CRUD ANGULAR

fatec – senai centro | Rua Caetés 741, Belo Horizonte

CRUD COM ANGULAR 9

Rubem Cândido dos Santos

2020

Sumário

[Introdução 2](#_Toc39569025)

[Definição 2](#_Toc39569026)

[O que o Java Script faz? 2](#_Toc39569027)

[Client x Server 2](#_Toc39569028)

[HTML x CSS x JavaScript, “a receita do bolo” 3](#_Toc39569029)

[02 – Configuração do Ambiente 3](#_Toc39569030)

[Instalando o Node 3](#_Toc39569031)

[Instalando Visual Studio Code 10](#_Toc39569032)

[Plugin Code Runner 13](#_Toc39569033)

[Plugins Material Icon Theme 14](#_Toc39569034)

[Teste 14](#_Toc39569035)

[Outra forma de criar repositório e abrindo no VS Code 17](#_Toc39569036)

[Criando repositório via Prompt 18](#_Toc39569037)

[Instalando JSON Server 27](#_Toc39569038)

[Criando arquivos de dependência 28](#_Toc39569039)

[Instalando a dependência do JSON Server 29](#_Toc39569040)

[Github 30](#_Toc39569041)

[Conclusão 30](#_Toc39569042)

[BIBLIOGRAFIA BÁSICA: 31](#_Toc39569043)

[BIBLIOGRAFIA COMPLEMENTAR: 31](#_Toc39569044)

# Introdução

## Definição

Angular é um framework JavaScript desenvolvido pela Google para criação de aplicações Web SPA (Single Page Aplication) baseada em **componentes**.

AngularJs foi a primeira versão do angular

Angular 2 – Baseada em componentes

Angular 4, 5, 6, 7, 8, 9, ...

CLI – Command Line Interface

npm i –g @angular/cli

G é de global

ng new minha-app

ng são as duas primeiras consoantes de angular

TypeScript, TS é uma linguagem Orientada a Objetos criada pela Microsoft. O código escrito em TypeScript é compilado para JavaScript. Um TypeScript é um superSet do JavaScript. O TypeScript tem tudo com que o JavaScript tem e acrescenta algumas funcionalidades como a **tipagem forte**, decorator, interfaces**.**

JavaScript é uma linguagem de programação que permite a você criar conteúdo que se atualiza dinamicamente, controlar múltimídias, imagens animadas.

ECMAScript é o nome oficial da linguagem JavaScript.

## O que o Java Script faz?

Toda vez que uma página da web faz mais do que simplesmente mostrar a você informação estática — mostrando conteúdo que se atualiza em um intervalo de tempo, mapas interativos ou gráficos 2D/3D animados, etc. Podemos esperar encontrar códigos escritos em JavaScript.

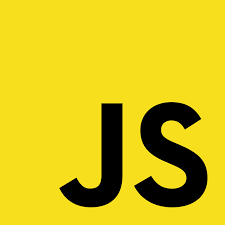


Figura1. Logo da JavaScript

# Client x Server

As máquinas que acessam, consomem serviços, são considerados clientes. Os equipamentos, dispositivos que disponibilizam serviços, são os servidores.

Exemplo:

Quando acesso o Google Class Room estou como cliente.

Os materiais, slides, atividades estão disponibilizados em um servidor.

# HTML x CSS x JavaScript, “a receita do bolo”

Na Web, o JavaScript serve como uma parte de interação, porém, ele trabalha com outras tecnologias de construção Web muito utilizadas.

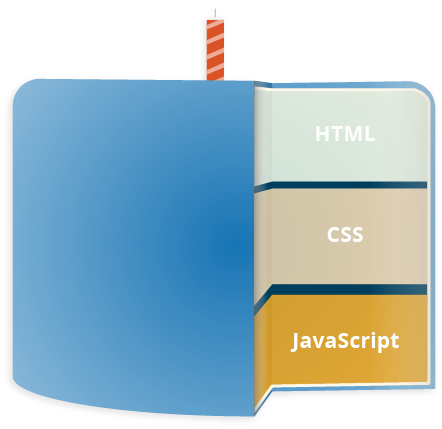


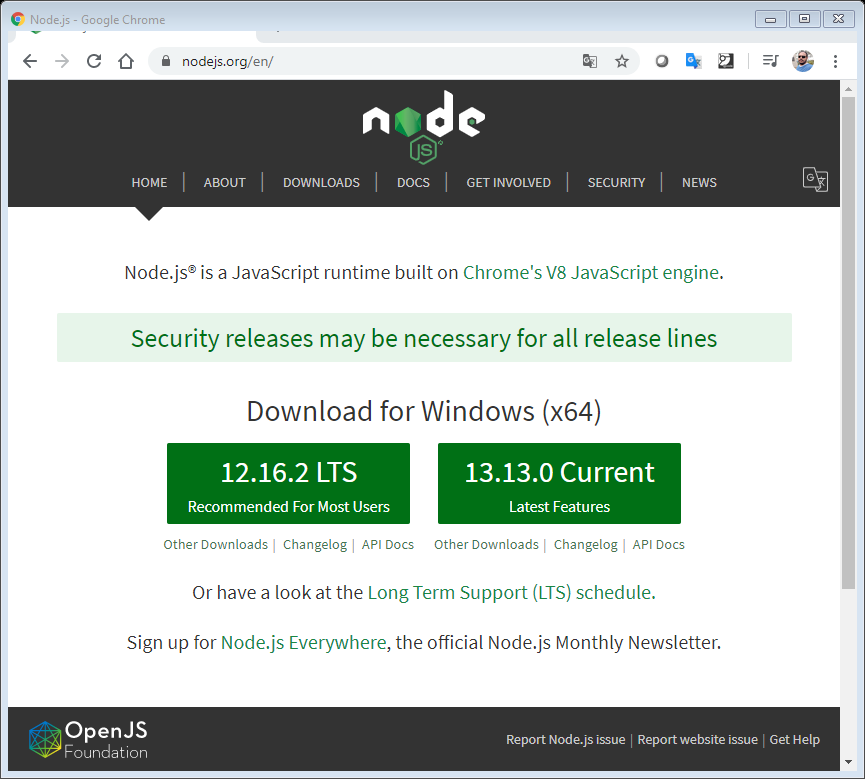
Figura 2. O conjunto de duas tecnologias e uma linguagem de programação

01 – Visão Geral

# 02 – Configuração do Ambiente

## Instalando o Node

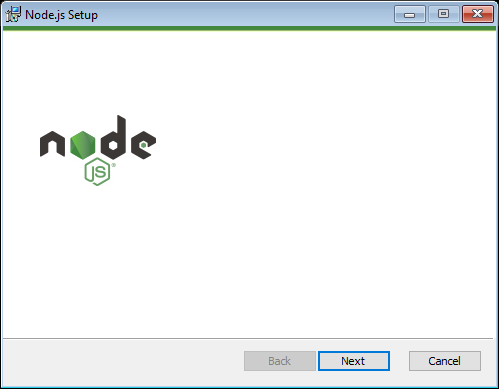
Acesse o site nodejs.org e baixar a versão LTS, que é a versão que possui um suporte mais longo.



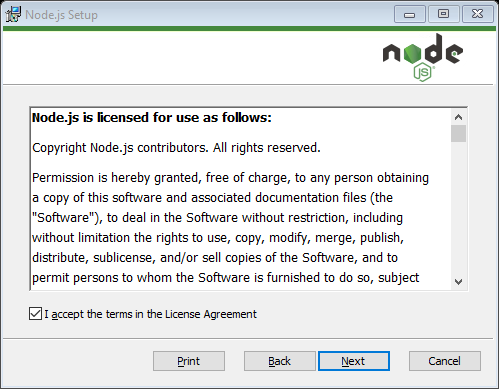
Duplo clique no arquivo



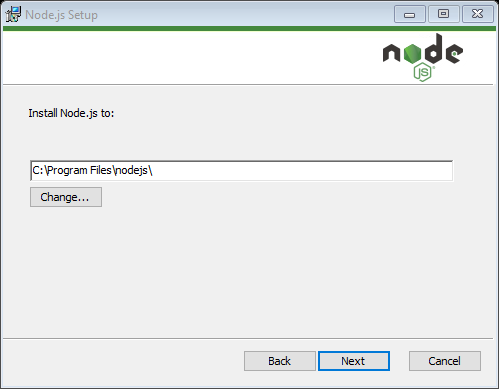
Siga a instalação



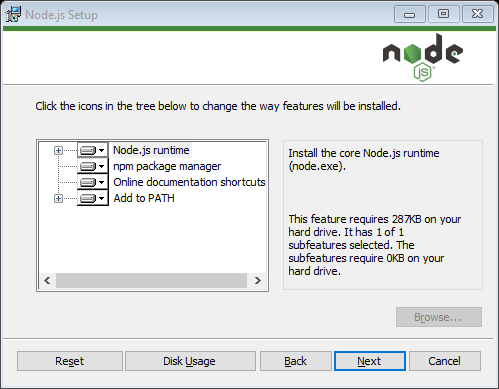
Aceite os termos e clique em Next



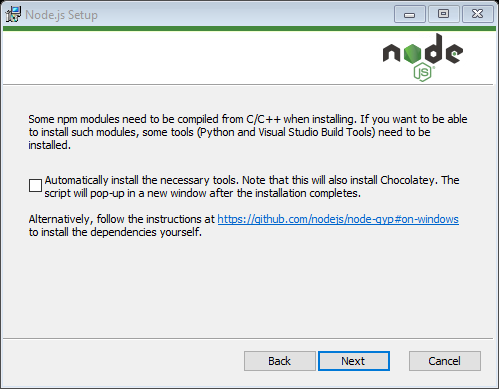
Selecione o local de instalação e clique em Next



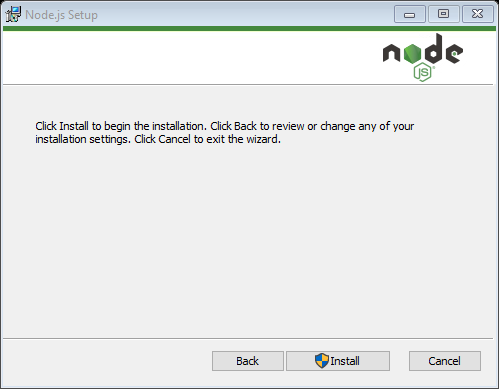
Deixe todos os itens selecionados e clique em Next



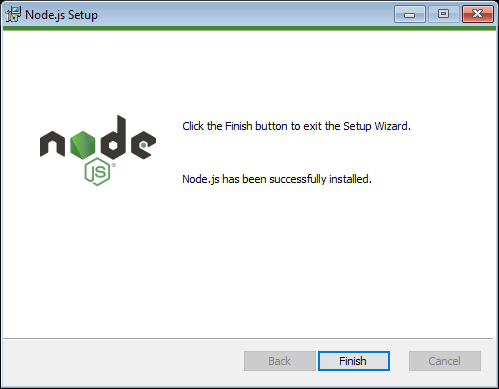
Next



Clique em Install



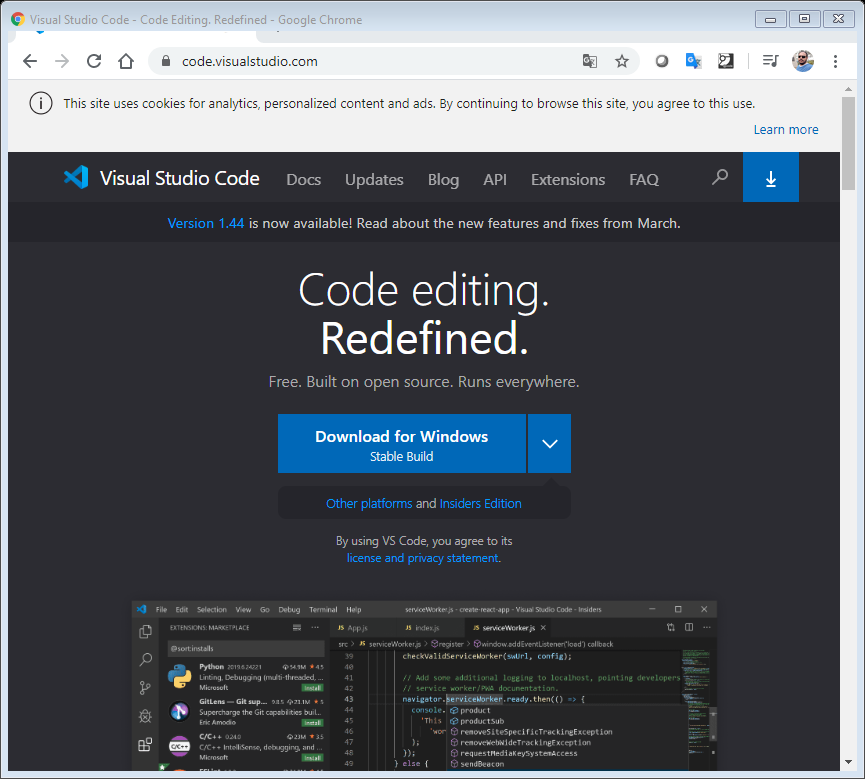
Clique em Finish



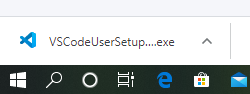
## Instalando Visual Studio Code

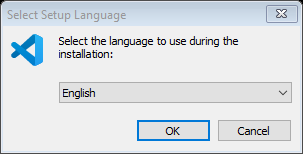
Vamos instalar o editor de texto Visual Studio Code no link

<https://code.visualstudio.com/>

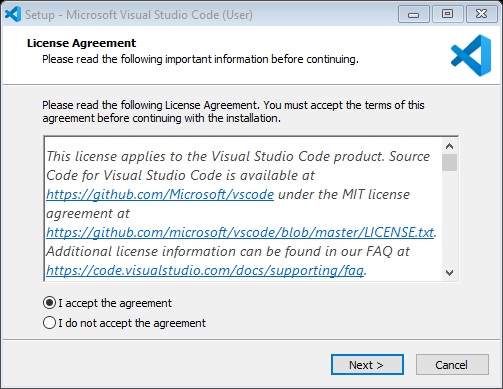


Instale seguindo o wizard de instalação

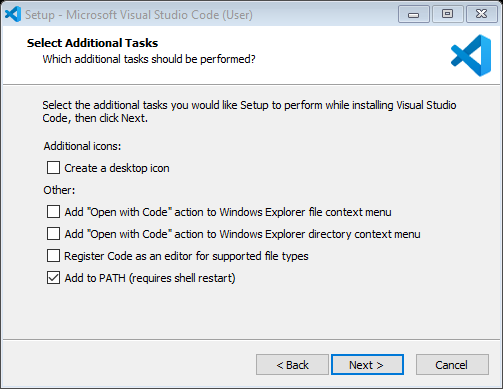




Aceite os termos



Marque a opção Add to PATH (requires shel restart)



Clique em Finish



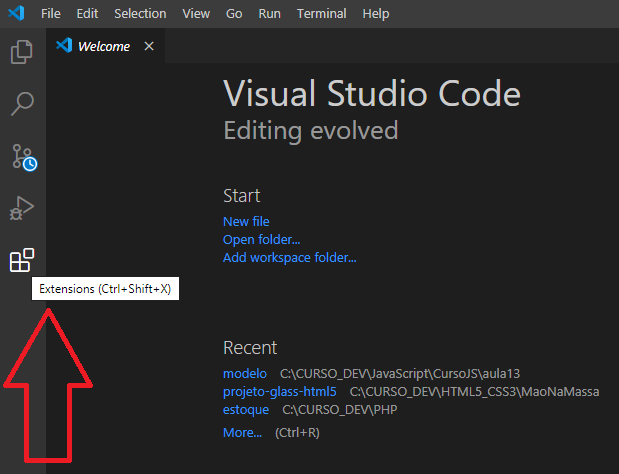
Reinicie o sistema operacional

Instalando o Shell Command

Com o Visual Studio aberto, pressione as teclas Ctrl + Shift + P e digite code

## Plugin Code Runner

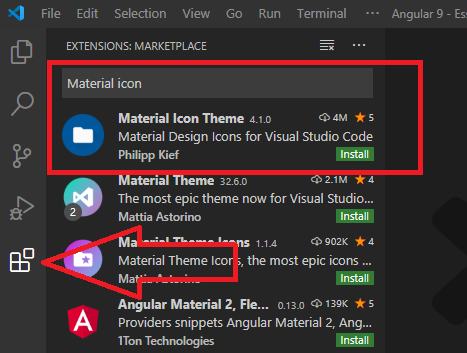
Após a instalação do Visual Studio Code, instale o plugin Code Runner. Clique no ícone de extensões, para facilitar a execução do JavaScript dentro do Visual Studio Code.



