

VISUALIZANDO UM COMPONENTE

Vamos criar uma **view** a partir do entendimento da arquitetura do Angular.

1. No terminal no VSCode, digite **ctrl + c**, para poder digitar, e depois digite: **ng g c views/home**.

```
PS C:\Users\vstef\angular\produto> ng g c views/home
CREATE src/app/views/home/home.component.html (19 bytes)
CREATE src/app/views/home/home.component.spec.ts (612 bytes)
CREATE src/app/views/home/home.component.ts (267 bytes)
CREATE src/app/views/home/home.component.css (0 bytes)
UPDATE src/app/app.module.ts (582 bytes)
```

Você sabia?

As letras que utilizamos significam: **“g”**: **generate** e **“c”**: **component**; **views pasta -> home** com nome dos arquivos criados.

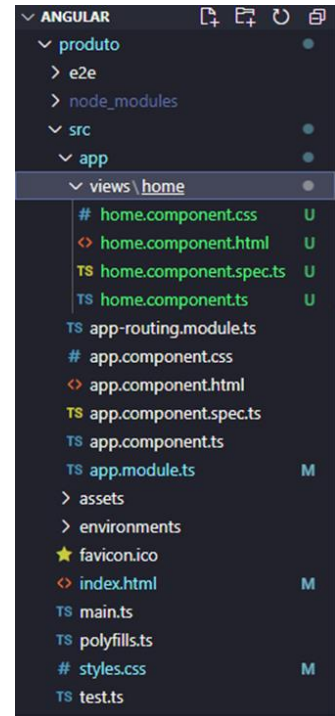


É possível criar os arquivos manualmente, porém você terá que editar e adicionar informações que são pré-configuradas. Quando você utiliza o **comando ng g c**, serão criados todos os componentes em **html**, **css**, **ts** e **spec.ts**, automaticamente, e esses arquivos passam dados.

Por exemplo, no arquivo **app.module.ts** estarão as importações dos arquivos que você criou.

No entanto, ao criar uma pasta e todos os arquivos manualmente, você precisará fazer as importações e as declarações no **arquivo app.module.ts** (arquivo por arquivo).

Já o comando de “**ng g c <nome do componente>**” (no Terminal) faz isso automaticamente, trazendo as configurações e evitando o trabalho manual.

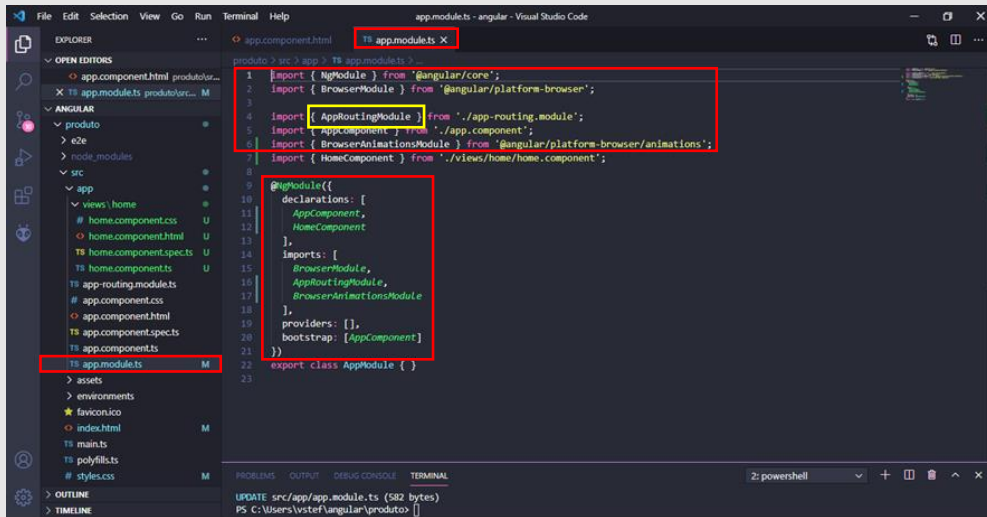


```
//páginas
import { HomeComponent } from './views/home/home.component';
import { HeaderComponent } from './views/header/header.component';
import { FooterComponent } from './views/footer/footer.component';
import { JogosComponent } from './views/jogos/jogos.component';

@NgModule({
  declarations: [
    AppComponent,
    HomeComponent,
    HeaderComponent,
    JogosComponent,
    FooterComponent,
  ],
})
```

Para essa aplicação, não utilizaremos o arquivo com o final **.spec.ts (teste)**, pois o foco deste curso é a introdução básica ao *framework* Angular.

