

Atividade

Site com Angular

Neste conteúdo, você acompanhará como criar o layout de um site com Angular. O site terá a home e a página de login, sem funcionalidades, por enquanto.

The image displays two screenshots of a game store website. The left screenshot shows the homepage ('Produtos') featuring three game controllers (one pink, one blue, one white) against a brick wall background. Below them are two game cards: 'Jogo 1' (R\$ 200,00) and 'Jogo 2' (R\$ 300,00). The right screenshot shows the 'Login' page with fields for 'email@example.com' and 'Senha', and a 'Login' button. Both pages have a dark header with 'Produtos' and 'Login' buttons, and a footer bar at the bottom.

Essa atividade será dividida em 3 partes principais:

- Preparação do ambiente com Angular
- Componentes do Angular Material
- Publicação no GitHub

Preparação do Ambiente

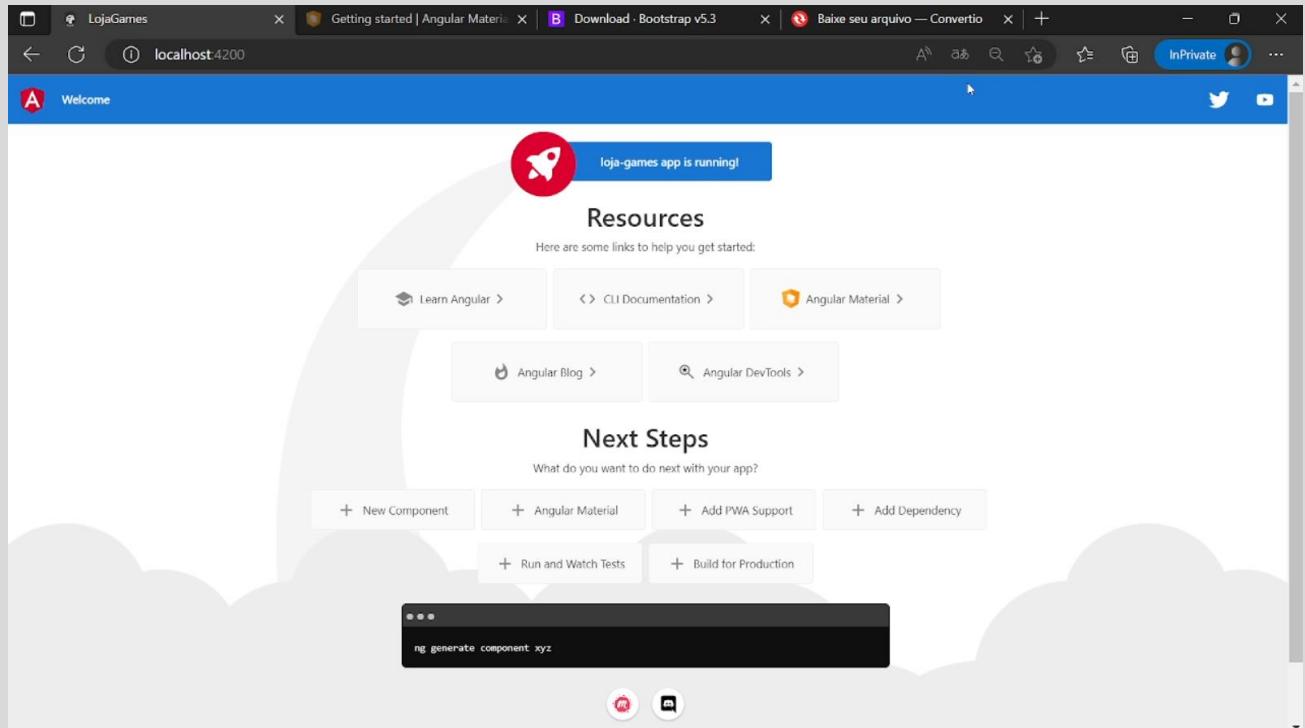
Para essa atividade, você já deve ter instalado em sua máquina o VSCode, o Node.js, o Angular e o Bootstrap.