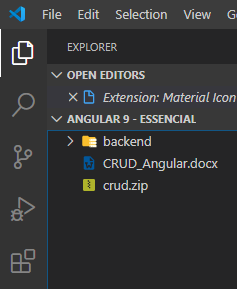
Digite runner e em Code Runner e clique em Install



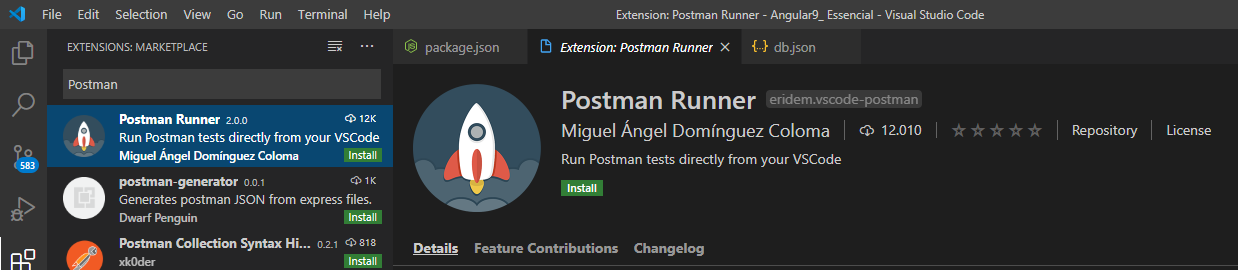
## Plugins Material Icon Theme

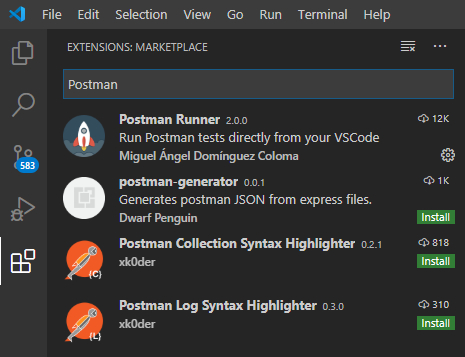


Essa extensão cria ícones no Visual Studio



Postman – Extensão



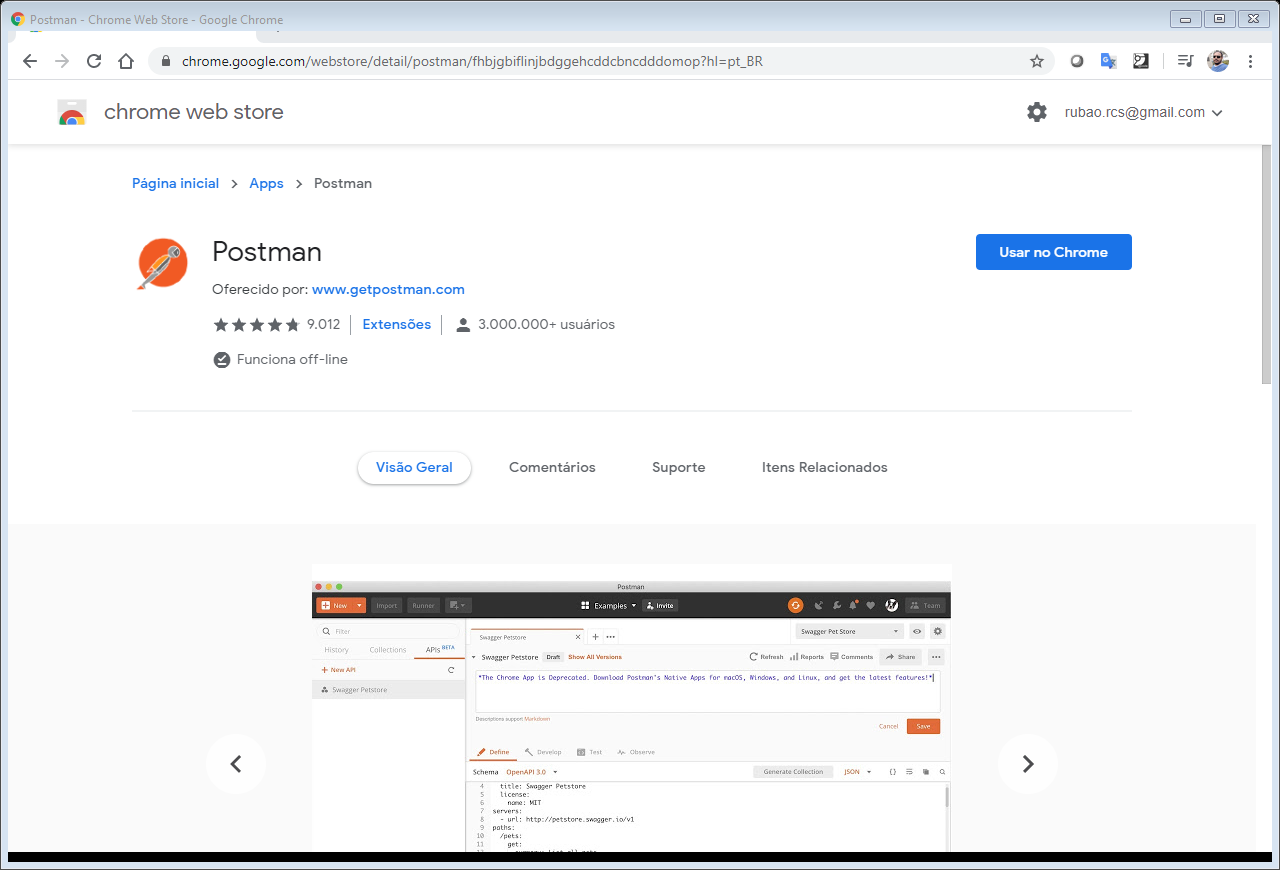


Postman extensão para o Chrome

Faça uma busca no google por Postman e instale a extensão

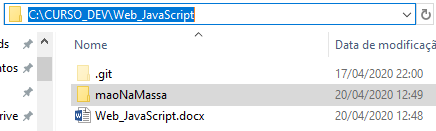
<https://chrome.google.com/webstore/detail/postman/fhbjgbiflinjbdggehcddcbncdddomop?hl=pt_BR>

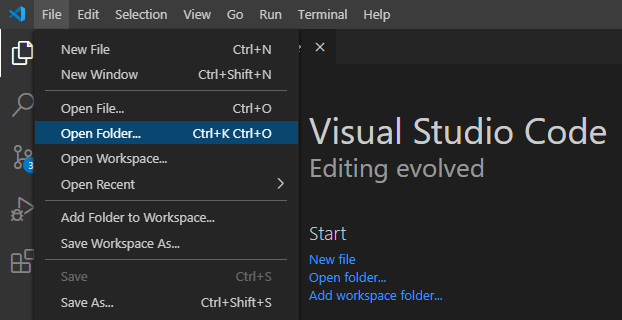
Usar no Chrome



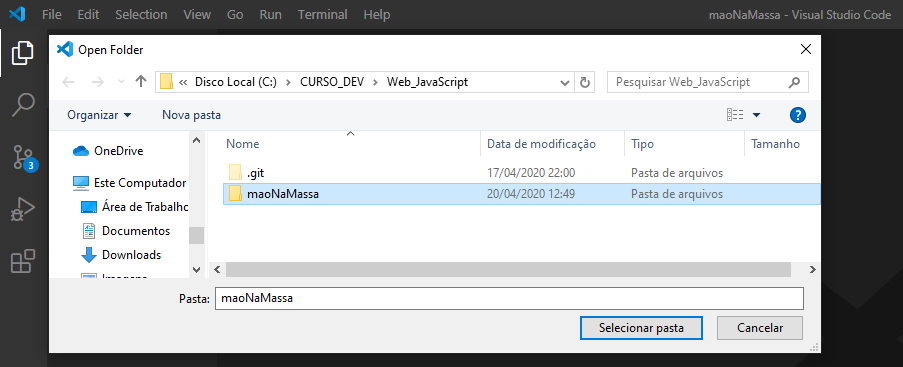
# Teste

Crie um diretório e abra o Visual Studio clicando em Open Folder

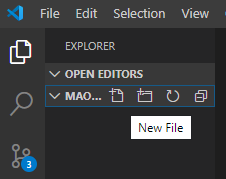




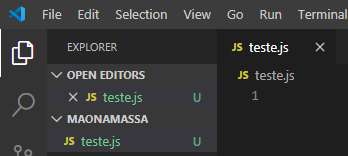
Selecione o diretório criado



Crie um novo arquivo



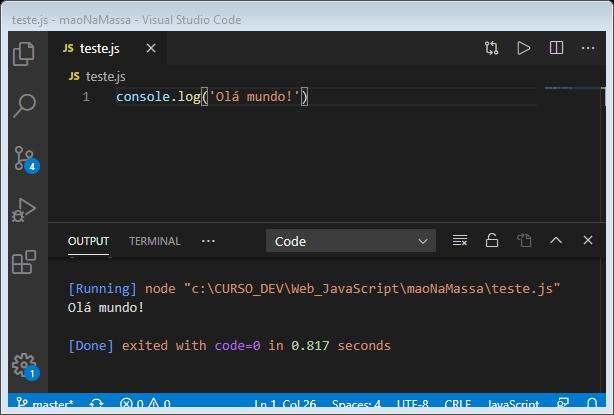
Crie um arquivo com a extensão .js



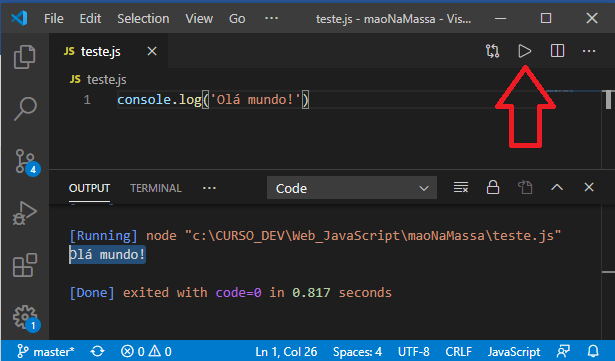
Digite o código

console.log('Olá mundo!')

Pressione o atalho Ctrl + Alt + N ou no sinal de player



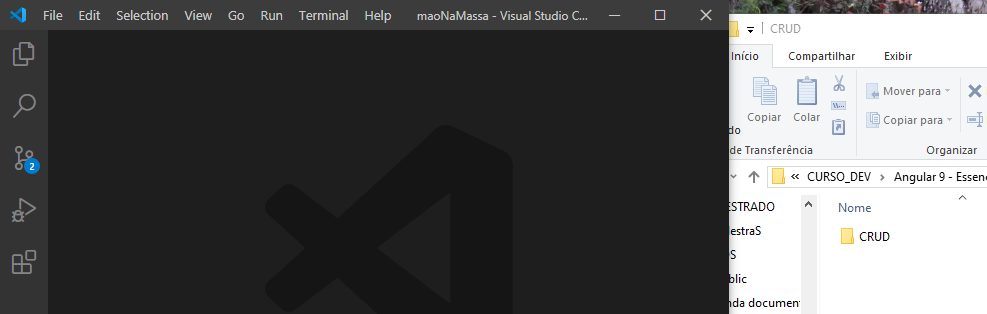
console.log('Olá mundo!')

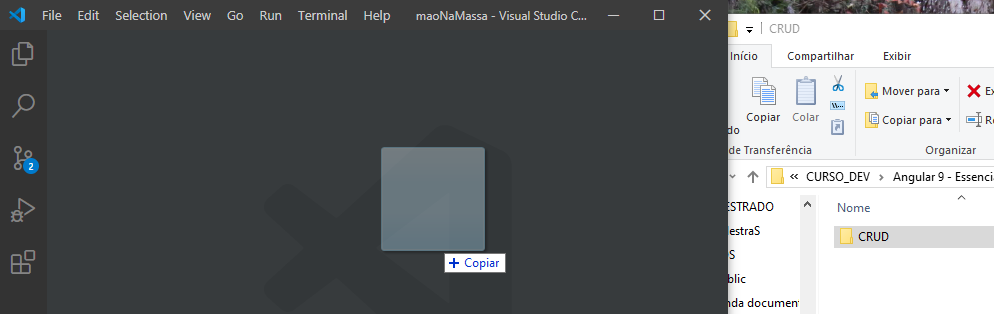


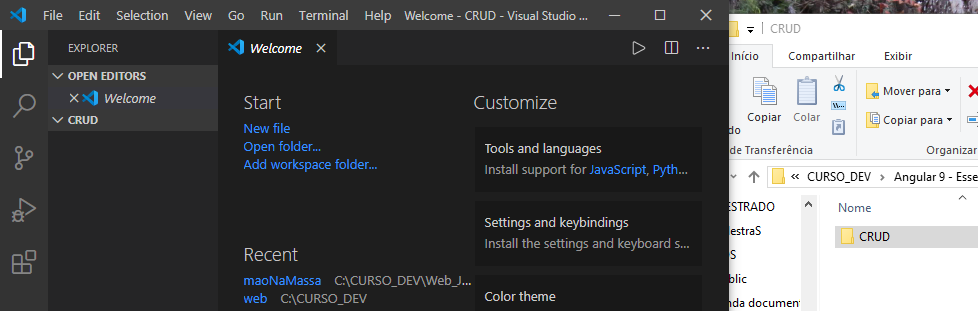
# Outra forma de criar repositório e abrindo no VS Code

Crie um diretório e abra-o no Visual Studio Code

Uma opção é arrastar o diretório para dentro do visual Studio Code



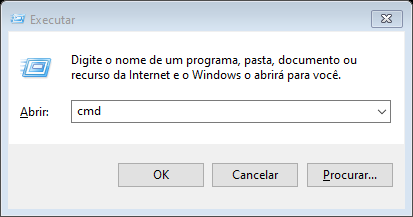




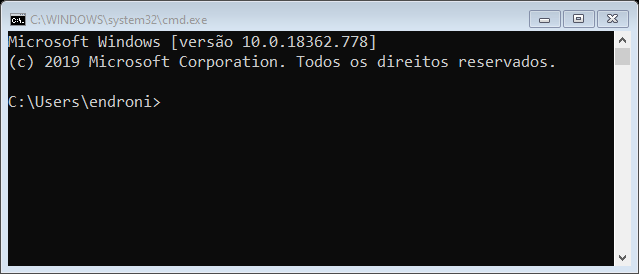
# Criando repositório via Prompt

Podemos criar diretório, arquivo e iniciando o VS Code via comando.

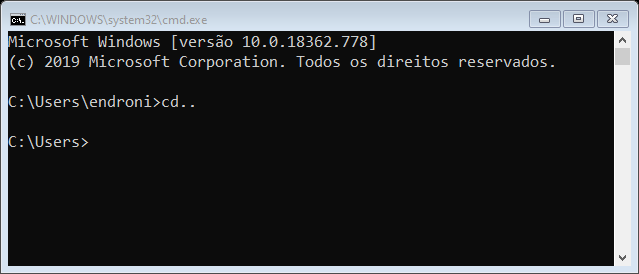
Pressione o atalho Windows + R, digite cmd e pressione a tecla Enter



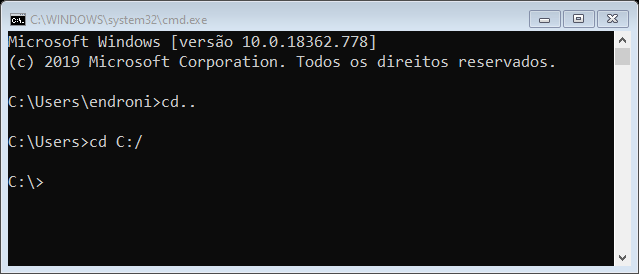
Navegue até o diretório necessário.



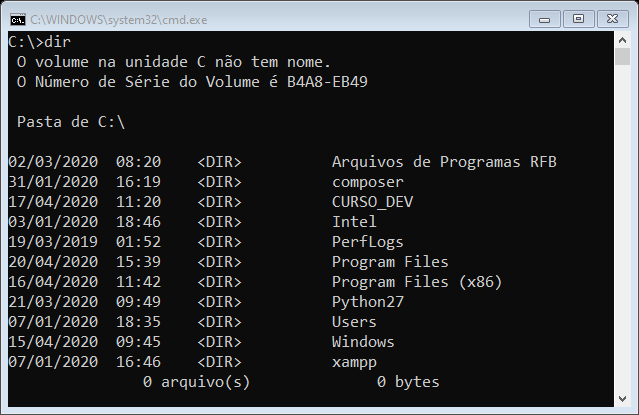
cd.. vai para um nível acima



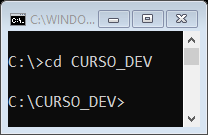
cd caminho navega para o local desejado.



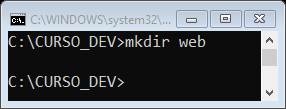
dir – exibe o conteúdo local



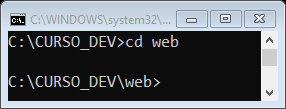
No meu caso eu quero acessar o diretório CURSO\_DEV



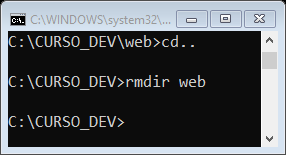
mkdir diretório – cria um diretório com o nome especificado



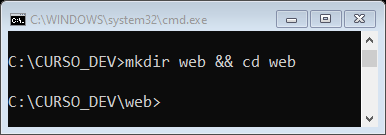
Acessando o diretório



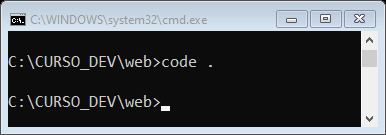
rmdir diretório – exclui o diretório



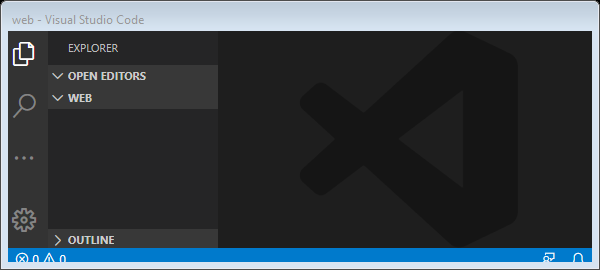
Podemos criar o diretório e acessar concatenando dois comandos com &&



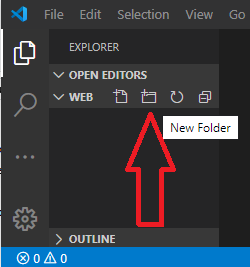
code . - O ponto inicia o VS Code no diretório

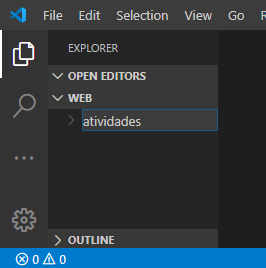


VS Code iniciado no diretório

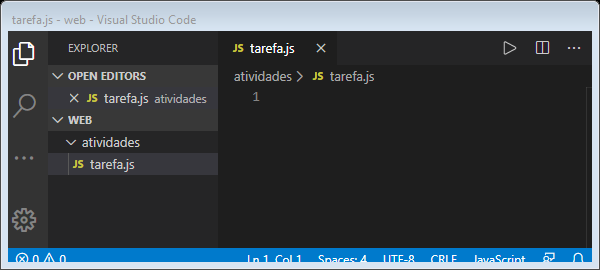


Podemos criar diretórios ou arquivos no VS Code

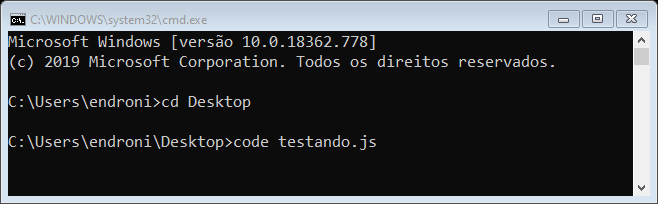




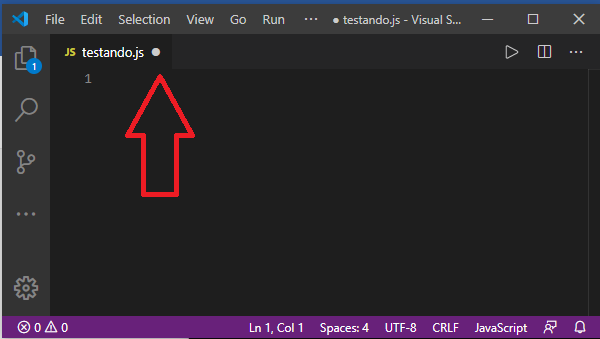
Criando arquivo clicando com o botão direito no diretório e selecionando a opção New File

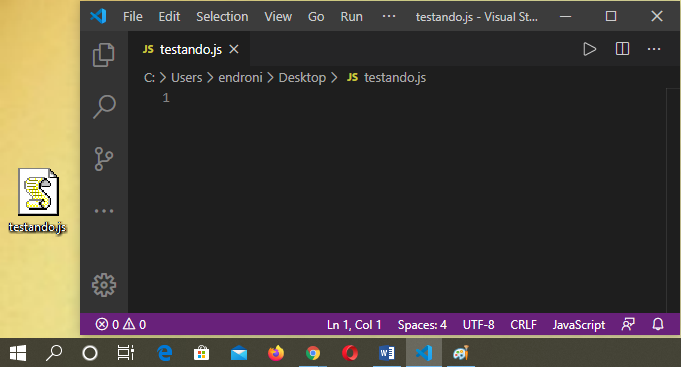


Podemos também criar um arquivo diretamente pelo prompt:

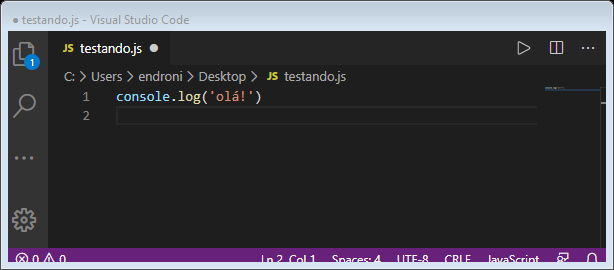


O Visual Studio Code irá ser aberto com um arquivo não existente. A bolinha significa que o arquivo não está salvo.

