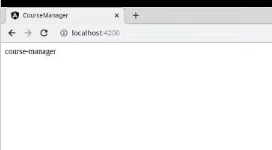
O title fica conforme está indicado na página.

E esse title vai aparecer aqui na página do template. Ele está entre “{{}}” isso significa que é uma interpolação

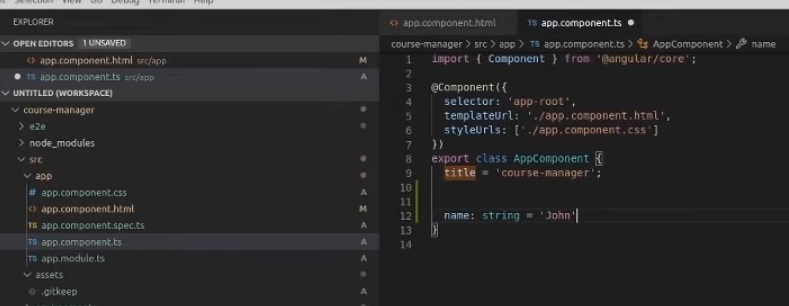




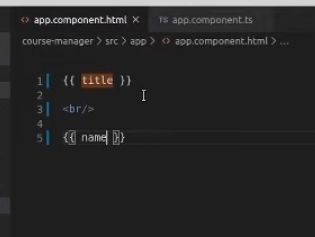
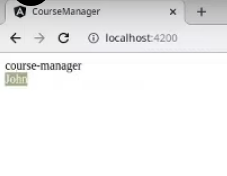
Se apagar tudo que existe dentro do template e deixarmos apenas o “tittle” dentro desse arquivo, ficará dessa forma na página.



Se for criado um novo componente no arquivo app.components.ts, ele também poderá ser interpolado no template.

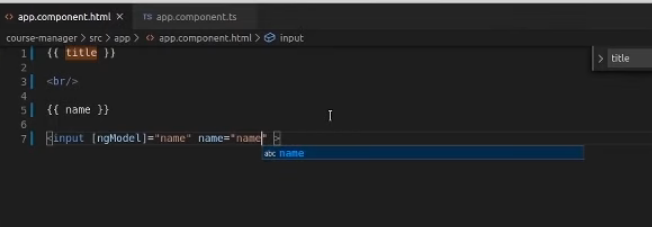


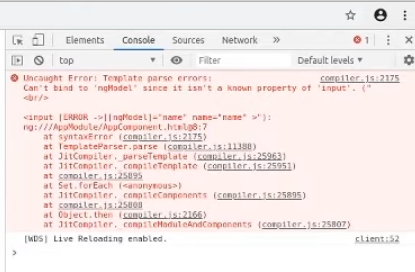
No template se adicionarmos essa interpolação chamada “name” ela irá aparecer no HTML.

O angular visa fortificar o HTML ou seja, ele te dá todo o recurso do HTML mais algumas características peculiares, por exemplo a interpolação. Ele te dá um pouco mais de poder ao trabalhar com o HTML.

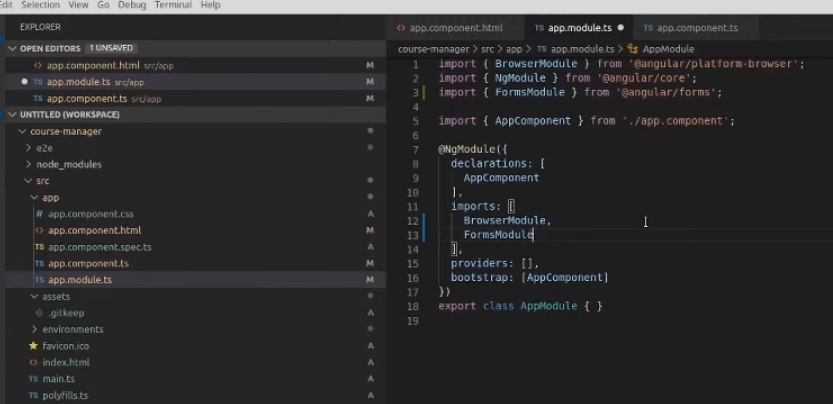
Dentro do template, podemos colocar também o input de texto.