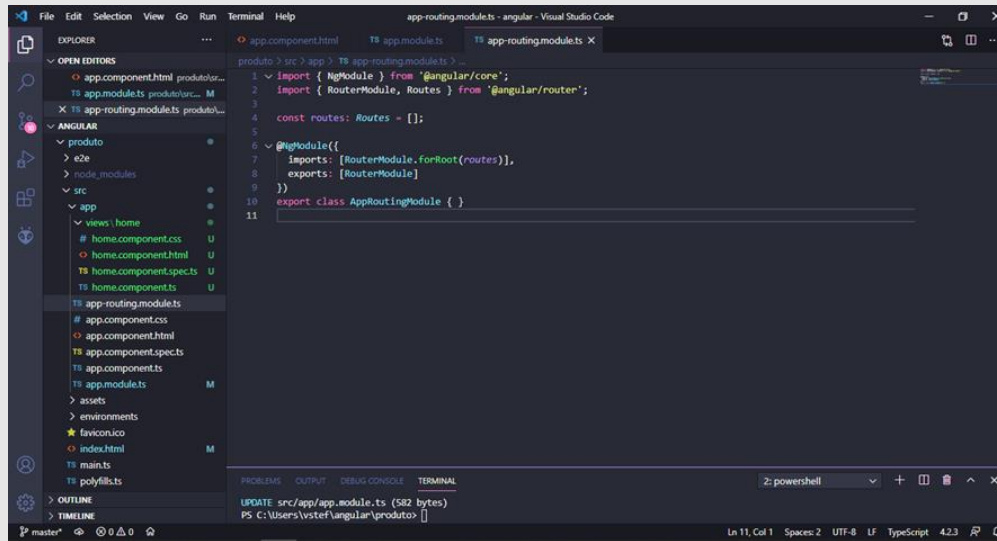
2. Ao criar um arquivo pelo Terminal, são adicionadas informações a outros arquivos, por exemplo, ao **app.module.ts**:



a) O **projeto** já faz a **importação** de alguns componentes, como nas **linhas 2, 4, 5 e 6**.

b) Perceba que, à linha 9, são adicionados os nomes dos arquivos de componentes do Front-End: **declarations** e **imports**, que são configurações de módulos importados pela **raiz AppModule**. Quando você importa, o projeto permite o uso, pois reconhece que ele faz parte.

Na tela a seguir, temos o **app-routing.module.ts**, um arquivo de configuração utilizado para criar **rotas de mapeamento** no site. Como padrão, ele vem configurado conforme apresentamos; além disso, ele será o local onde, posteriormente, adicionaremos informações.



Essas importações são utilizadas para arquivos, componentes e elementos da biblioteca Material.

3. Para criar uma pasta, digite: **mkdir nome-da-pasta**.

Importante

Atente-se ao caminho em que está criando sua pasta. Nesse caso, ela foi criada dentro de meus usuários. Para acessá-la, vá em **disco local (C:)**, depois em **usuários** e, por fim, em **vstef**.



```
Microsoft Windows [versão 10.0.19042.867]
(c) 2020 Microsoft Corporation. Todos os direitos reservados.

C:\Users\vstef>mkdir angular

C:\Users\vstef>cd angular
```

4. Acesse a pasta criada digitando **cd** e o nome da pasta.

5. Crie um **workspace** para iniciar a aplicação. Para isso, digite: **ng new nome-da-workspace**.

Esse comando abrirá algumas perguntas, mas basta colocar **Y** (yes) para todas e, na opção da estilização, **apertar “enter” na selecionada (CSS)**. Atenção! Esse procedimento pode demorar alguns minutos.

```
C:\Users\vstef\angular>ng new produto
? Do you want to enforce stricter type checking and stricter bundle budgets in the workspace?
  This setting helps improve maintainability and catch bugs ahead of time.
  For more information, see https://angular.io/strict Yes
? Would you like to add Angular routing? Yes
? Which stylesheet format would you like to use? CSS
CREATE produto/angular.json (3631 bytes)
CREATE produto/package.json (1197 bytes)
CREATE produto/README.md (1016 bytes)
CREATE produto/tsconfig.json (783 bytes)
CREATE produto/tslint.json (3185 bytes)
CREATE produto/.editorconfig (274 bytes)
CREATE produto/.gitignore (631 bytes)
CREATE produto/.browserslistrc (703 bytes)
CREATE produto/karma.conf.js (1424 bytes)
CREATE produto/tsconfig.app.json (287 bytes)
CREATE produto/tsconfig.spec.json (333 bytes)
CREATE produto/src/favicon.ico (948 bytes)
CREATE produto/src/index.html (293 bytes)
CREATE produto/src/main.ts (372 bytes)
CREATE produto/src/polyfills.ts (2830 bytes)
CREATE produto/src/styles.css (80 bytes)
CREATE produto/src/test.ts (753 bytes)
CREATE produto/src/assets/.gitkeep (0 bytes)
CREATE produto/src/environments/environment.prod.ts (51 bytes)
CREATE produto/src/environments/environment.ts (662 bytes)
CREATE produto/src/app/app-routing.module.ts (245 bytes)
CREATE produto/src/app/app.module.ts (393 bytes)
CREATE produto/src/app/app.component.html (24846 bytes)
CREATE produto/src/app/app.component.spec.ts (1060 bytes)
CREATE produto/src/app/app.component.ts (211 bytes)
CREATE produto/src/app/app.component.css (0 bytes)
CREATE produto/e2e/protractor.conf.js (904 bytes)
CREATE produto/e2e/tsconfig.json (274 bytes)
CREATE produto/e2e/src/app.e2e-spec.ts (658 bytes)
CREATE produto/e2e/src/app.po.ts (274 bytes)
\ Installing packages (npm)...
```

Após a criação de seu **workspace**, está pronto seu ambiente.

Dica!



Para mais informações, acesse a documentação do Angular: <https://angular.io/guide/setup-local>.

Caso não conheça os **comandos do terminal**, você pode acessar o site do **Dev Content**, que explica alguns comandos:

<https://devcontent.com.br/artigos/windows/principais-comandos-prompt>.