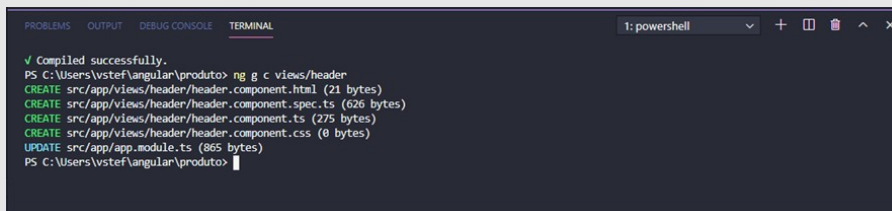


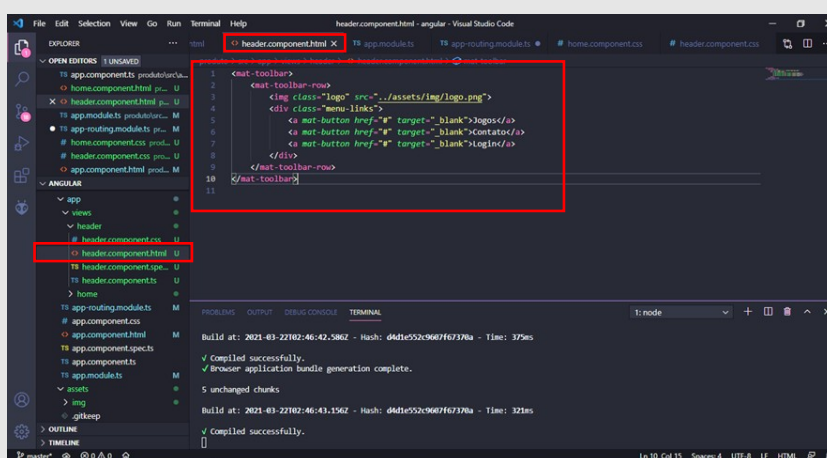
## SEPARANDO HEADER E FOOTER

1. Crie uma pasta chamada **header** dentro de **views**.



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
1: powershell
✓ Compiled successfully.
PS C:\Users\vstef\angular\produto> ng g c views/header
CREATE src/app/views/header/header.component.html (21 bytes)
CREATE src/app/views/header/header.component.spec.ts (626 bytes)
CREATE src/app/views/header/header.component.ts (275 bytes)
CREATE src/app/views/header/header.component.css (0 bytes)
UPDATE src/app/app.module.ts (865 bytes)
PS C:\Users\vstef\angular\produto>
```

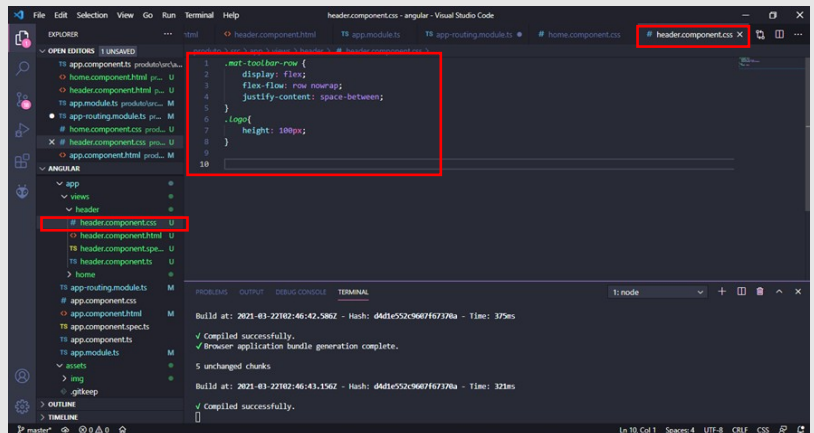
2. Adicione o conteúdo do **header** ao arquivo **header.component.html**.



```
header.component.html - angular - Visual Studio Code
1 <mat-toolbar>
2   <mat-toolbar-row>
3     
4     <div class="menu-links">
5       <a mat-button href="#" target="_blank">logos/a>
6       <a mat-button href="#" target="_blank">contatos/a>
7       <a mat-button href="#" target="_blank">login/a>
8     </div>
9   </mat-toolbar-row>
10 </mat-toolbar>
11
```

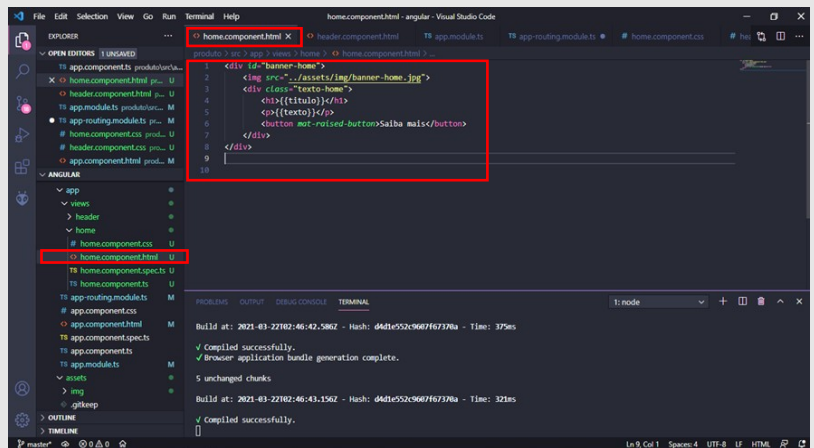
The Explorer sidebar on the left shows the project structure with 'header.component.html' selected under the 'views' folder. The bottom terminal shows successful build and compilation messages.