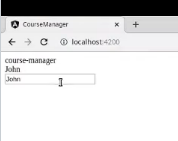


Ele não vai funcionar logo de cara porque esta com erro. Um erro bem comum para quem esta iniciando com esse tipo de recurso.

Ele deu um erro de que não possui ngModel, que é um modelo de formulários e ele não foi aplicado em nenhum lugar no código. Então precisaremos adiciona-lo dentro do arquivo app.module.ts. Importando e usando dessa forma, deixará nossa pagina mais leve, pois estaremos importando apenas aquilo que iremos utilizar.

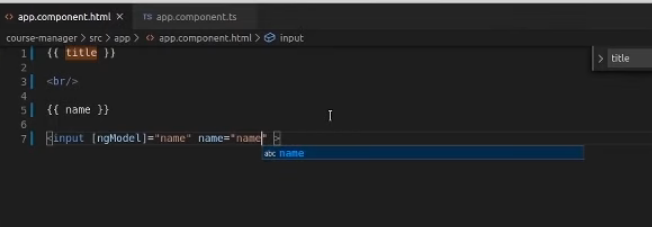


Fazendo somente essa alteração, salvando e voltando para a página, aparecerá nossa caixinha do input.

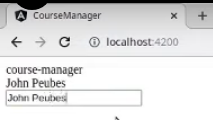


Quando colocamos o ngModel dentro do template apenas entre [], ele será apenas um valor exibido, não podendo ser atualizado ou alterado.

One way data binding – somente exibe

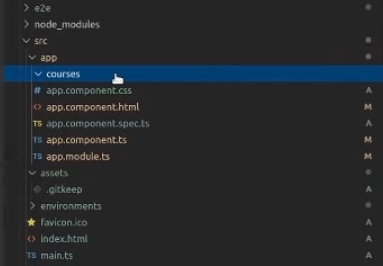
 

Two ways data binding – exibe e atualiza / altera simultaneamente, onde esse valor estiver sendo exibido na nossa página

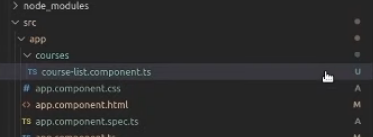
 

### Criando componentes e construindo o projeto

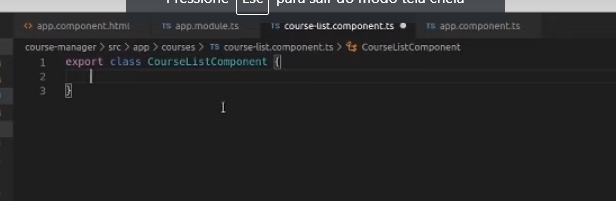
Dentro da pasta de ‘app’, criar a pasta chamada ‘courses’.



Criar um arquivo dentro da pasta ‘courses’ chamado de ‘course-list.component.ts’. Os componentes sempre terão o nomedoarquivo.component.ts



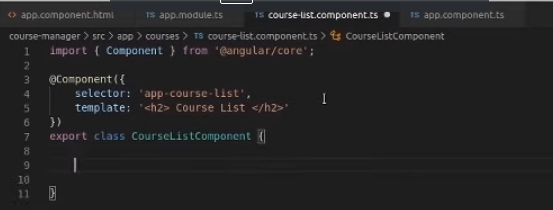
Sempre utilizar camelcase nas nossas classes, ou seja, letras iniciais sempre maiúsculas.



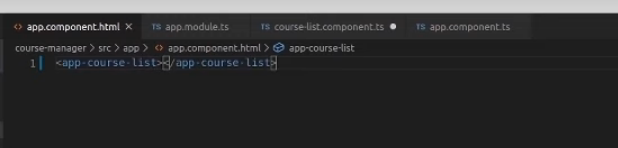
Sempre precisa colocar o import para funcionar os dados na página.



