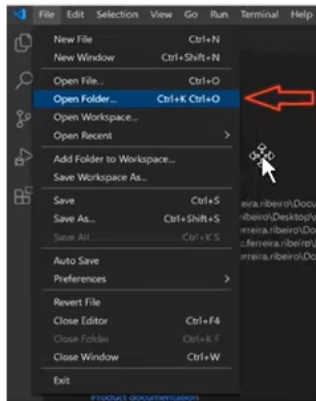
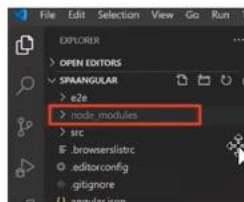


Abrindo seu projeto no Visual Studio Code

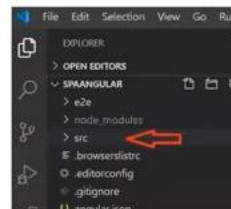


- Abrir a pastinha através do commando Ctrl+K / Ctrl+O.
- Este processo trará toda estrutura inicial do Angular para dentro de sua IDE.

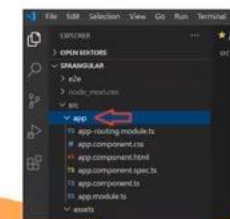
Conhecendo alguns diretórios



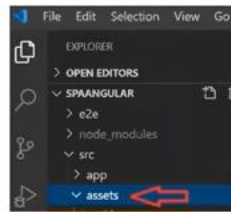
- Aqui estão todos os pacotes e dependências que o Angular NPM baixou e instalou na nossa aplicação.



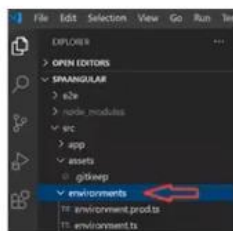
- Dentro de SRC existem algumas pastinhas que ele já traz pronta e que fazem parte da estrutura inicial.



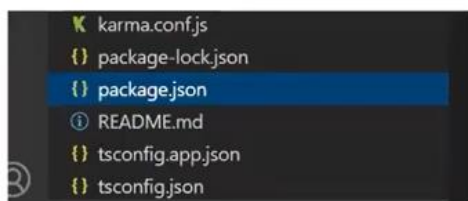
- Na pasta APP, ficam basicamente todos os nossos componentes e arquivos de modulo.



- Na pastinha Assets, colocaremos nossos arquivos de estilo comuns em toda a aplicação.



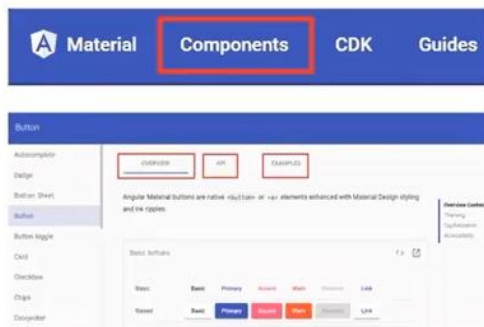
- Esta pastinha nos ajuda a criar diferentes ambientes para que a aplicação tenha um comportamento único em cada um deles, facilitando no deploy do código.



- Neste arquivo estão as dependências do nosso projeto. Aqui Podemos ver as versões utilizadas em cada pacote.

Utilizando o Angular Material

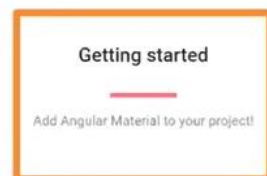
Para nosso projeto iremos utilizar o Angular Material (<https://material.angular.io/>). Neste site conseguimos visualizar todos os componentes prontos que ele traz.



Ao acessar o site, clicando em **Components**, conseguimos visualizar uma lista com todos os componentes que Podemos utilizar.

Escolhendo um componente, ele traz para você um **overview** sobre aquele componente, mostra também a **API**, ou seja, o módulo que teremos de importar para utilizar este componente em nossa aplicação, e por último, alguns **Exemplos** de utilização, para que você tenha uma previa de como ficará este componente na sua aplicação.

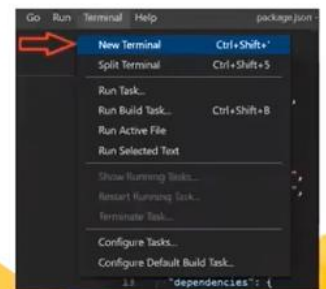
Na página inicial do site do **Angular Material**, clicando em **Getting started**, ele mostrará o que precisamos fazer em nossa aplicação para utilizarmos estes componentes prontos em nossa aplicação.