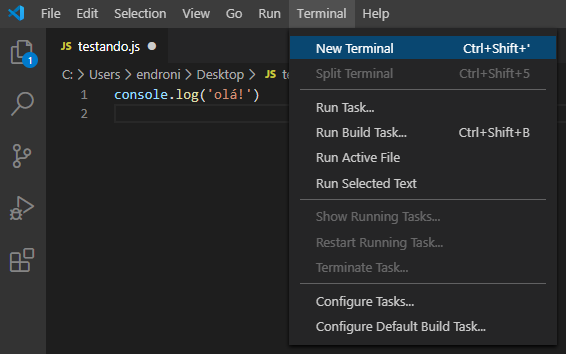
Ao mandar salvar, o arquivo será gerado automaticamente no local onde foi aberto.



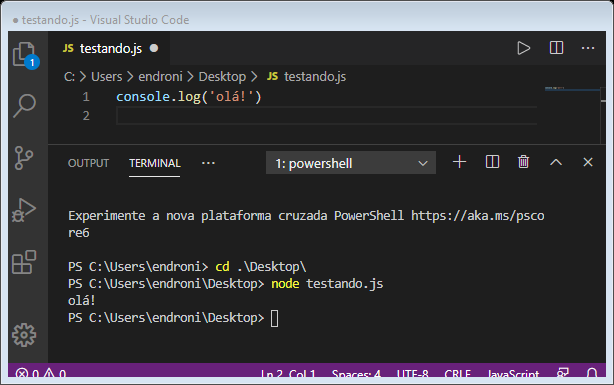
Ao escrever algum comando, podemos executar diretamente no terminal



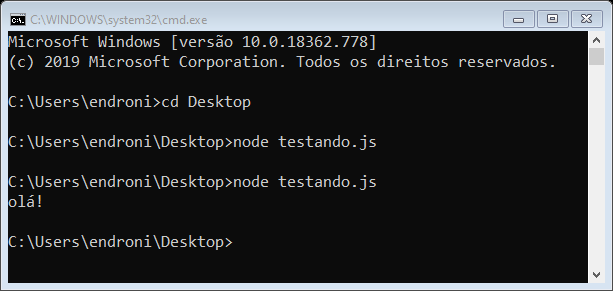
Abrindo o terminal clicando na aba Terminal > New Terminal



Podemos mandar executar o conteúdo de um arquivo no próprio terminal do Visual Studio Code:



E também no prompt de comando do Windows: **node nomeDoArquivo**

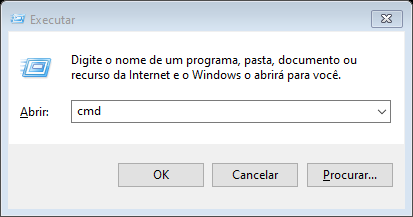


# Instalando JSON Server

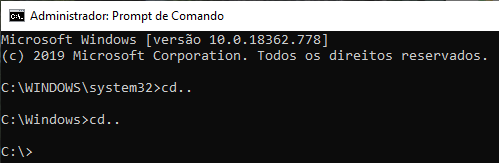
Para a instalação do JSON Server é necessário a existenca do NPM, o NPM é instalado por padrão, junto com o NODE Js. Portanto, já instalamos ele em capítulos anteriores.

NPM é o gerenciador de pacotes do JavaScript

Abra o Prompt de comando

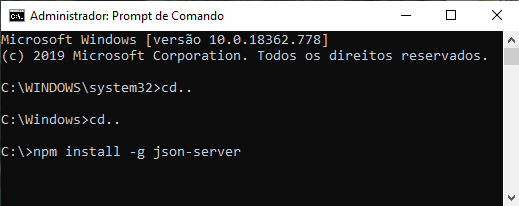


Navegue até a raiz da unidade

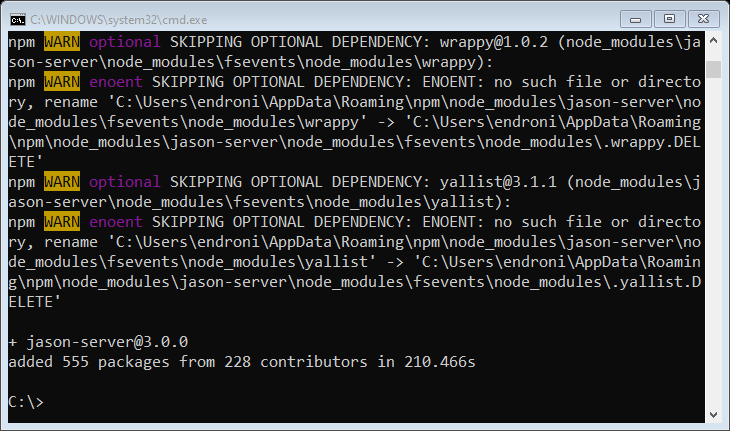


Digite o comando:

npm install –g json-server



Vai levar um tempo



Obs. No meu caso deu um erro

npm ERR! Unexpected end of JSON input while parsing near '...0","strip-ansi":"^0.3'

Só consegui instalar depois de limpar o cach com o comando abaixo

npm cach clean –force

# Criando arquivos de dependência

Crie um diretório chamado backend

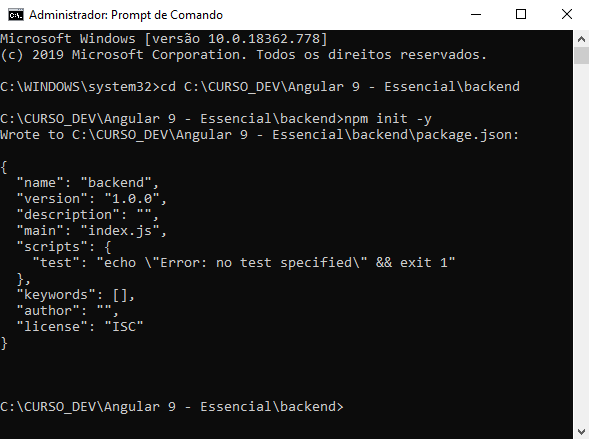
Abra o prompt de comando como administrador

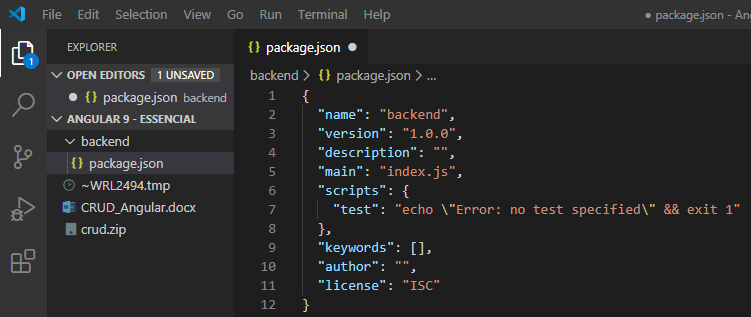
Acesse o diretório backend e inicialize o npm dentro desse diretório.

Digite o comando

npm init –y

-y serve para responder todas as perguntas com respostas padrões.

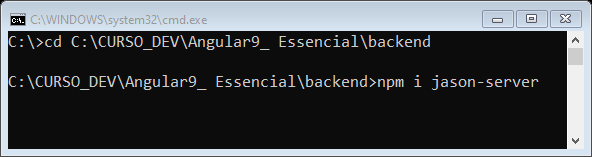


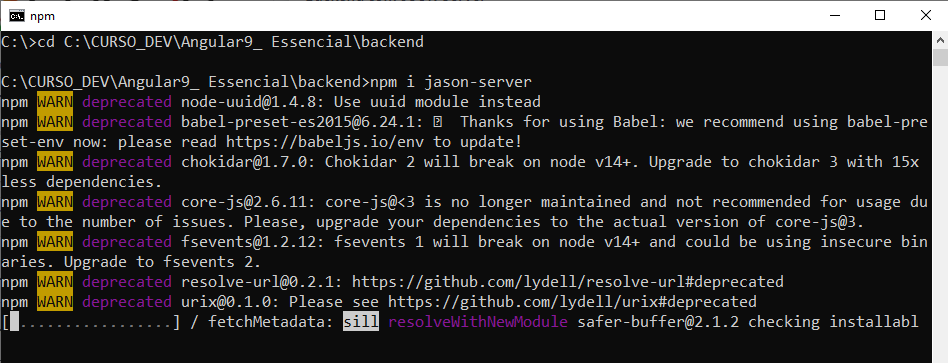


# Instalando a dependência do JSON Server

Digite o comando:

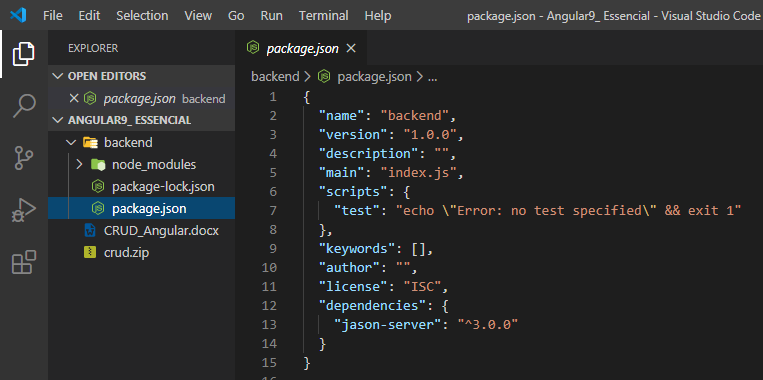
npm i json-server







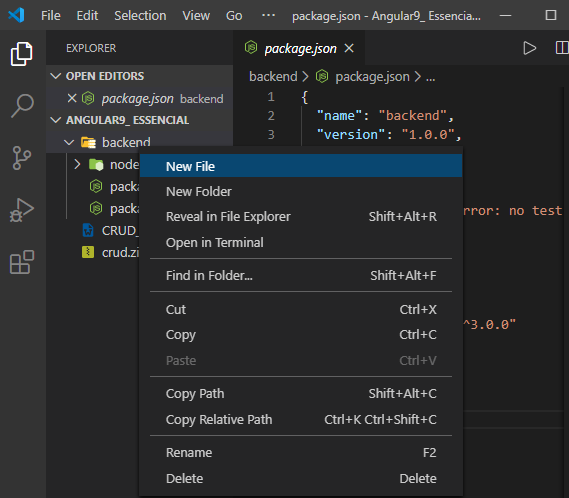
Arquivo de dependência criada:

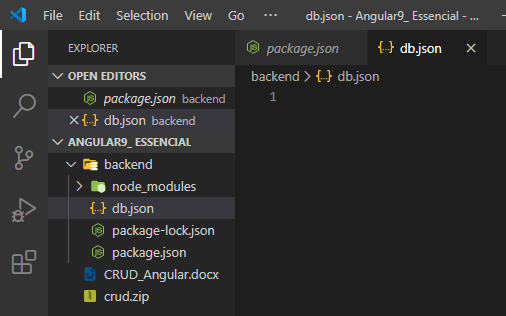


Json lê o arquivo que tem json e cria uma api

# Criar arquivo db.json

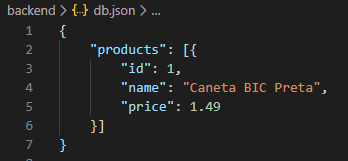
Dentro do diretório backend, crie um arquivo com nome db.jsaon



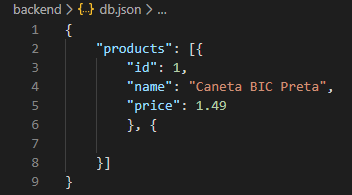


Vamos criar um objeto e dentro desse objeto teremos todos os end points (pontos de saída):

Products com um array



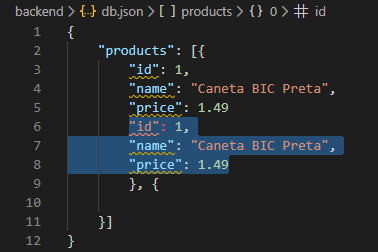
Vamos criar outro produto



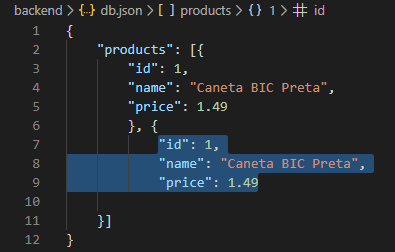
Vamos copiar, selecionando as linhas



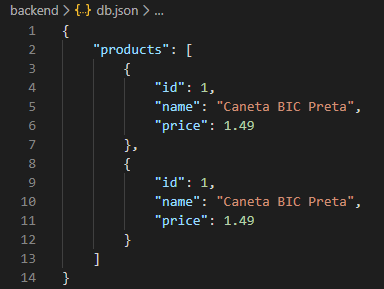
Pressione Shift + Alt + Seta para baixo para copiar e colar logo abaixo as linhas selecionadas:



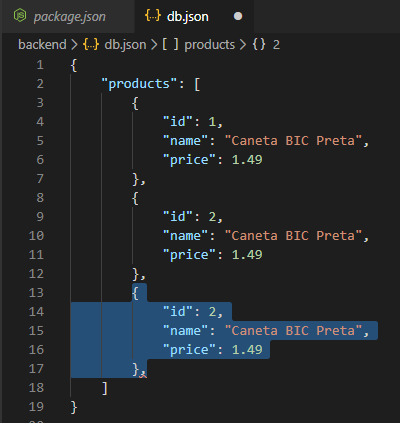
Inserindo as linhas no objeto abaixo com o atalho Alt + Seta para baixo:



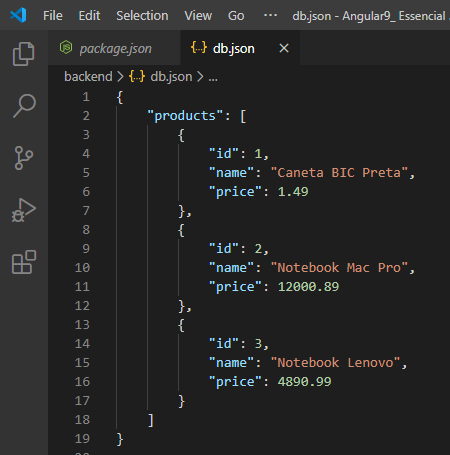
Agora vamos ajustar o endentamento do código com o atalho Shift + Alt + F



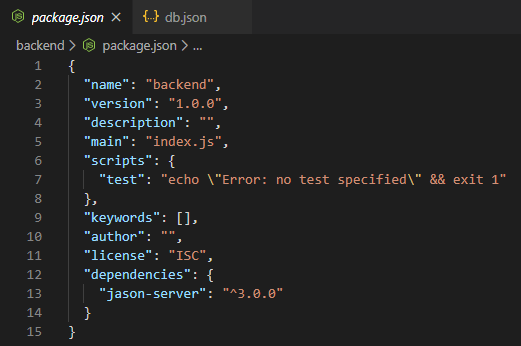
Vamos adicionar outro objeto. Adicionando uma virgula e repetir o comando Alt + Shift + Seta para baixo



Retire a ultima vírgula e vamos ajustar os produtos



Vamos voltar ao arquivo pacage.json



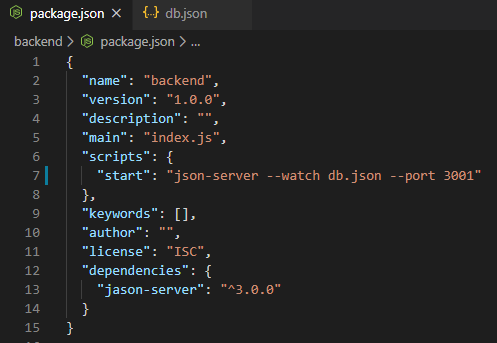
Vamos alterar a linha 7 removendo a linha de test

**“start” –** Inicializando com start

**Json-server –** Chamar o json server

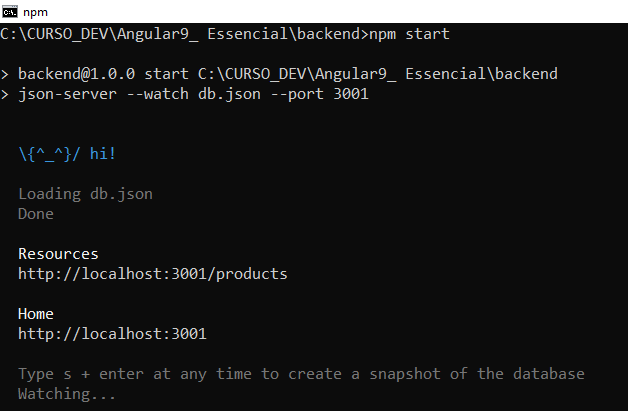
**Watch –** Ficar monitorando o db.json

**port 3001 –** Porta de serviço



Vamos startar com o comando abaixo dentro da pasta backend:

npm start



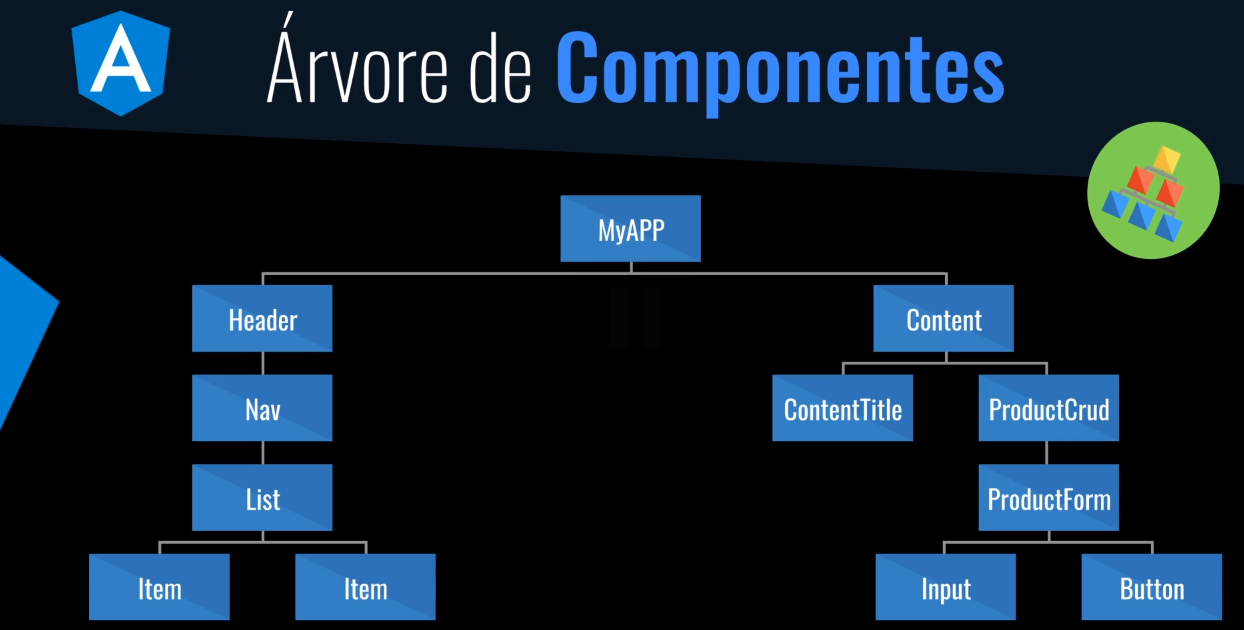
Obs. Tive problemas e para resolver, reinstalei o json-server.