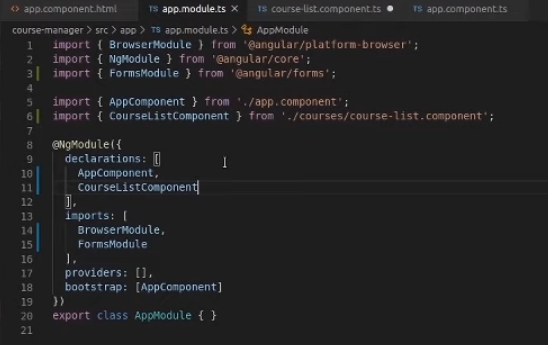
Precisaremos acrescentar o @Component e indicar nossos itens.



Precisa adicionar essa tag no nosso template.

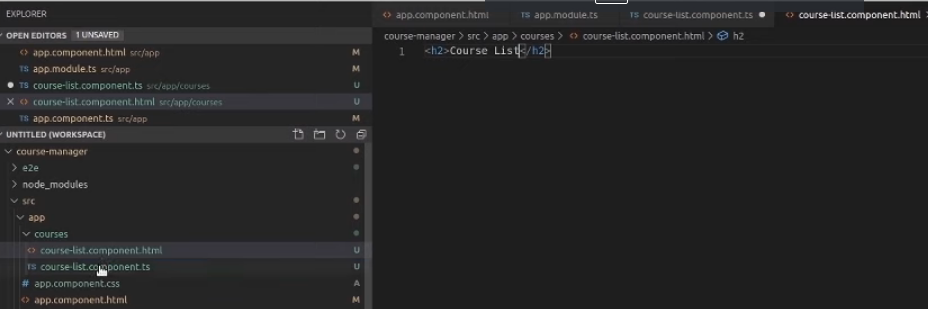


No arquivo app.module.ts precisa declarar o componente courselistcomponent no declaration desse arquivo, caso contrário, ocorrerá um erro e não irá rodar a página.

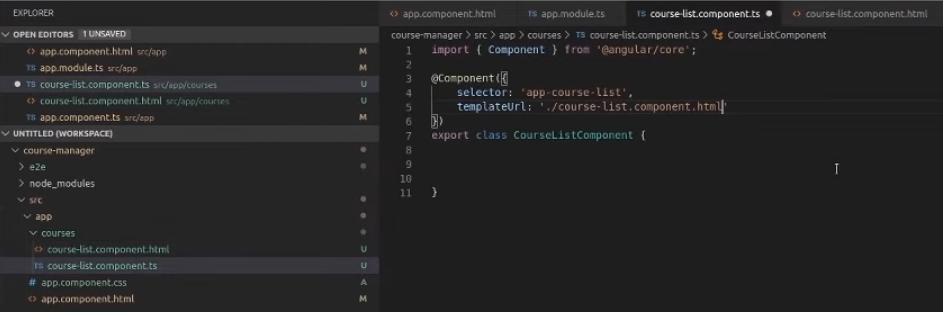


**Toda vez que criar um componente precisa declarar ele no modo que corresponde a ele.**

Como boa prática não é interessante colocar dentro do nosso componente os itens de HTML, é recomendável que se crie um novo arquivo template, onde ficaram armazenados esses dados HTML.



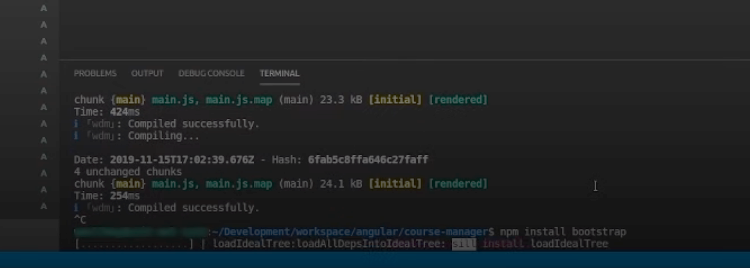
E no nosso arquivo componente iremos alterar de template, para templateUrl e colocar o caminho do nosso arquivo template HTML que criamos.