3. Não se esqueça de copiar a estilização e colá-la em **header.component.css**.



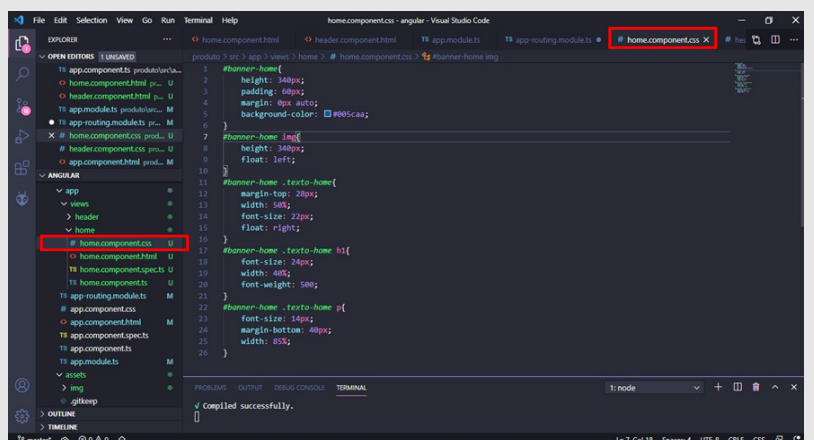
```
1 .mat-toolbar-row {
2   display: flex;
3   flex-flow: row nowrap;
4   justify-content: space-between;
5 }
6 .logo {
7   height: 100px;
8 }
9
```

Sua **home.component.html** ficará assim:



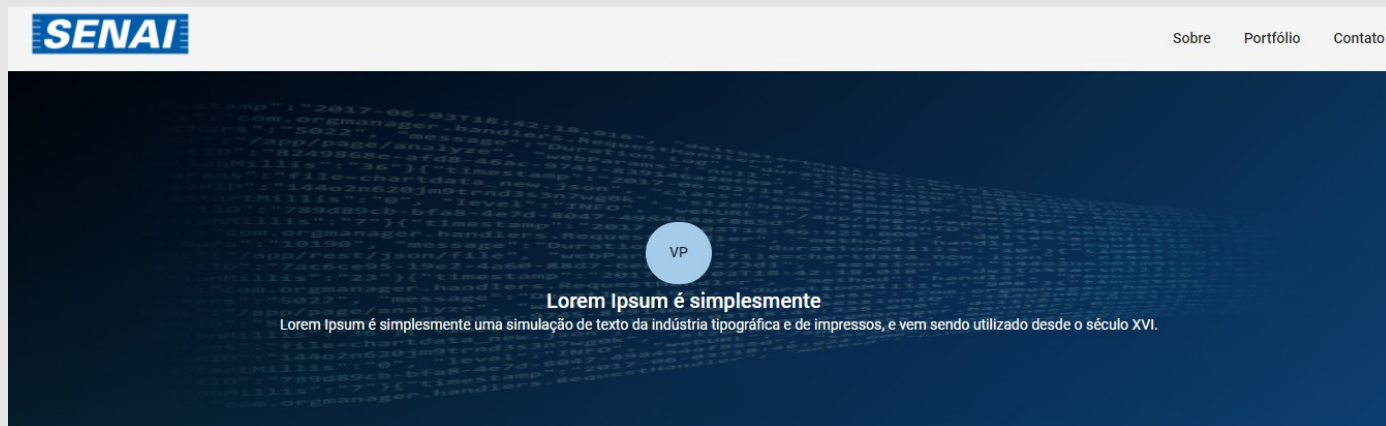
```
1 <div id="banner-home">
2   
3   <div class="texto-home">
4     <h1>[[titulo]]</h1>
5     <p>[[texto]]</p>
6     <button mat-raised-button>Saiba mais</button>
7   </div>
8 </div>
9
```

E o arquivo **home.component.css**, assim:



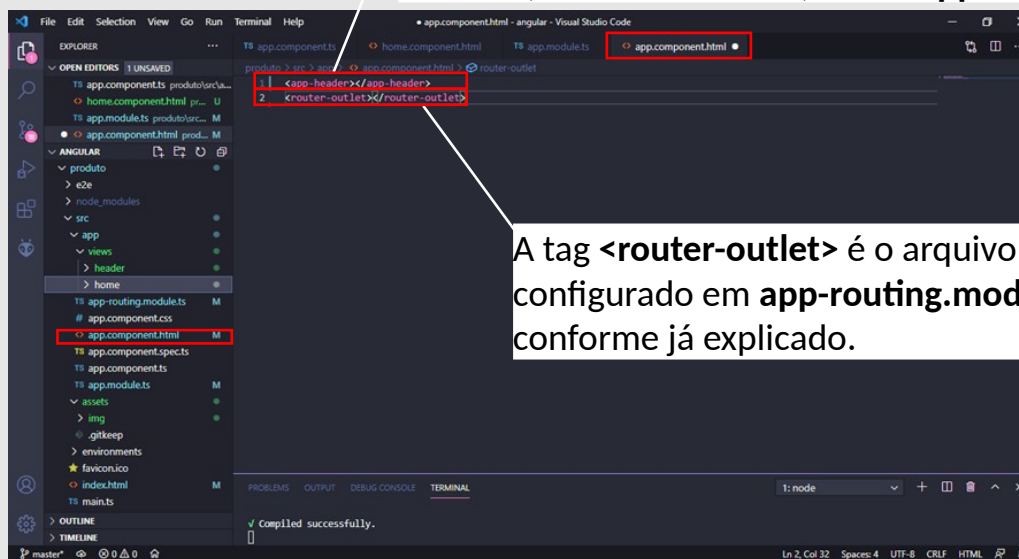
```
1 #banner-home {
2   height: 340px;
3   padding: 40px;
4   margin: 0px auto;
5   background-color: #000ca9;
6 }
7 #banner-home img {
8   height: 340px;
9   float: left;
10 }
11 #banner-home .texto-home {
12   margin-top: 28px;
13   width: 58%;
14   font-size: 22px;
15   float: right;
16 }
17 #banner-home .texto-home h1 {
18   font-size: 24px;
19   width: 48%;
20   font-weight: 500;
21 }
22 #banner-home .texto-home p {
23   font-size: 14px;
24   margin-bottom: 40px;
25   width: 85%;
26 }