# Realizando testes

Vamos abrir o Postman com o atalho Shift + Ctrl + P e digitar Postman

# FrontEnd

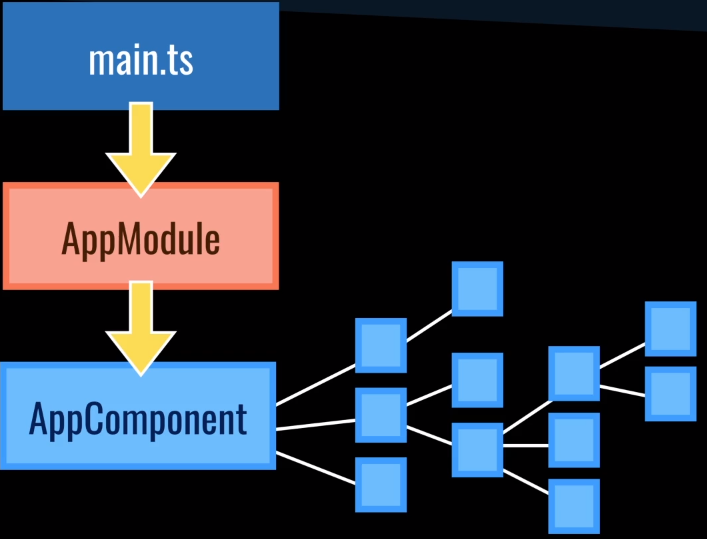
## Árvore de Componentes



# Conceitos essenciais do Angular

# main.ts

Inicianlização da APP

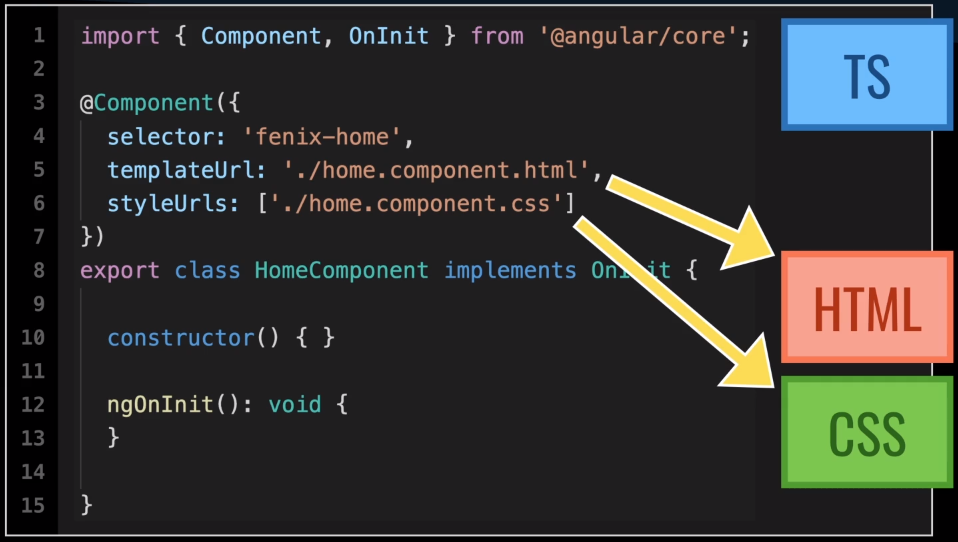


# O que é um Componente

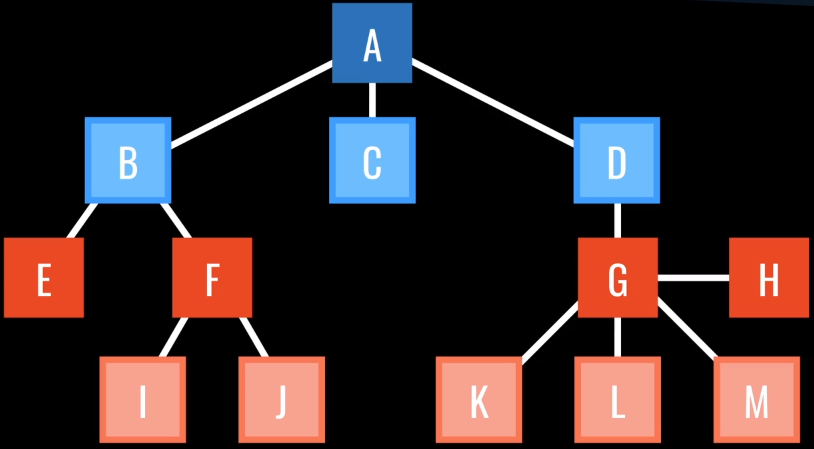
Um pedaço em uma aplicação



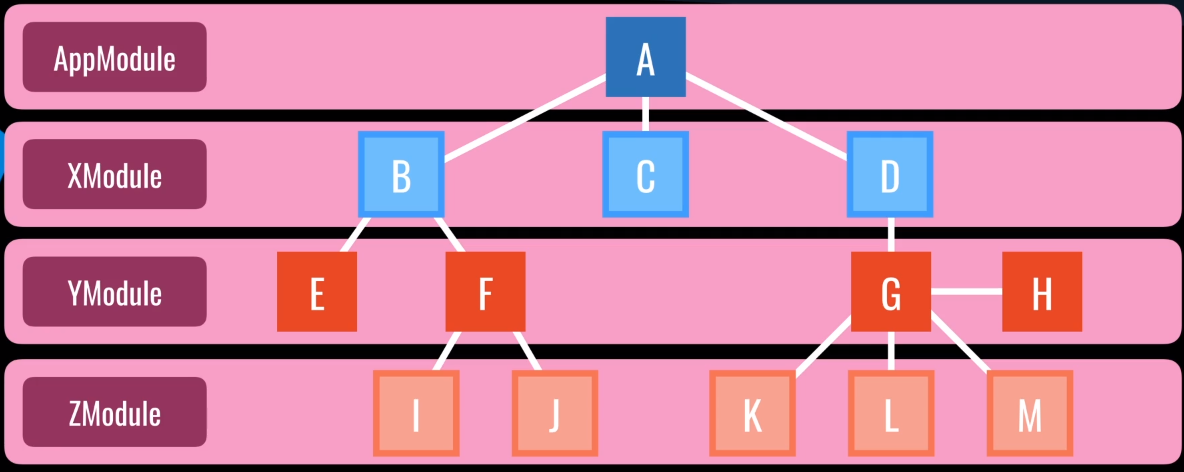
<app-home></app-home>



# Organização Usando Módulo



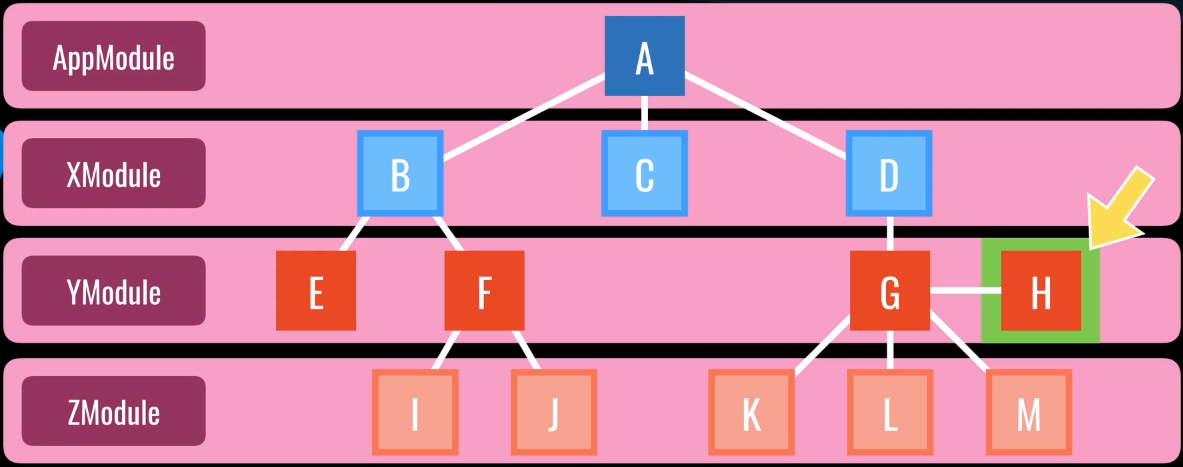
Pode ser dividido em vários módulos



O coponente H é o único componente que só pode ser visualizado dentro do próprio módulo, pois é o único que está relacionado apenas com um outro componente (G) estando no mesmo módulo.

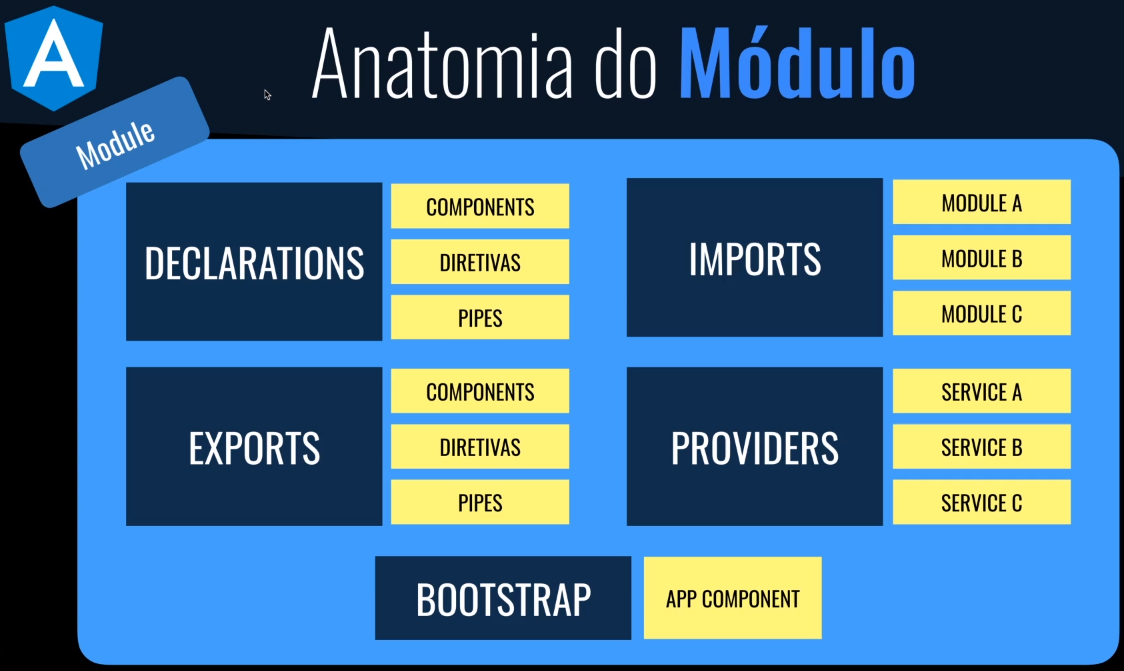
No exemplo, o componente H, poderia representar o módulo do RH ou do FINANCEIRO. Ou seja, possui informações que dizem a respeito apenas àquele módulo.

Todos os outros componentes precisam estar acessíveis para outros módulos.



# Atributos de um módulo

Exemplos de atributos de um módulo



Dentro de um módulo teremos cinco atributos: Declarations, Imports, Exports, Providers e Bootstrap

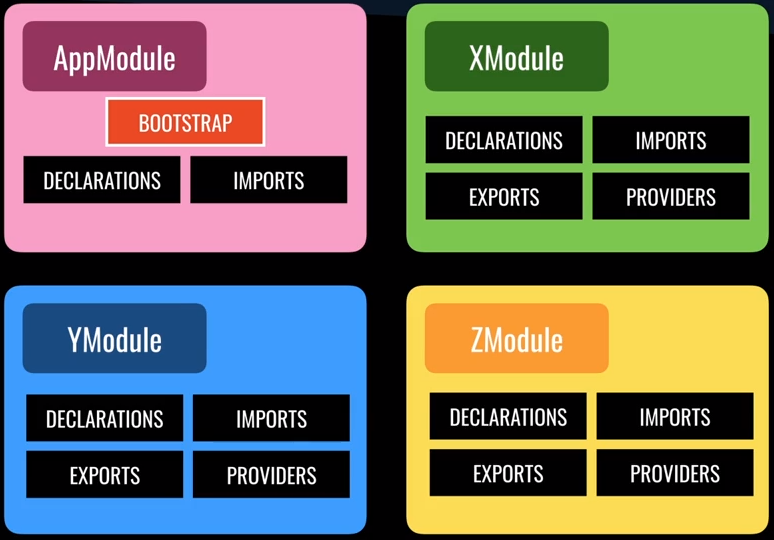
**Exports**: Todos os componentes, diretirvas e pipes que precisarem estar visíveis fora do módulo, precisam ser replicados no atributo Exports.

**Imports** Todos os módulos necessários. Ou seja, qualquer módulo necessário por esse módulo, será relacionado pelo componente imports. Esses imposts podem ser outros módulos da sua própria aplicação ou outros módulos como uma biblioteca de componentes visuais.

**Providers**: São os Services

**Bootstrap**: Será onde definimos os componentes que serão carregados naquele módulo. O componente Bootstrap é necessário apenas no módulo que irá inicializar a aplicação (APP Module).

# Organização Usando Módulo



O AppModule não precisa exportar nada, ele importa os outros módulos.

# Criando APP com o Angular CLI

## CLI Command Line Interface

Instalando uma dependência no Node, essa dependência irá criar no Node um command, para que daí, possamos criar nosso projeto, componentes e módulos.

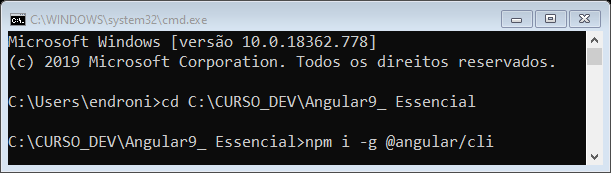
No prompt de comando, navegue até o diretório do seu projeto e digite o comando:

npm i –g @angular/cli

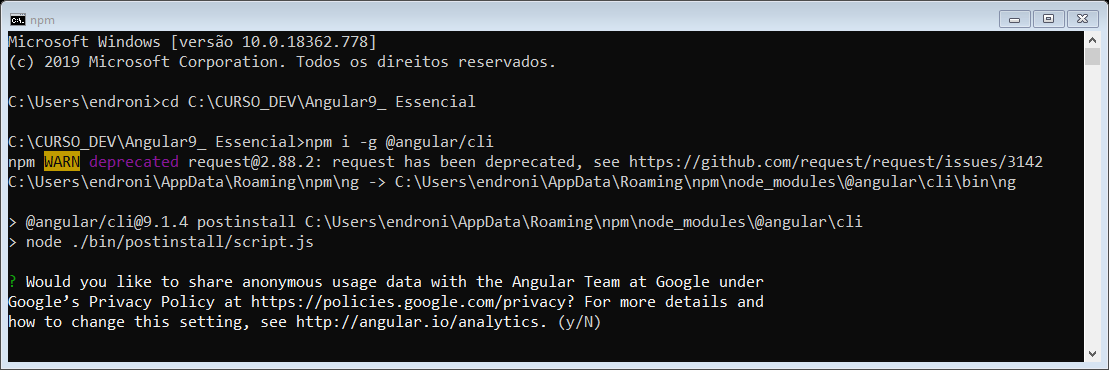
i – de install

g – de global

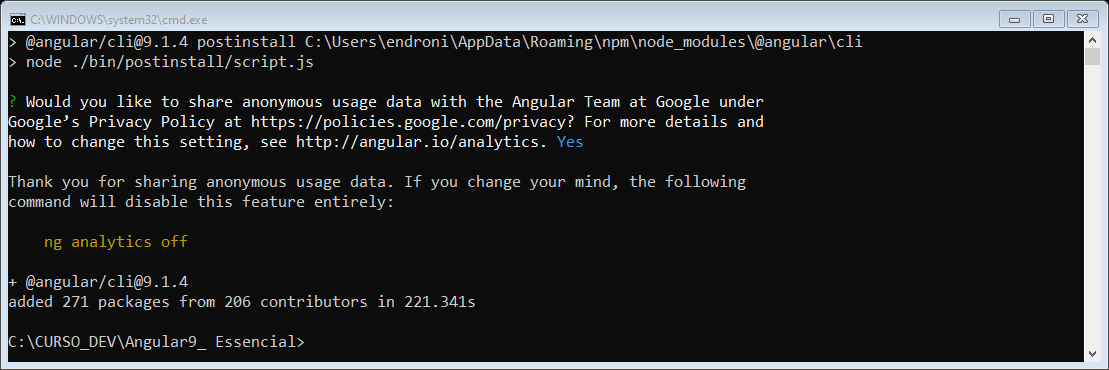
@angluar/cli – comand line interface para uso do angular



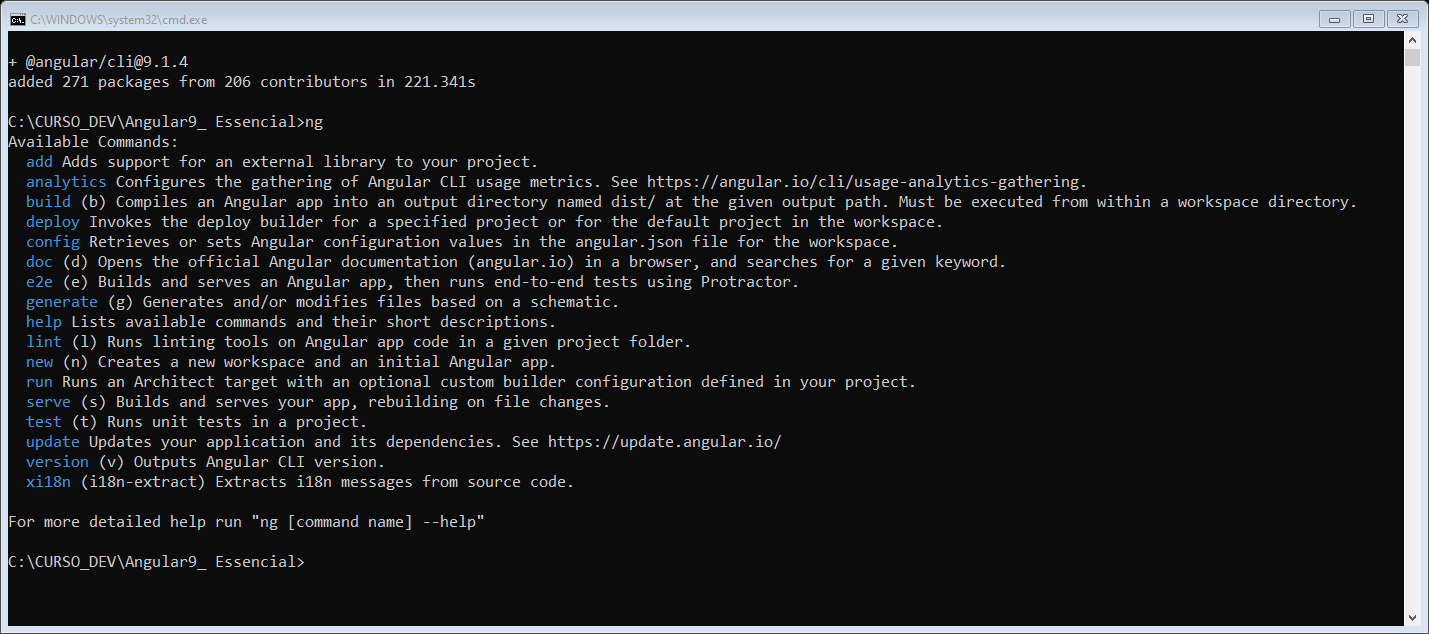
Como o meu projeto será público, vou dar permissões digitando Y e pressionando a tecla Enter do teclado.



