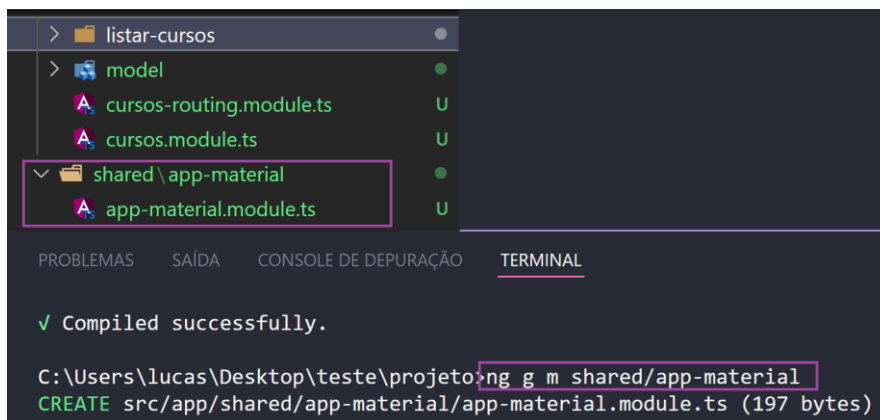




## Imports Compartilhados

**Componentes** tendem a usar os **mesmos imports com frequência**, por exemplo, se usar o Angular Material na sua aplicação provavelmente você vai querer **padronizar todas as telas/componentes** com o mesmo estilo.

Por uma questão de organização podemos criar um módulo para esses “imports” compartilhados (*não é obrigatório*) centralizando-os em um mesmo lugar, tornando o código mais limpo. A criação do módulo não foge muito do padrão basta usar o comando `'ng g m xxxx'`:



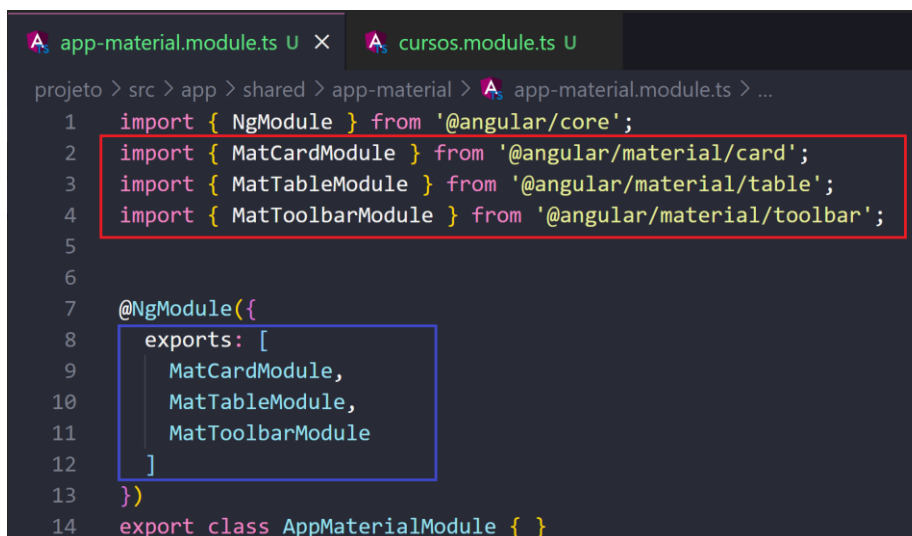
```
> listar-cursos
> model
  cursos-routing.module.ts U
  cursos.module.ts U
  shared\app-material
    app-material.module.ts U

PROBLEMAS  SAÍDA  CONSOLE DE DEPURACÃO  TERMINAL

✓ Compiled successfully.

C:\Users\lucas\Desktop\teste\projeto>ng g m shared/app-material
CREATE src/app/shared/app-material/app-material.module.ts (197 bytes)
```

o que muda na **estrutura do módulo** é que não precisamos de **declarações** (ou seja, esse módulo não tem componentes próprios), apenas de **exports[]** (imports do Angular material que ele vai expor para outros módulos):



```
projeto > src > app > shared > app-material > app-material.module.ts > ...
1  import { NgModule } from '@angular/core';
2  import { MatCardModule } from '@angular/material/card';
3  import { MatTableModule } from '@angular/material/table';
4  import { MatToolbarModule } from '@angular/material/toolbar';
5
6
7  @NgModule({
8    exports: [
9      MatCardModule,
10     MatTableModule,
11     MatToolbarModule
12   ]
13 })
14 export class AppMaterialModule { }
```

Agora basta fazer o import do módulo compartilhado nos módulos que vão utilizar seus imports.