Para trazer melhor estética para nossa aplicação, iremos instalar o boostrap, utilizando o seguinte código no terminal

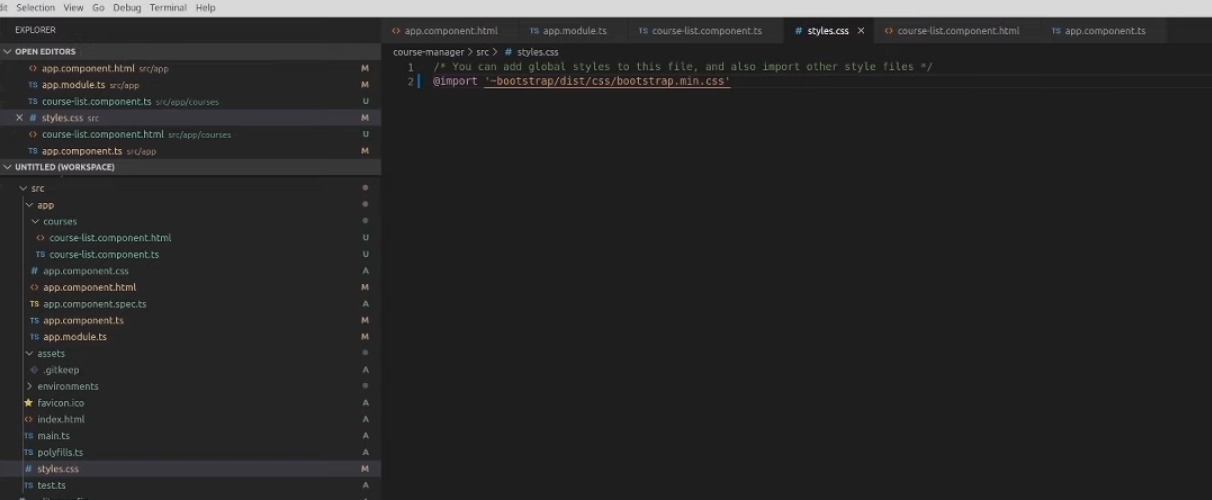
npm install bootstrap

Ele vai fazer o download dos arquivos do bootstrap.



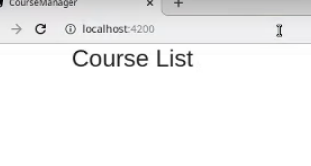
E para ter certeza de que foi instalado com sucesso você vai na pasta node.modules e procurar a pasta com o nome de bootstrap.

E para instalar na nossa aplicação, temos que abrir o style.css (ou o arquivo que está indicado no angular.json na parte de CSS) e importar os dados do bootstrap para eles.



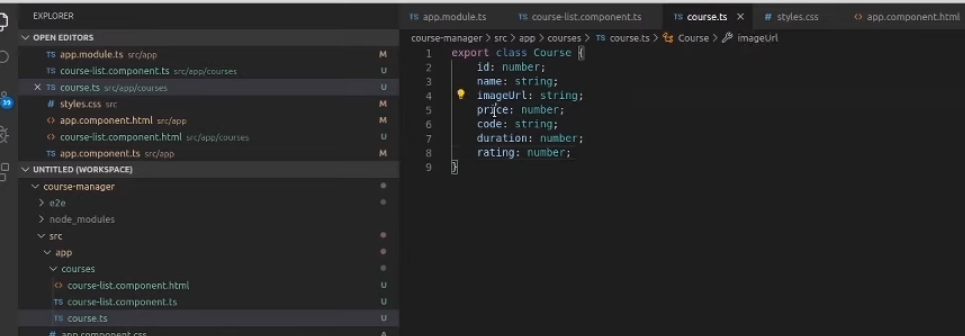
E para verificar se esta funcionando corretamente podemos fazer o teste adicionando um container no template.