Pronto:



O comando ng lista um help para outros comandos



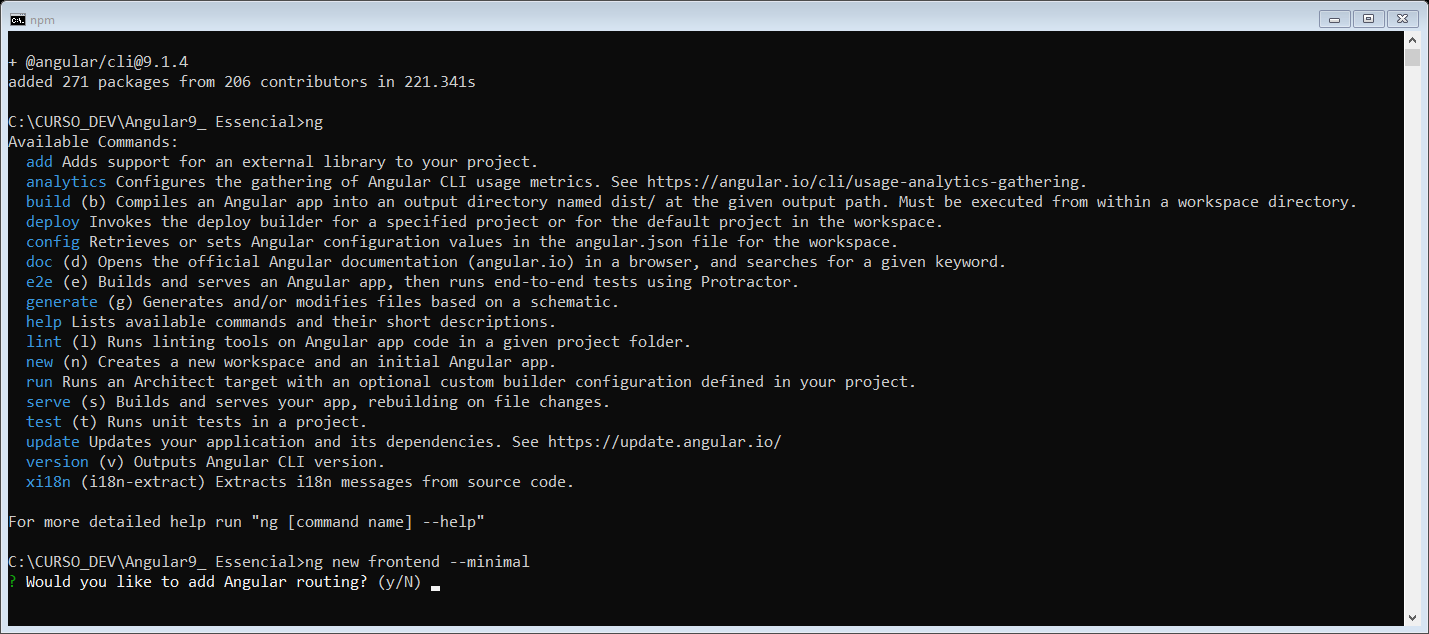
Agora digite o comando abaixo para criar um projeto

ng new frontend ---minimal

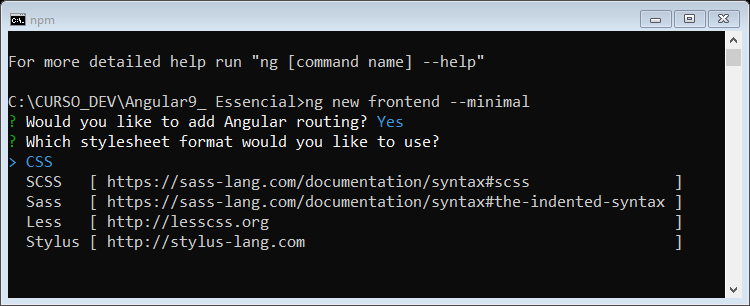
frontend será o nome do meu projeto

--minimal é uma fleg que retira alguns componentes desnecessários.

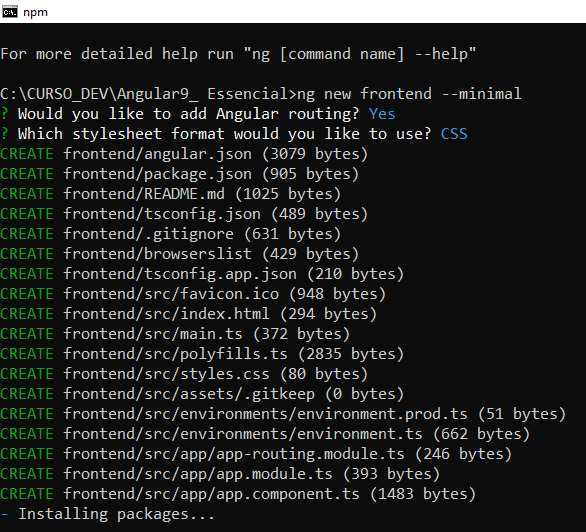
Ele irá fazer duas perguntas. A primeira é se vamos querer colocar a parte de rotas. Vamos dar um Y para confirmar e pressione Enter



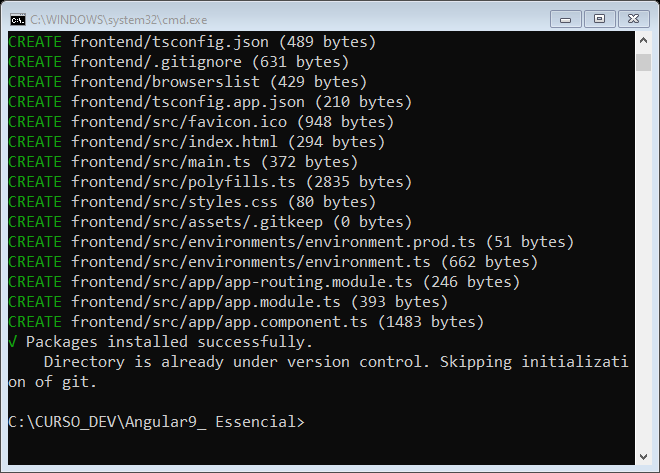
A segunda pergunta é se vamos querer utilizar o CSS, SCSS, Sass ou Less. Vamos utilizar o CSS puro mesmo. Caso ele não esteja selecionado, podemos utilizar a seta do mouse para cima ou para baixo e pressionar a tecla Enter para escolher.



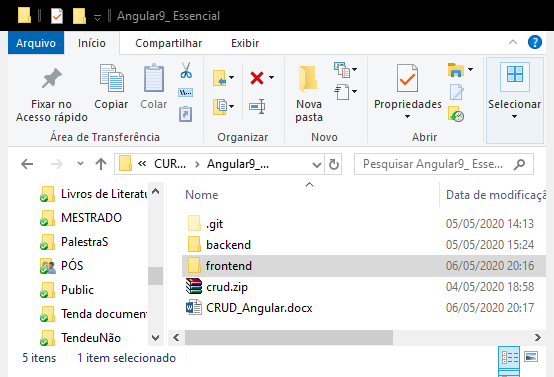
Aguarde a instalação:



Instalado com sucesso

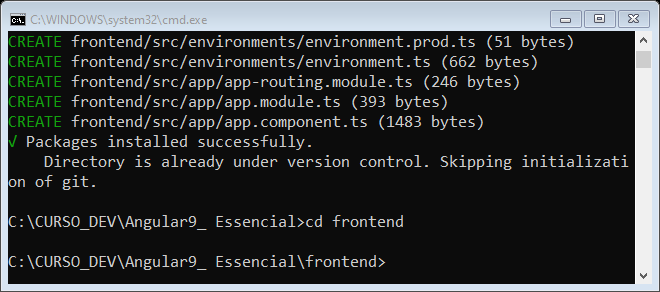


E cria um diretório com o nome que definimos antes

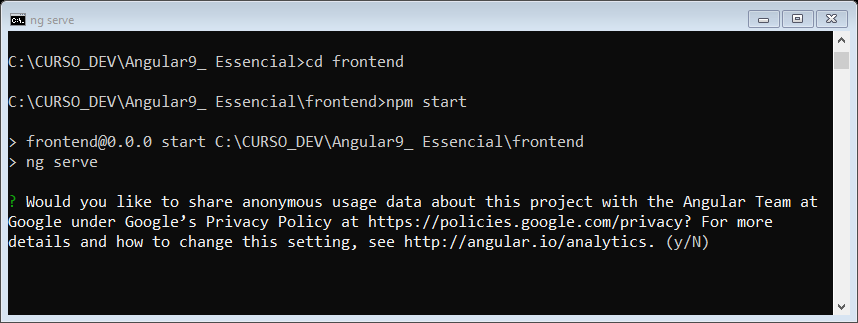


Podemos utilizar o comando cls para limpar.

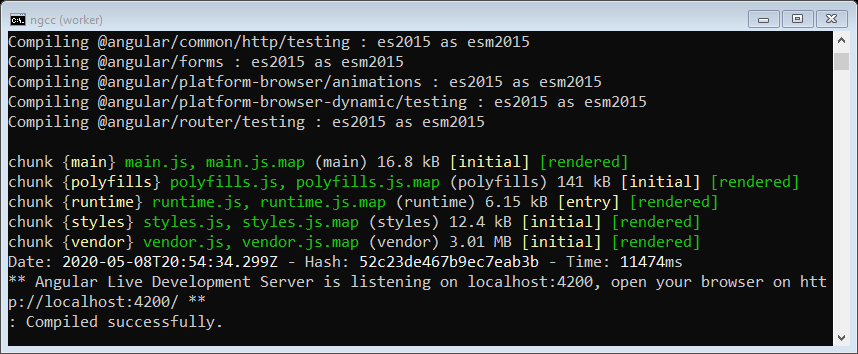
Entre dentro do diretório



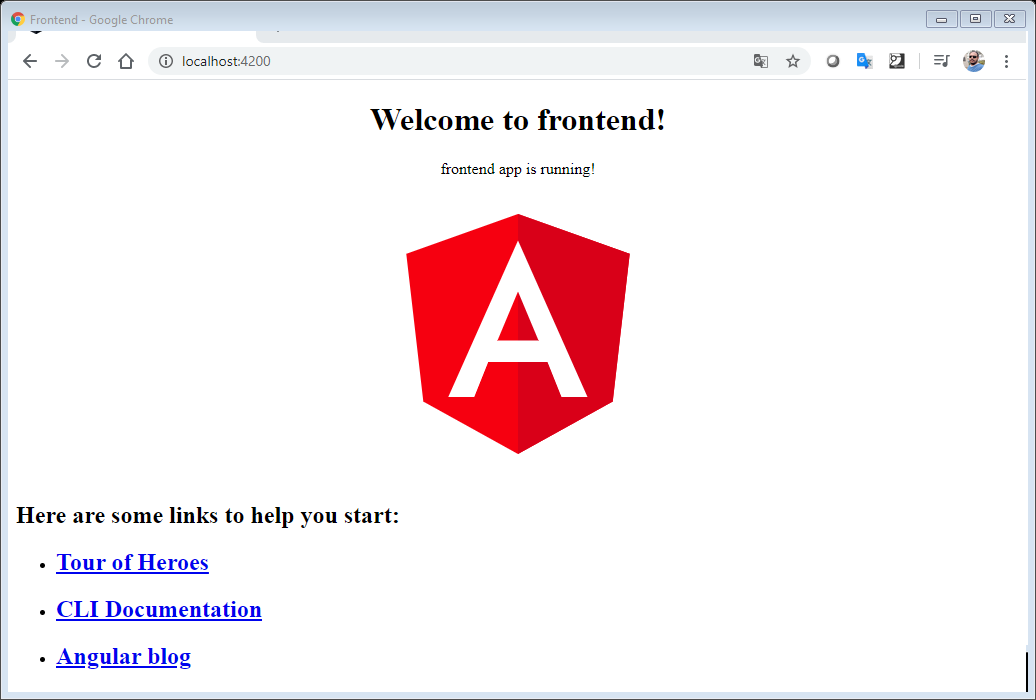
Repita o mesmo comando do backend: npm start