```

4. Agora que separamos o HTML e o CSS, precisamos chamar o *header* no template, pois em **app-routing.module.ts** configuramos que, ao iniciar a página, será aberto o arquivo **home.component.html**, ficando assim:



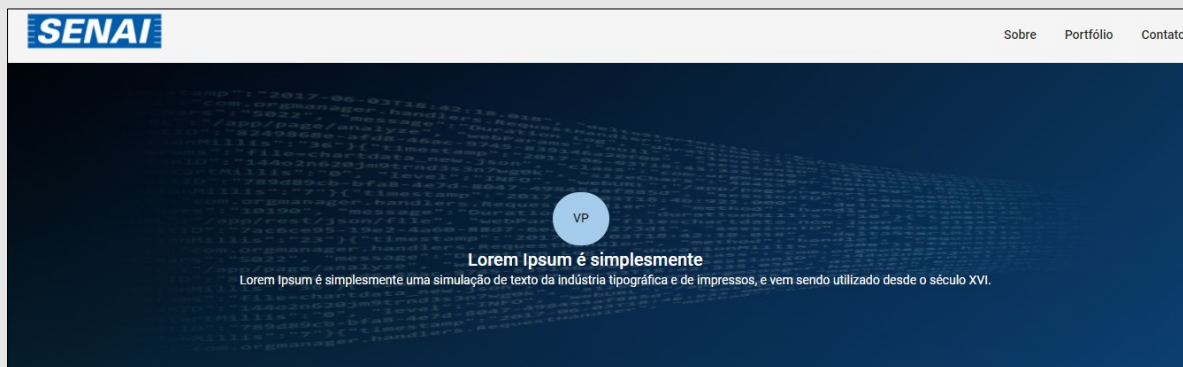
5. Para adicionar a tag **<header>**, acesse o arquivo **app.component.html** e adicione a seguinte linha:

A tag **<app-header>** chamará o *header* para o *template* **<app-header>**, pois o nome da pasta é *header*; caso fosse *footer*, seria **<app-footer>**.



A tag **<router-outlet>** é o arquivo default, configurado em **app-routing.module.ts**, conforme já explicado.

Ao carregar a página, veja que nosso **header** está de volta. Isso fará com que o código não tenha redundância, evitando repetição.



### Dica!

Lembra quando você fazia um código em HTML e tinha que copiar o **<header>** e o **<footer>** iguais para todas as páginas? Fazendo dessa forma, você não precisará repeti-los, basta fazer o procedimento apenas uma única vez. Quando adicionados em **app.component.html**, replicarão a informação para todas as páginas, alterando apenas o conteúdo.