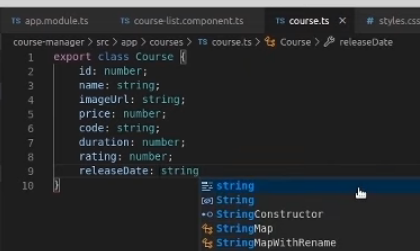




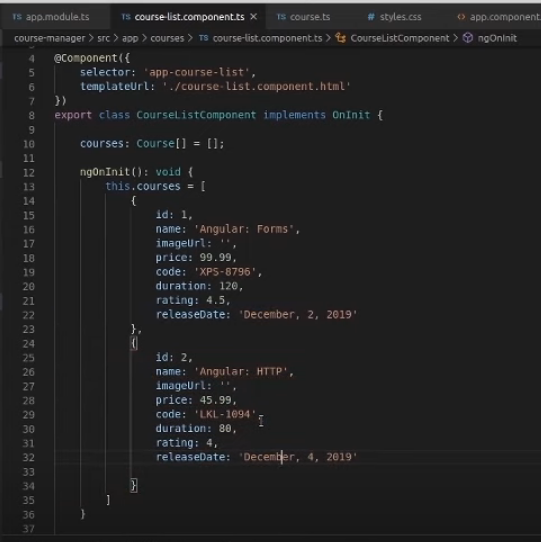
E se rodar a pagina sera possível observar que já esta com a formatação do bootstrap.

Agora iremos criar um novo arquivo dentro da pasta ‘courses’, chamado courses.ts onde ficará a classe e a identificação dos itens dessa nova classe.

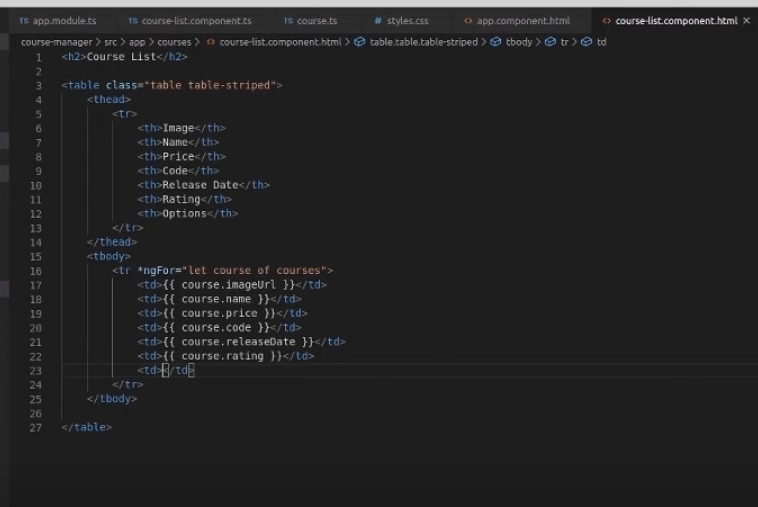


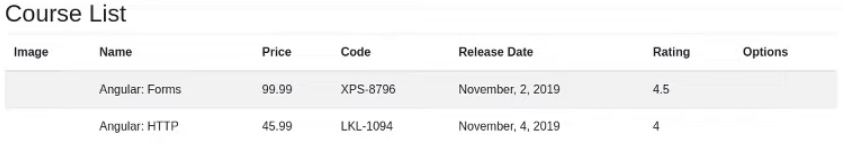


Dentro do arquivo course-list.component.ts devemos exportar a classe que criamos no arquivo courses.ts.



Agora iremos exibir esses dois cursos no nosso HTML.





### Criando componentes com atributos interdependentes

### Link para o projeto no repositório do GitHub

## Lidando com vários componentes

### O que é injeção de independência

### Tratando eventos data binding

### Transformando datas com Pipe

### Protegendo rotas com guards

### Ativando rotas para acessar componentes

### Trabalhando com formulários e templates

### Link para o projeto no repositório do GitHub

## Segregando responsabilidades

### Fazendo requisições em HTTP

### Trabalhando com delete

### Segregando a aplicação em módulos

### Conhecendo a pasta shared

### Conhecendo a pasta core

### Tratando erros com angular

### Recomendações do especialista

### Link para o projeto no repositório do GitHub