Aceite com y

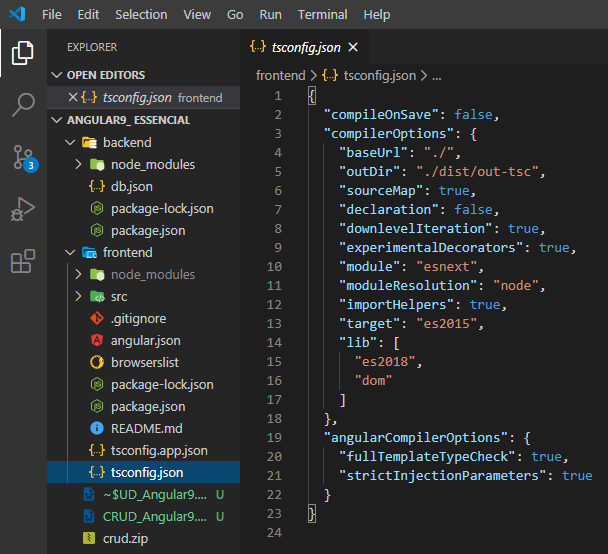


Ao digitar o endereço <http://localhost:4200/>

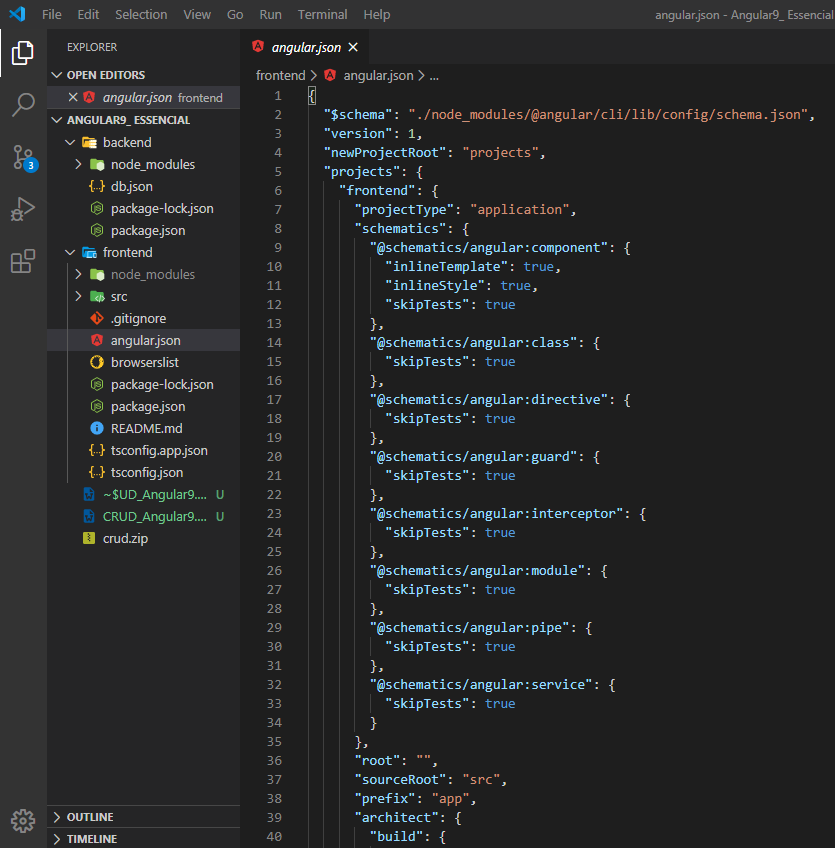


# Um pouco sobre os arquivos e diretórios

O arquivo tsconfig.json é o arquivo onde ficam as configurações



O arquivo angular.json também possui configurações



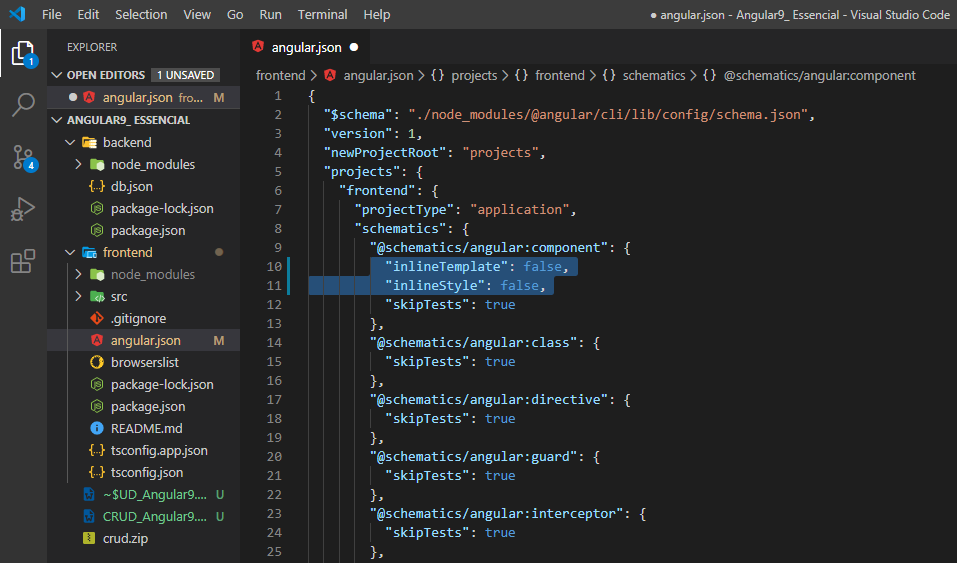
# Alterações no angular.json

## inlineTemplate: false

Para que o HTML venha de um arquivo externo

## inlineStyle: false

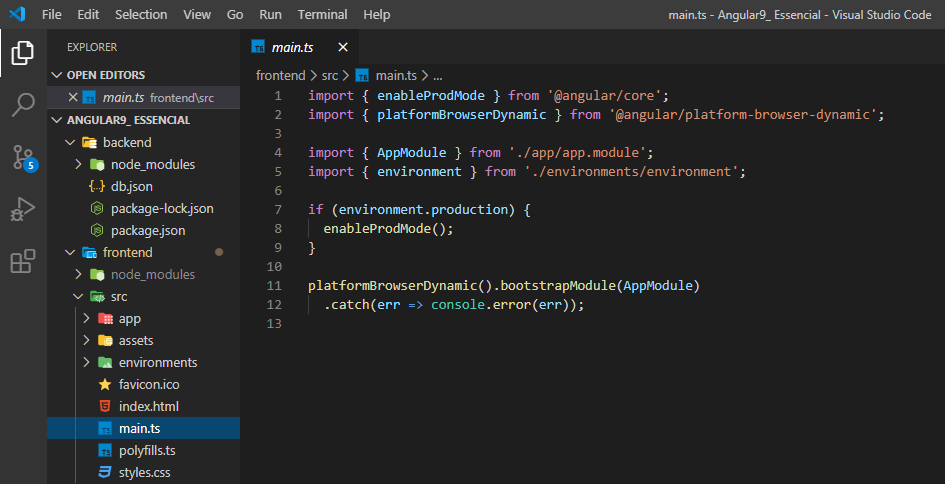
Para que o CSS venha de um arquivo externo



# src

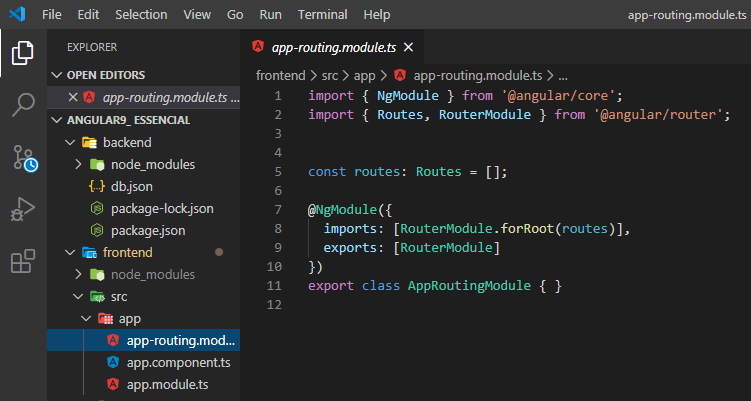
Dentro de src é onde fica o código fonte

main.ts – Porta de entrada, (carrega o módulo principal) é o arquivo que irá importar alguns módulos, como o AppModule.



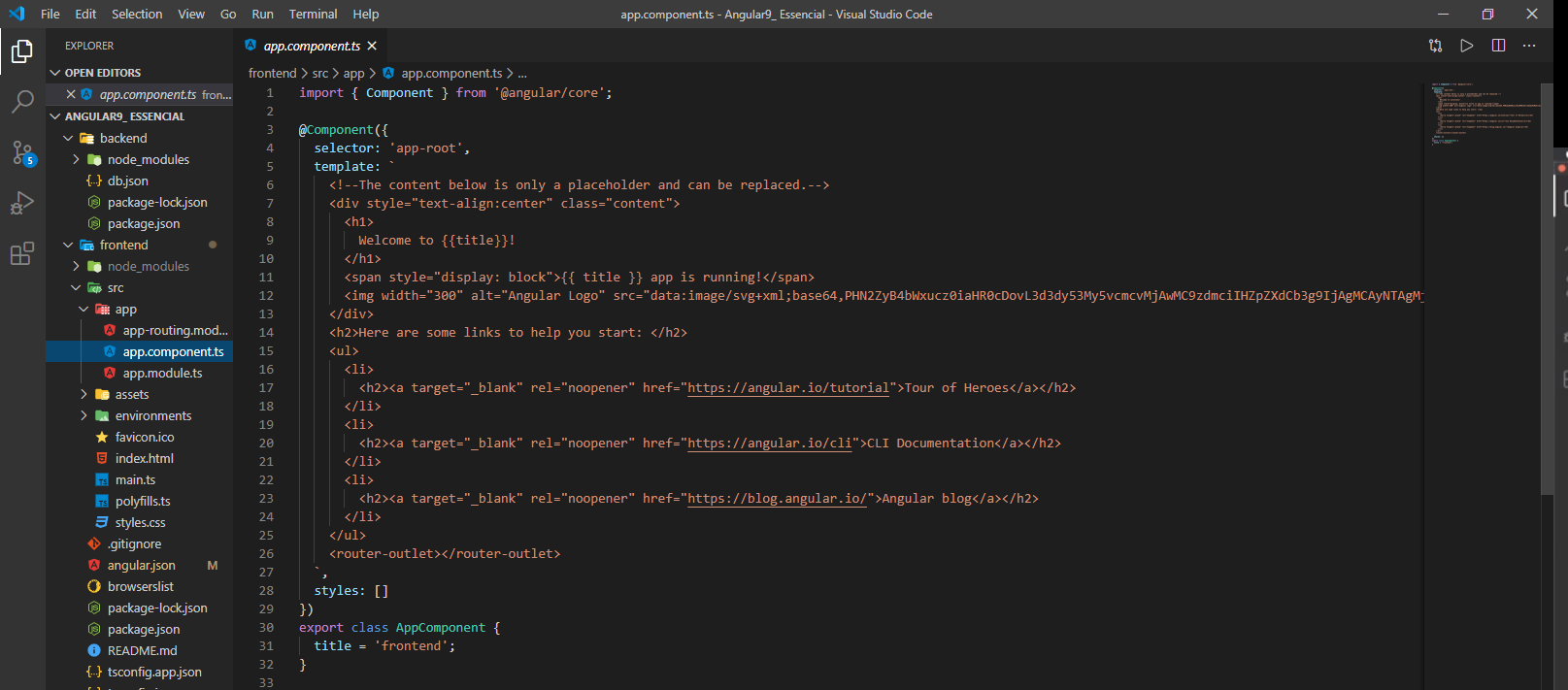
app

Dentro de app temos o módulo de rotas

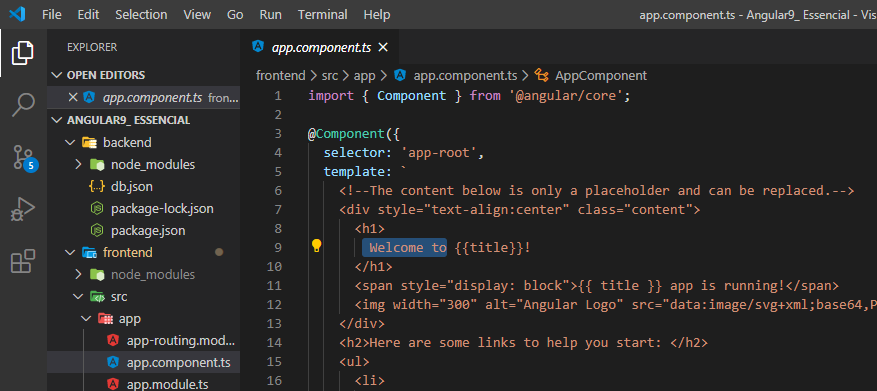


# app.component.ts

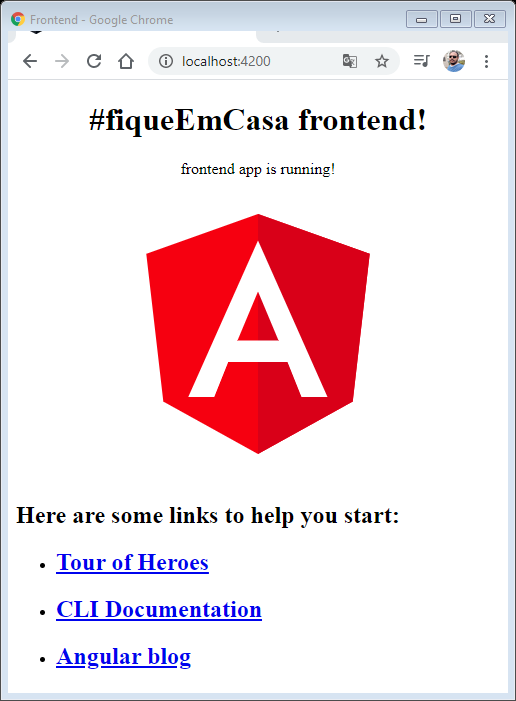
Como nós pedimos para instalar no tipo minimal. As configurações de html ficam dentro do arquivo typeScript no arquivo app.component.ts



Exemplo: Vamos alterar o h1

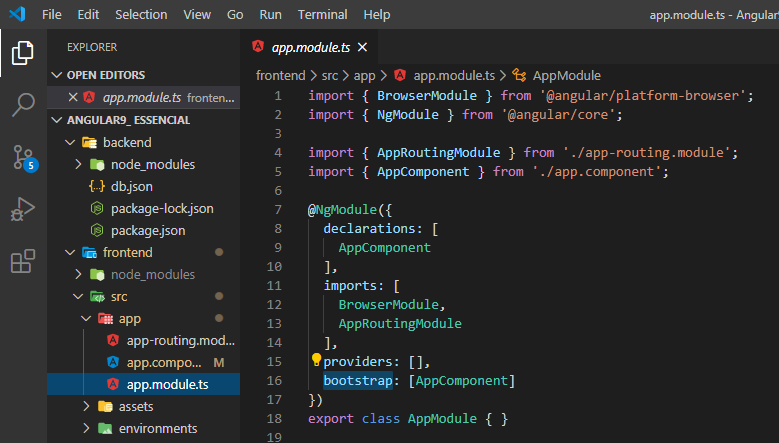


Alterando o html



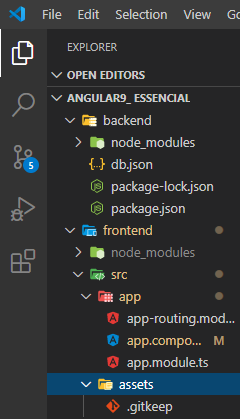
# app.module.ts

Nesse arquivo temos uma parte que é o bootstrap, que aponta para o módulo que irá inicializar o projeto.



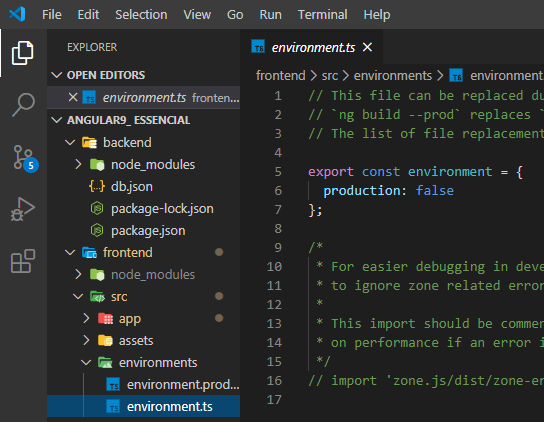
# assets

Dentro de assets ficam imagens, arquivo de fonte



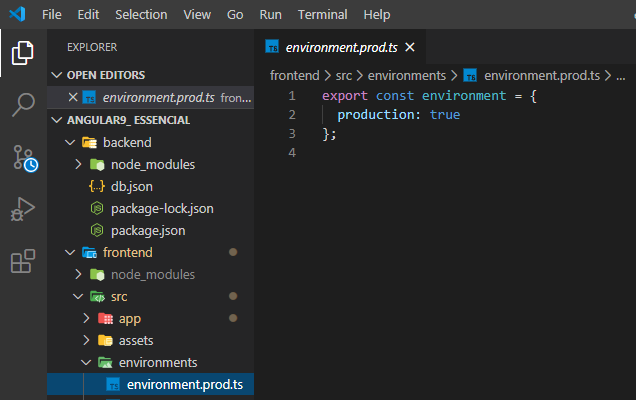
# envionments

Local onde podemos colocar variáveis de componentes.



No momento só temos a variável de componente **production**

# Variáveis de ambiente de produção

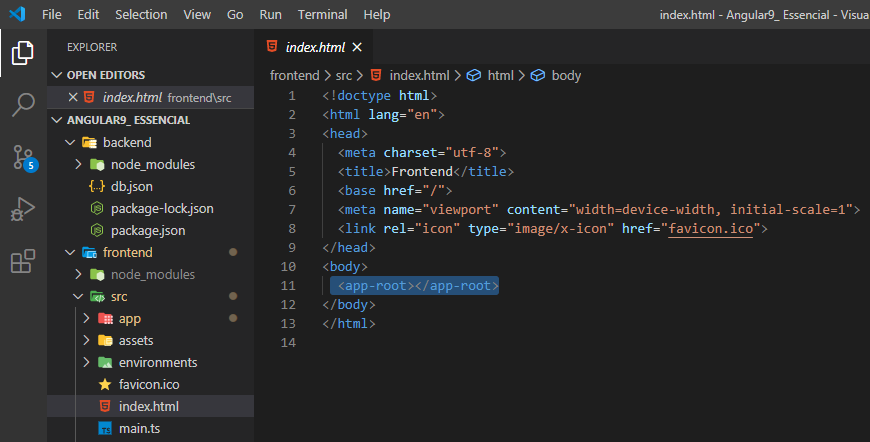


# Single Page Aplication – Aplicação com apenas uma página web

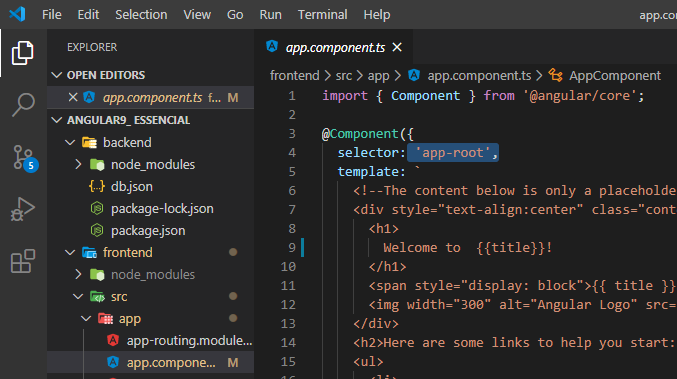
Temos um index.html onde o

 <app-root></app-root>

aponta para o componente raiz



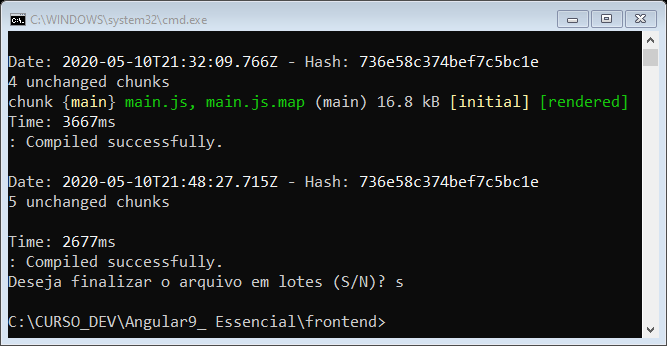
Observe o app-root no **seletor** – Módulo principal



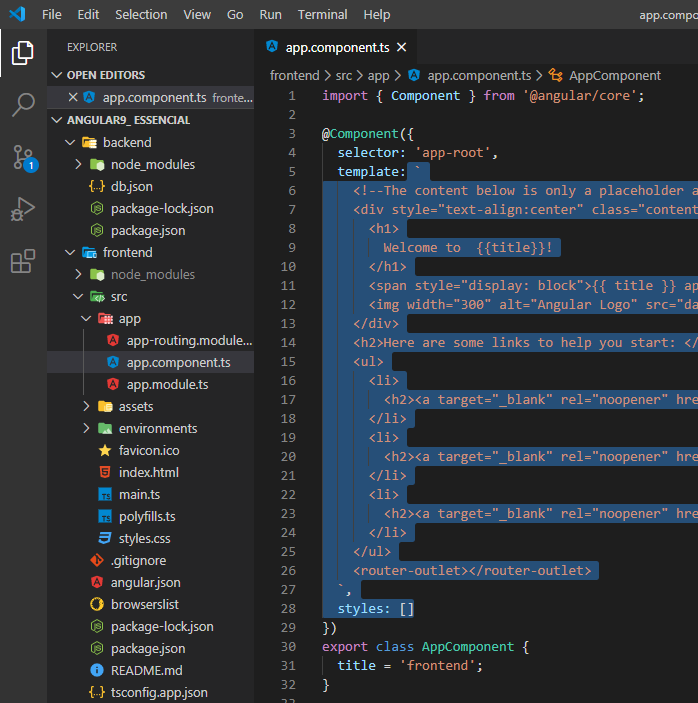
# HTML do AppComponent

Instalar os Componentes do Material

Para continuar, precisamos para o serviço. No prompt de comando, podemos utilizar o Ctrl + C e pressionar a Y para confirmar