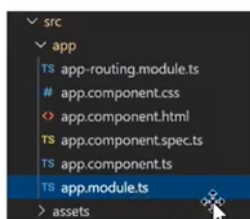
Começaremos instalando o Angular Material utilizando o seguinte comando: `ng add @angular/material`. Após executar este comando, irá aparecer uma mensagem para que você escolha o tema do Angular Material que você deseja utilizar. Para esta aula, iremos escolher a primeira opção: **Indigo/Pink**. Em seguida ele irá perguntar se você deseja utilizar os estilos globais do Angular, vamos utilizar e prosseguir. Em seguida ele irá perguntar se queremos utilizar as animações, aceitamos e prosseguimos.



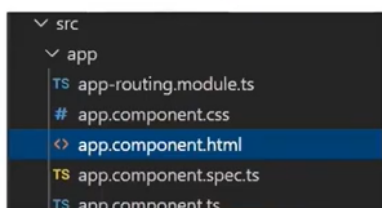
Dica: Clicando em Terminal / New Terminal, você consegue colocar todos os comandos nele e fazer a mesma execução que faria em um terminal externo.



Criando nosso primeiro componente



Dentro de nossa pastinha APP, temos o arquivo: **app.module.ts**, ele é o módulo raiz da nossa aplicação. Sempre que formos utilizar um novo componente, precisamos adicioná-lo neste arquivo para que nossa aplicação passe a enxergá-lo. Tanto os componentes criados por nós, como os que importamos, como por exemplo o do Angular Material.



O arquivo **app.component.html** é aquele arquivo da página inicial do Angular, que foi criado quando iniciamos um novo projeto, contendo a página de Boas vindas do Angular.



O arquivo **app-routing.module.ts** é responsável por mapear as rotas, falando para nossa aplicação qual arquivo ele deve abrir.

No arquivo **app.component.css** Podemos definir o estilo que será aplicado para este componente em específico.

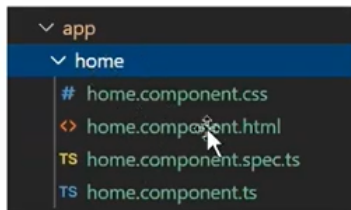


Dica: Iremos criar a aplicação nas pastas raiz do projeto, sem nenhum tipo de estruturação, pois amanhã vocês terão uma aula exclusiva para isso.

Para criarmos nosso primeiro componente, iremos utilizar o comando abaixo:

ng g c home (g = generate / c = component / home = nome do nosso componente).

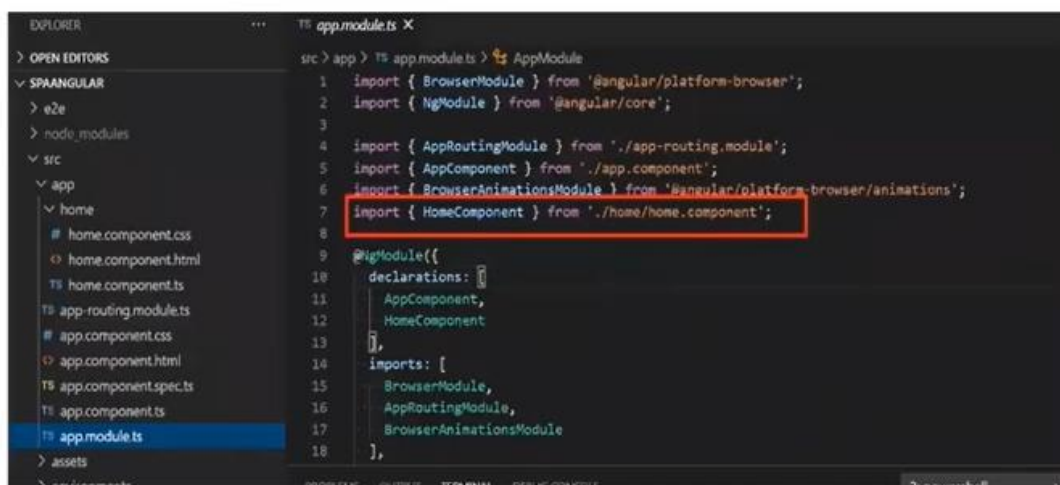
Após este comando, teremos:



Para esta aula não iremos utilizar o arquivo **home.component.spec.ts**, então, podemos excluí-lo, e ficar apenas com os arquivos de CSS, HTML e o Type Script.



Dica: Este arquivo está relacionado aos testes de sua aplicação. Para saber mais acesso o link: (<https://angular.io/guide/testing>)



Ao criar nosso módulo, no arquivo `app.module.ts`, é possível ver que ele foi automaticamente importado e inserido nas declarações deste arquivo.



Dica: Utilizando o atalho: CTRL + K + S você consegue visualizar todos os atalhos disponíveis no VSCode.