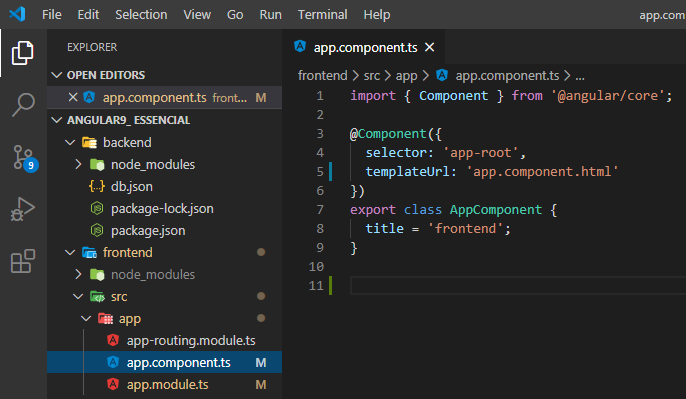


Vamos remover todo o código default do arquivo app.components.js

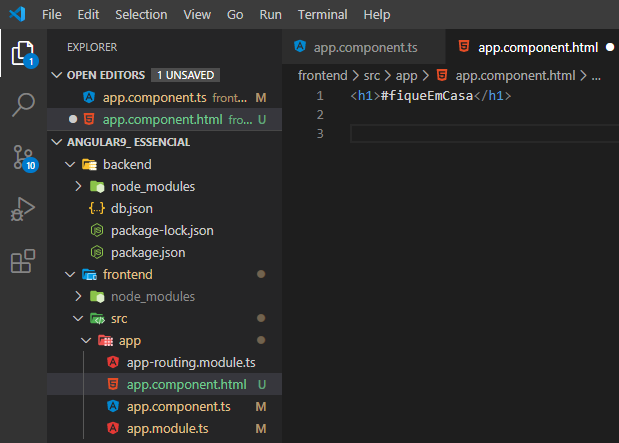


No templateUrl, vamos apontar para o arquivo: ‘app.component.html’

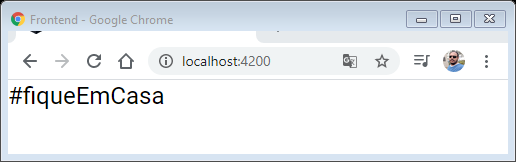
templateUrl: 'app.component.html'



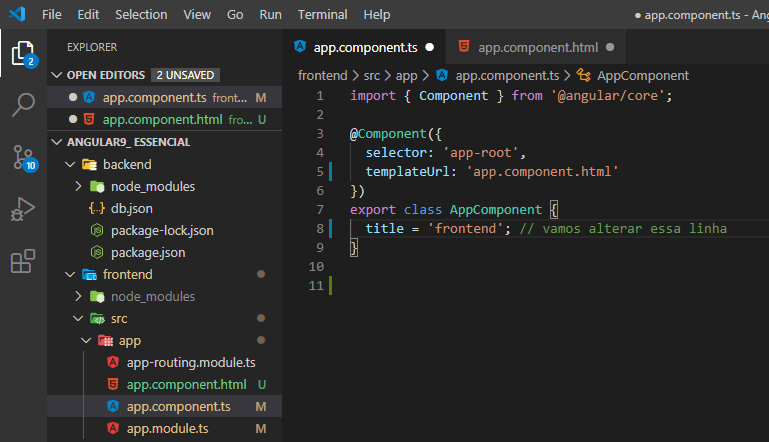
Vamos criar um arquivo com o mesmo nome e no arquivo vamos colocar um <h1> para testar

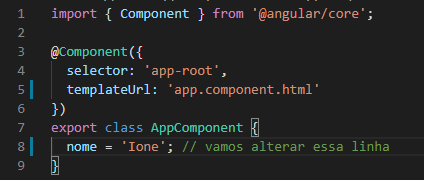


Agora testando no endereço <http://localhost:4200/>



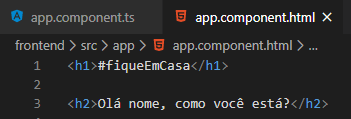
# Um recurso do angular



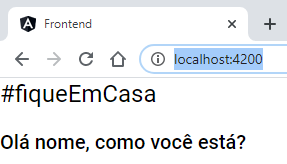


Utilizando variável em um componente externo

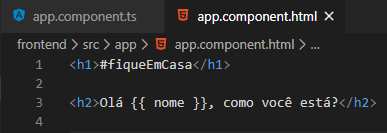
Em app.component.html



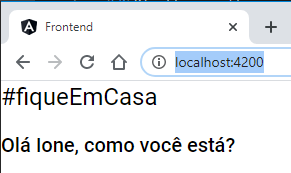
No navegador



Para puxar o valor da variável, envolva a variável com duas chaves

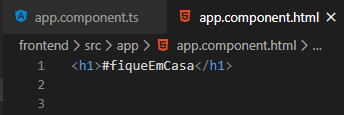


Resultado

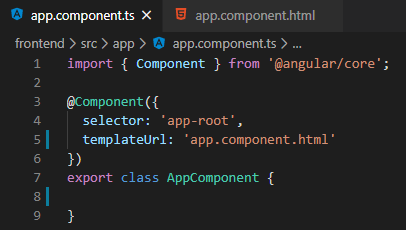


Vamos retirar as linhas que não vamos usar

No app.component.html



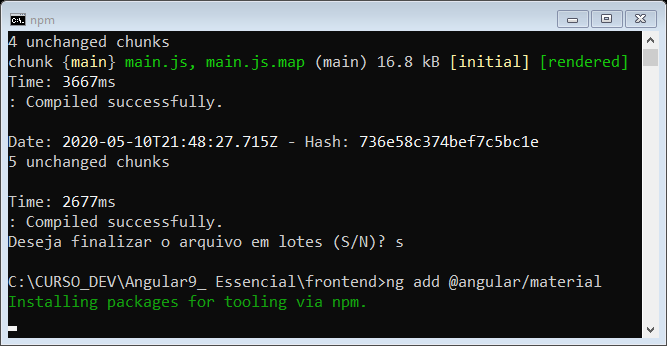
No app.component.ts



# Instalar componentes do material

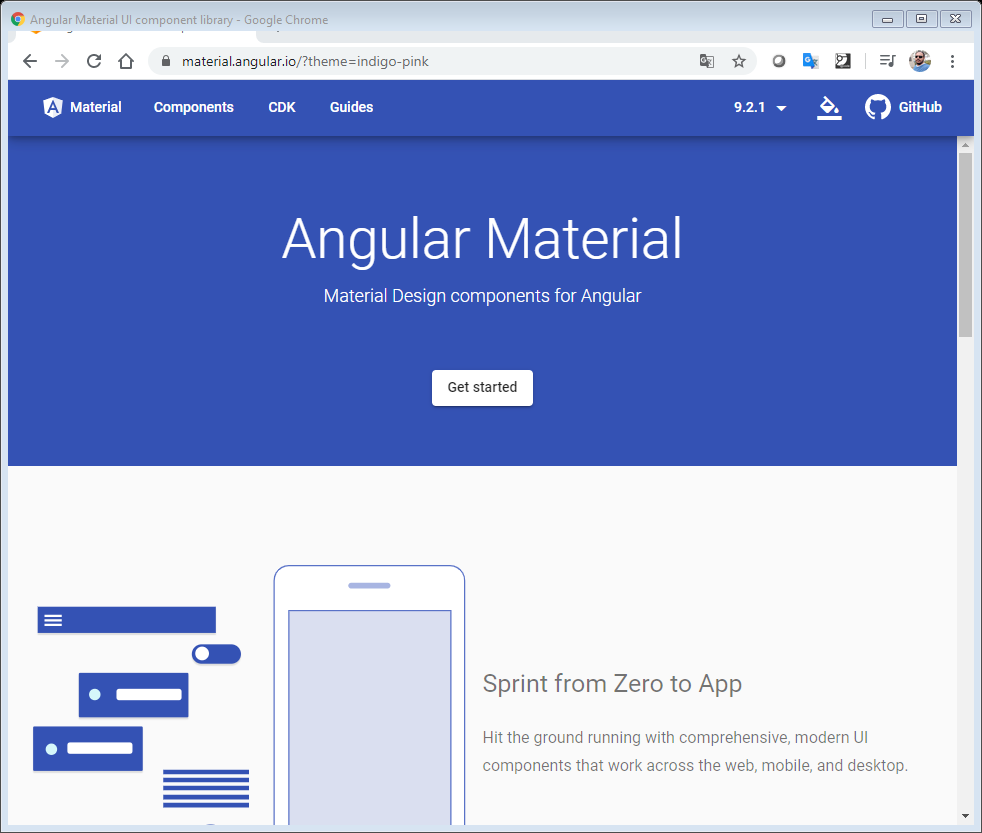
Podemos ter acesso a vários componentes do google instalando o material

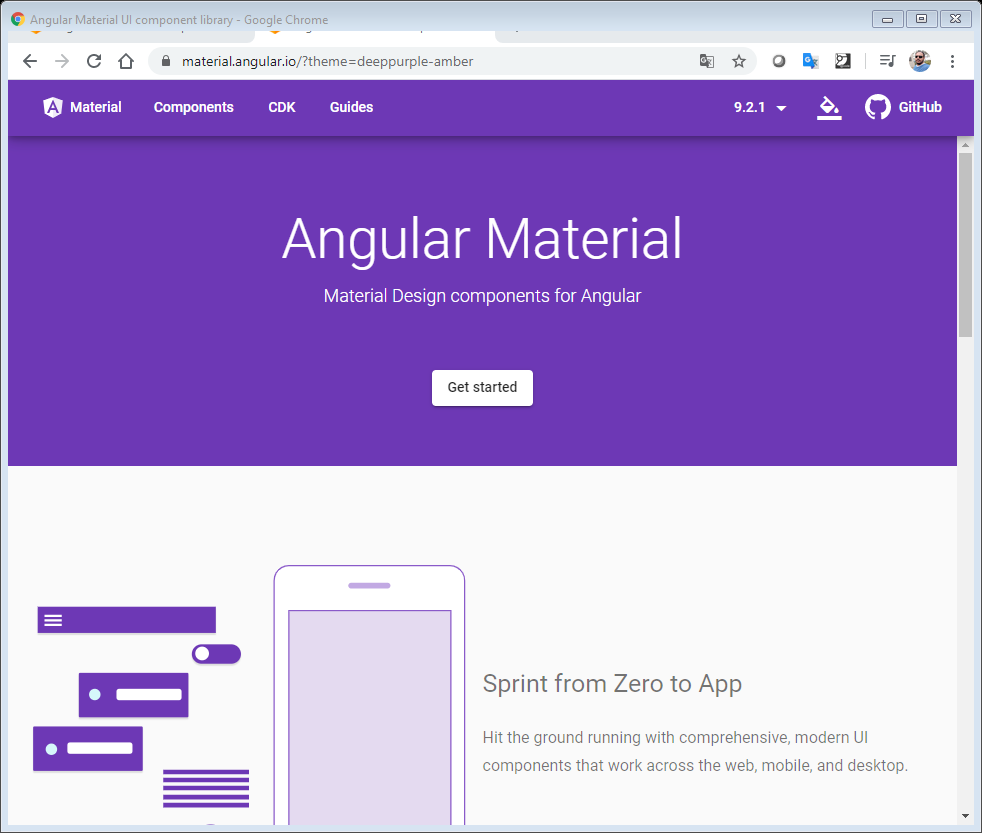
Digite o comando: ng add @angular/material

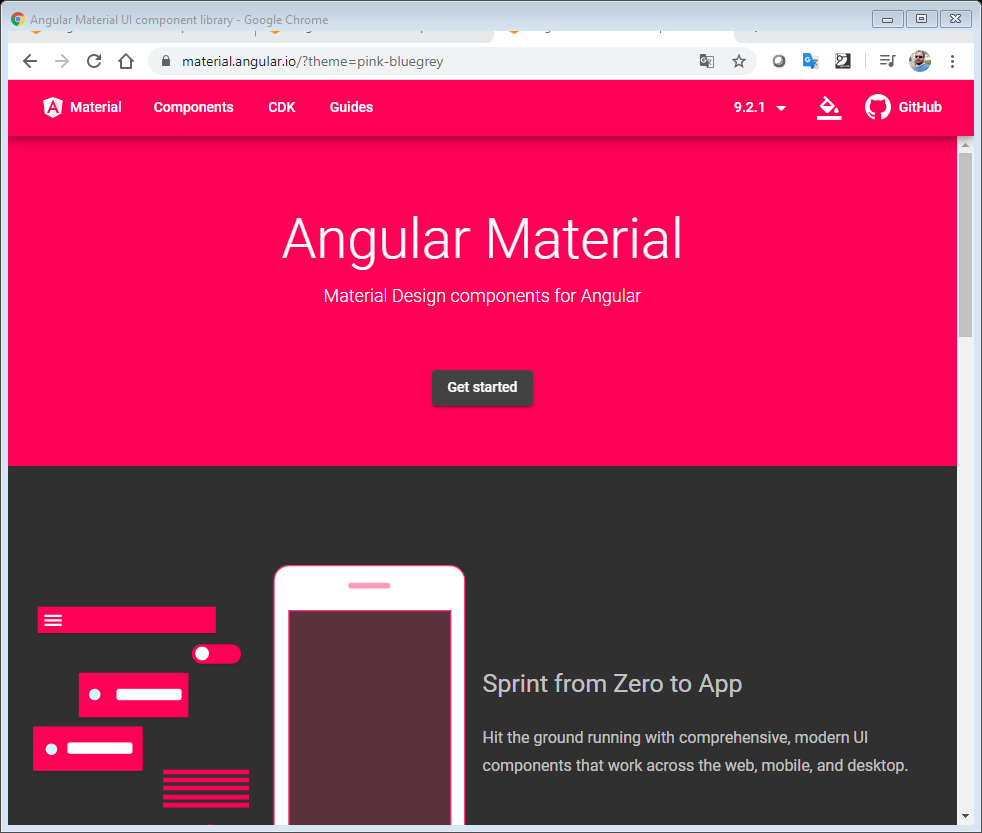


Agora precisamos responder algumas perguntas.

**Primeira pergunta –** Qual é o template (esquema de cores) que queremos utilizar. Podemos copiar a URL para observar quais são esses esquemas

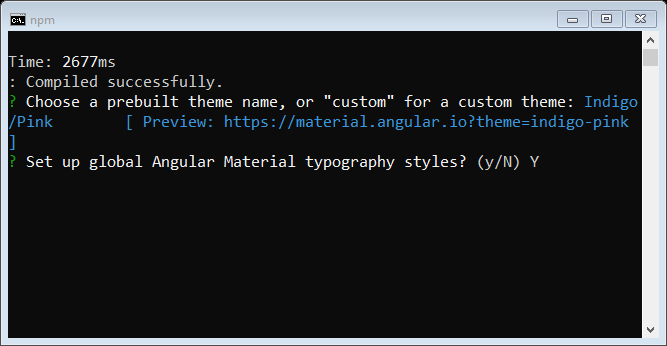






Vou escolher a primeira opção

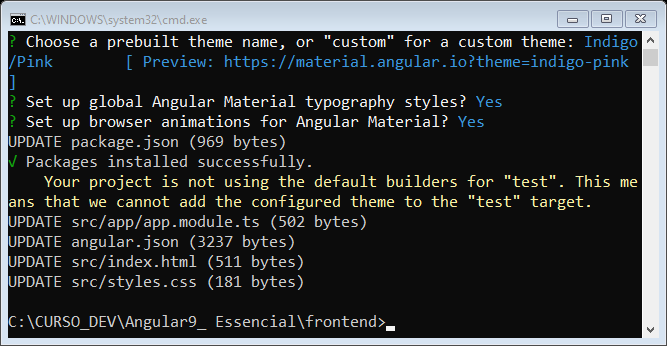
**Segunda pergunta –** Vamos aceitar a **tipografia** de modo global



**Terceira pergunta –** Vou aceitar as animações

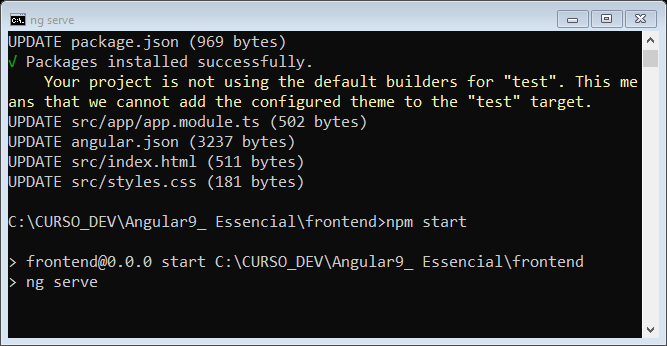


Aguarde a finalização



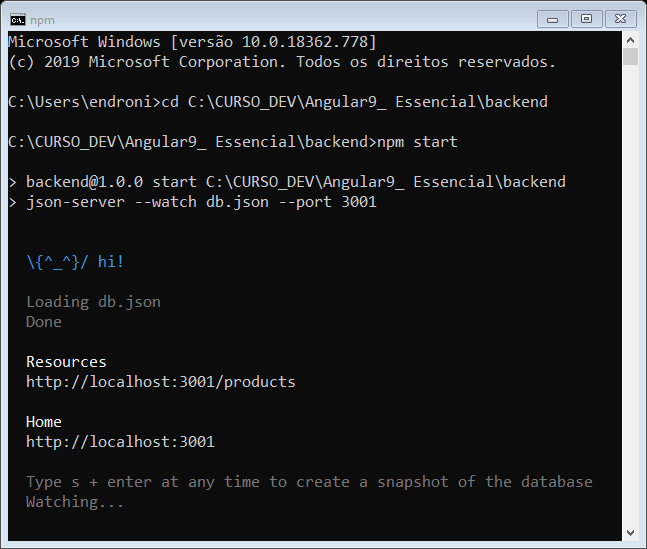
Agora vamos iniciar novamente

npm start

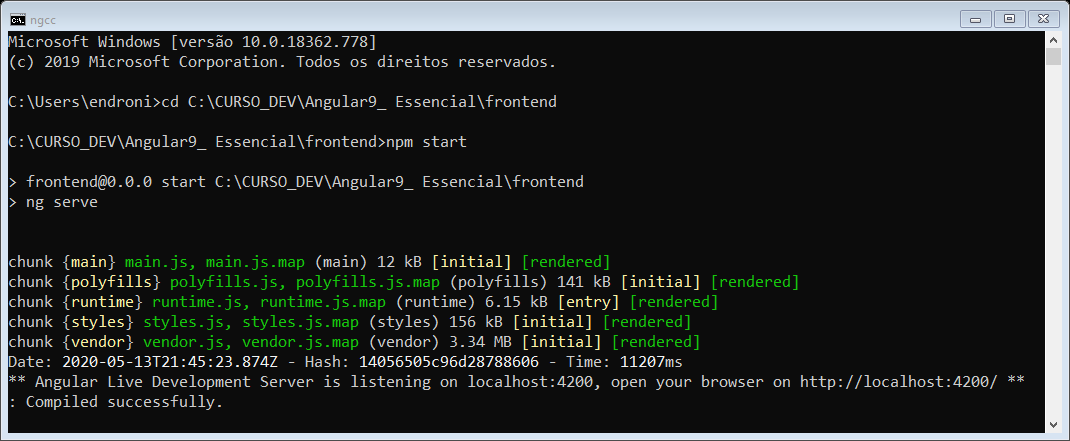


# Componente do Cabeçalho

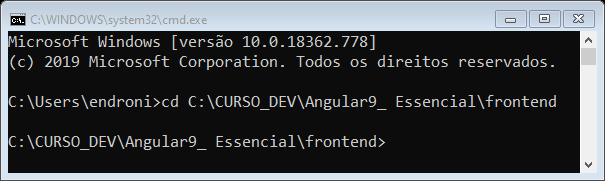
Web Service está rodando como backend



Temos Angular rodando



Vamos abrir outro prompt para instalar outros componentes



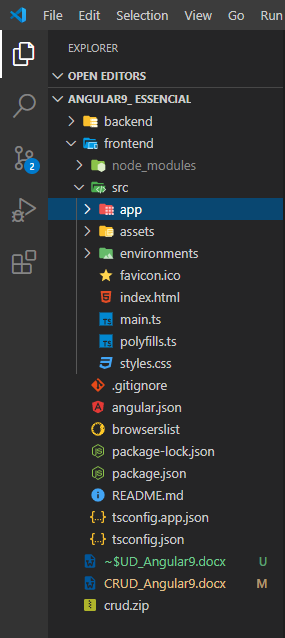
Comandos:

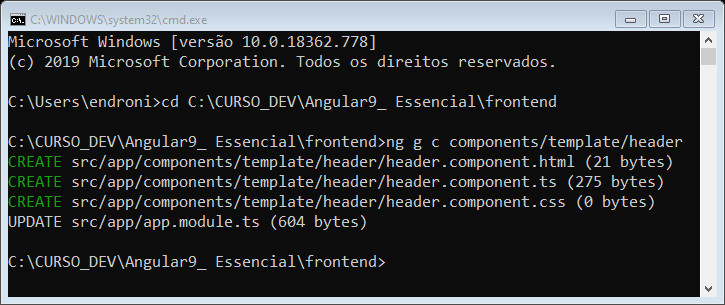
ng generate componente – Para gerar um componente

ng g c – Tem a mesma função do anterior

Vamos criar um componente e registrar dentro do módulo com o comando abaixo:

ng g c components/template/header



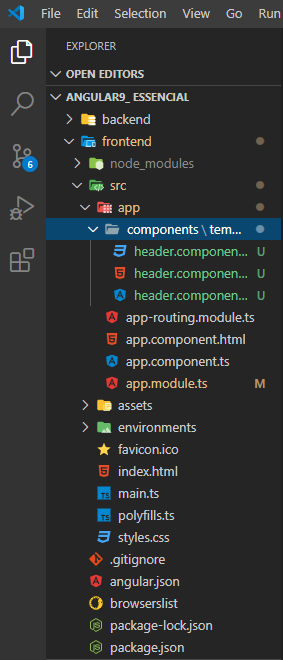


.html

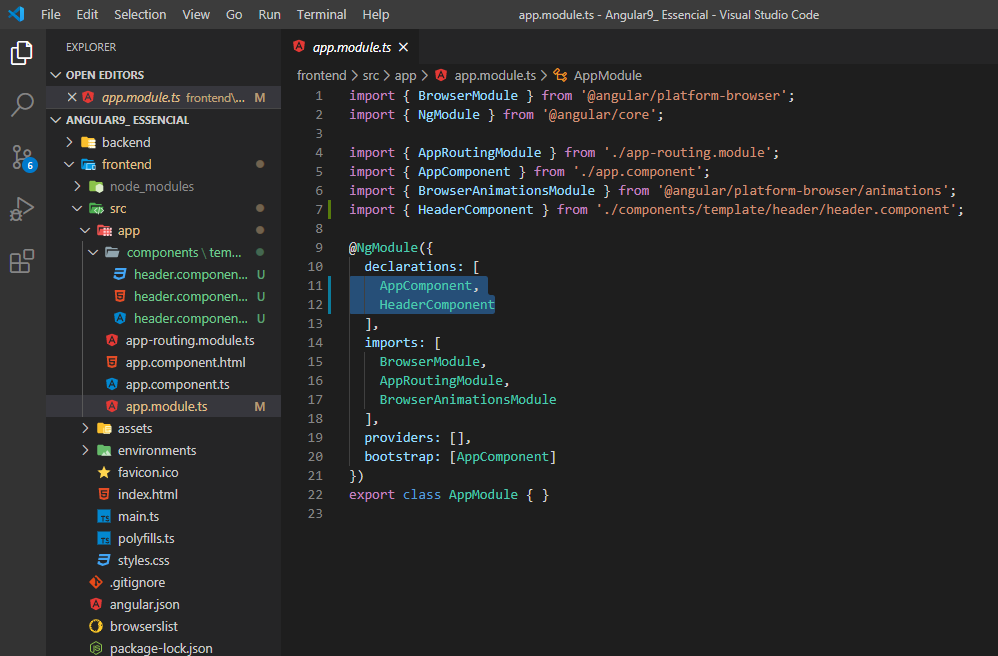
.css

.ts

O diretório componentes foi criado com os arquivos.



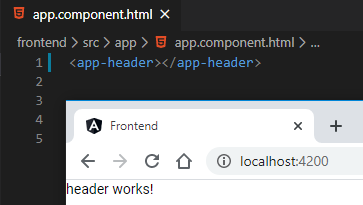
Observe que no arquivo app.module.ts



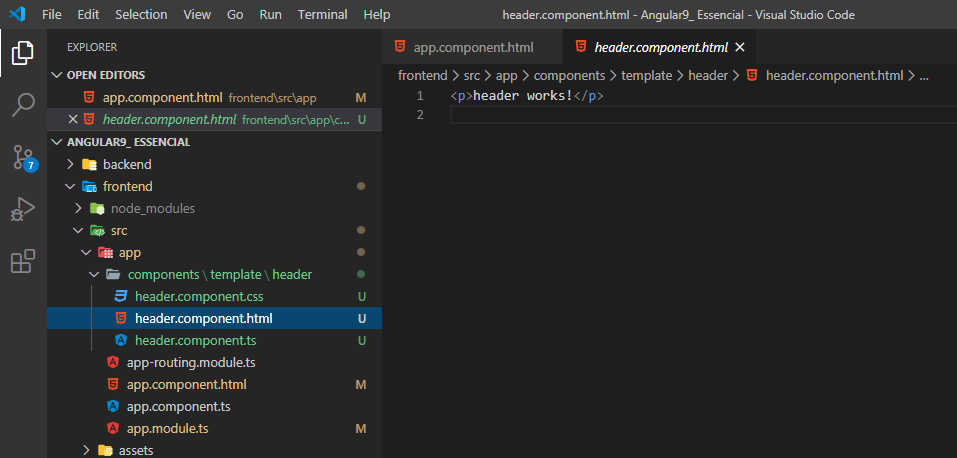
Em app.component.html vamos apagar o h1



Insira a tag <app-header></app-header> e a página já irá se atualizar.

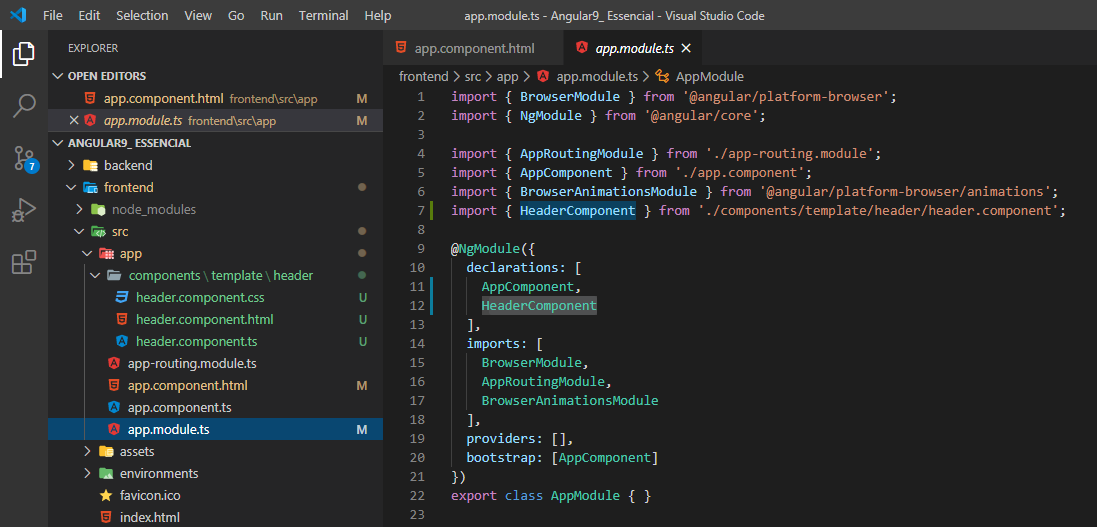


Agora precisamos começar a trabalhar o nosso template no arquivo header.component.html



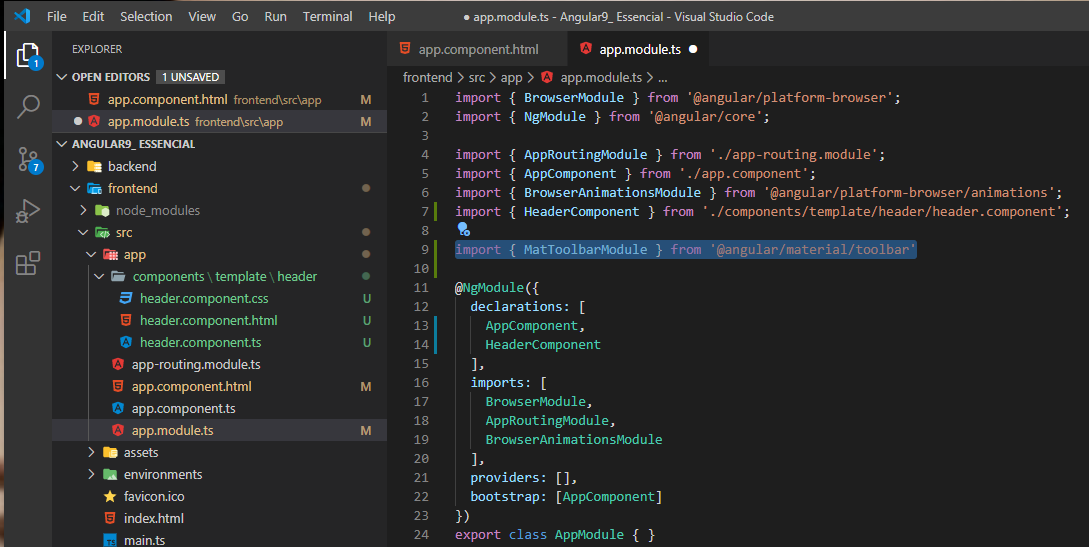
Precisamos importar agora o Toolbar do material que foi instalado anteriormente.

Vamos em app.module.js

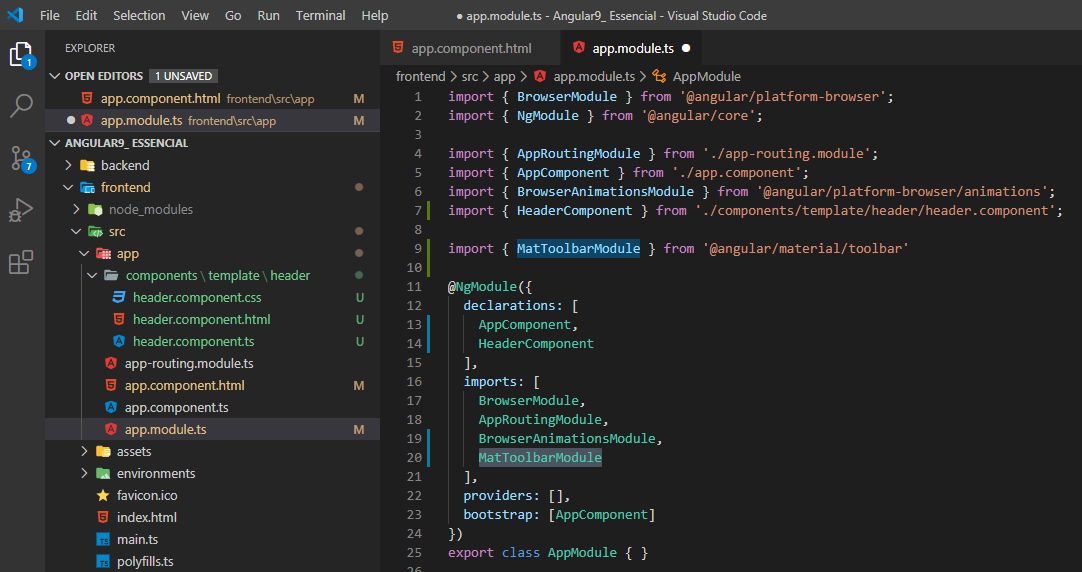


Insira a linha para importar

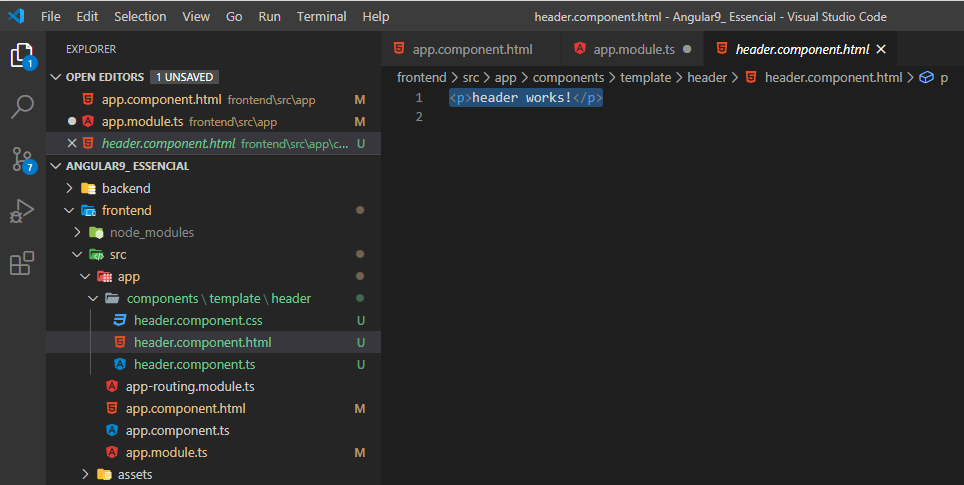
import { MatToolbarModule } from '@angular/material/toolbar';



Todo componente importado precisa ser declarado em imports



Em header.component.html vamos apagar o default

Insira o código

<mat-toolbar class="header mat-elevation-z4">

    <span>

        <a>

            <img class="logo" src="assets/img/logo.png" alt="logo"/>

        </a>

    </span>

    <span>

        <a>

            <i class="material-icons">

                dashboard

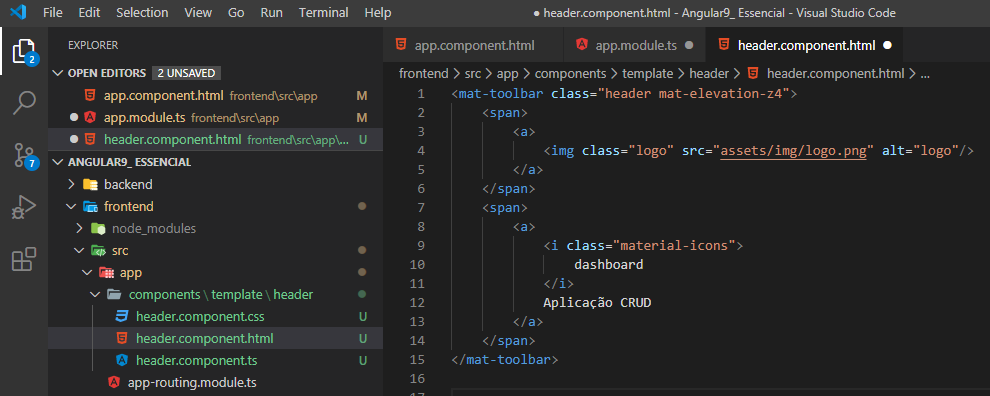
            </i>

            Aplicação CRUD

        </a>

    </span>

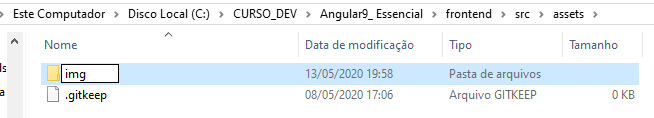
</mat-toolbar>



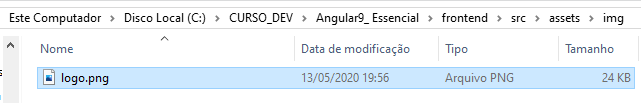
Vamos colocar uma logo.png no endereço

C:\CURSO\_DEV\Angular9\_ Essencial\frontend\src\assets

Crie um diretório img



Coloque a logo dentro da pasta img



No arquivo header.components.css insira o código

.header {

    display: flex;

    align-items: center;

}

.header a {

    display: flex;

    align-items: center;

    text-decoration: none;

}

.header .logo {

    max-height: 32px;

}

.header .title-group {

    padding-left: 25px;

}

.header .title-group i {

    padding-right: 5px;

}



# Github

https://github.com/endroni/CRUD\_Angular9.git

Repositório do Curso: <https://github.com/cod3rcursos/angular-crud>

# Conclusão

A linguagem JavaScript tem se popularizado muito dentre as linguagens de programação. É uma linguagem muito robusta, que não está mais presa apenas em websites, mas rompeu essa barreira e hoje é encontrada também em grades servidores rodando inúmeros serviços.

# BIBLIOGRAFIA BÁSICA:

FLANAGAN, David. JavaScript: o guia definitivo / David Flanagan; tradução: João Eduardo Nóbrega Tortello; revisão técnica: Luciana Nedel. – 6. ed. – Porto Alegre: Bookman, 2013. O Guia Definitivo. ed. O’Reilly.

SILVA, Mauricio Samy. **JavaScript: Guia do Programador:** Guia Completo das Funcionalidades de Linguagem JavaScript. 6ª edição. São Paulo. ed. Novatec 2010.

# BIBLIOGRAFIA COMPLEMENTAR:

MDN web docs. Mozilla. **O que é JavaScript?** Aprenda o melhor em desenvolvimento web. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript/First\_steps/O\_que\_e\_JavaScript>. Acesso em: 19 de março de 2020.

Documentação Oficial do Ecma. Standards@Internet Speed. Disponível em: <https://www.ecma-international.org/>. Acesso em: 21 de março de 2020.