1. Abra o projeto no VSCode e verifique no arquivo **angular.json** o bloco de código a seguir:

```
"styles": [  
    "node_modules/bootstrap/dist/css/bootstrap.css",  
    "@angular/material/prebuilt-themes/indigo-pink.css",  
    "src/styles.css"  
],  
"scripts": [  
    "node_modules/bootstrap/dist/js/bootstrap.js"  
]
```

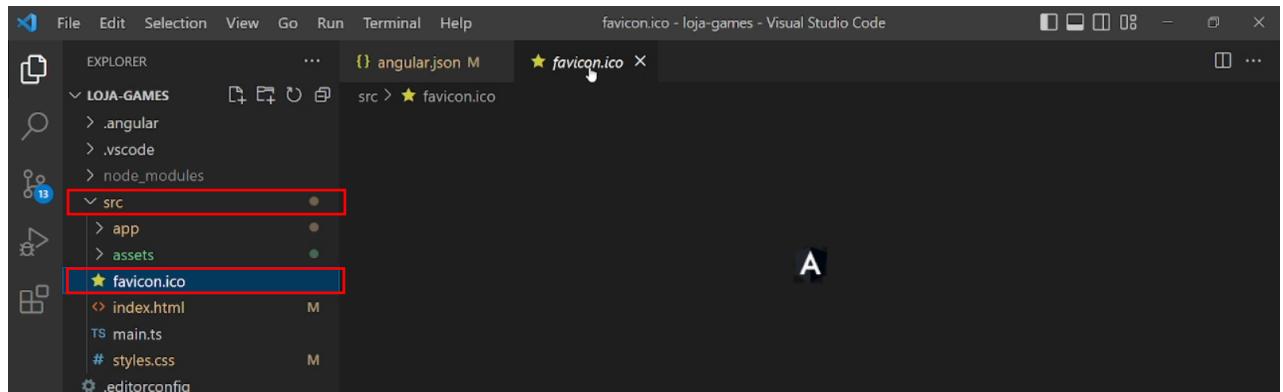
Esse bloco dentro do angular.json significa que o Angular e o Bootstrap foram instalados com sucesso. O Bootstrap precisa do arquivo css e js para funcionar.

2. Verifique o acesso pelo navegador.



Icon

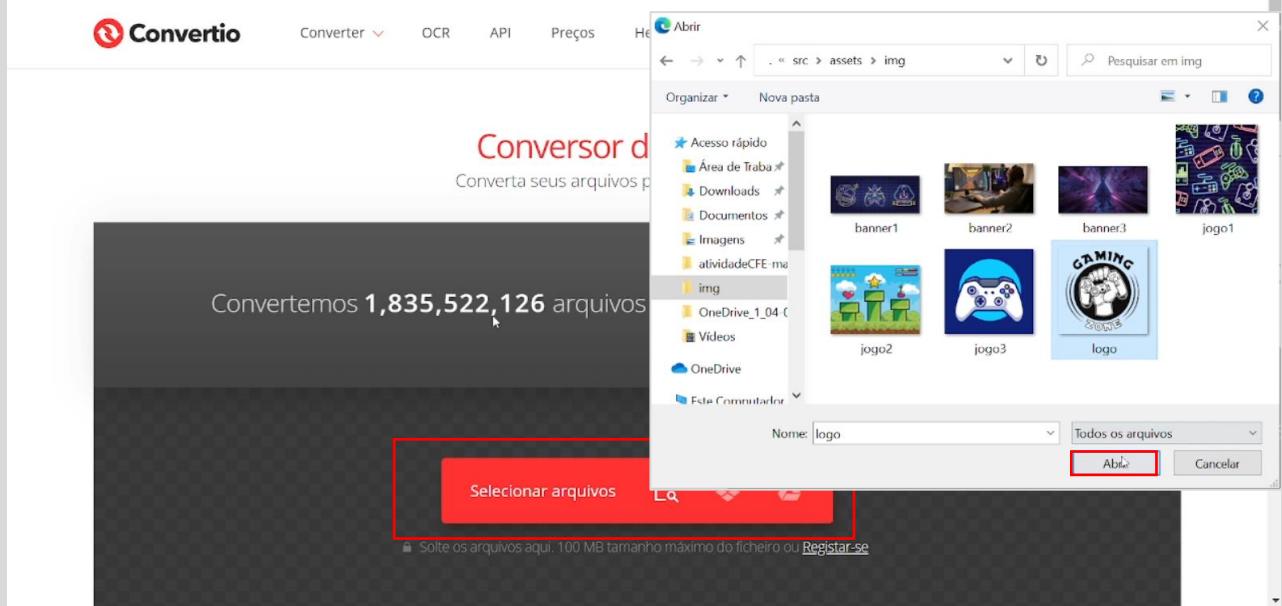
Você deve atualizar a imagem padrão do ícone (**favicon.ico**), que está dentro da pasta **src**.



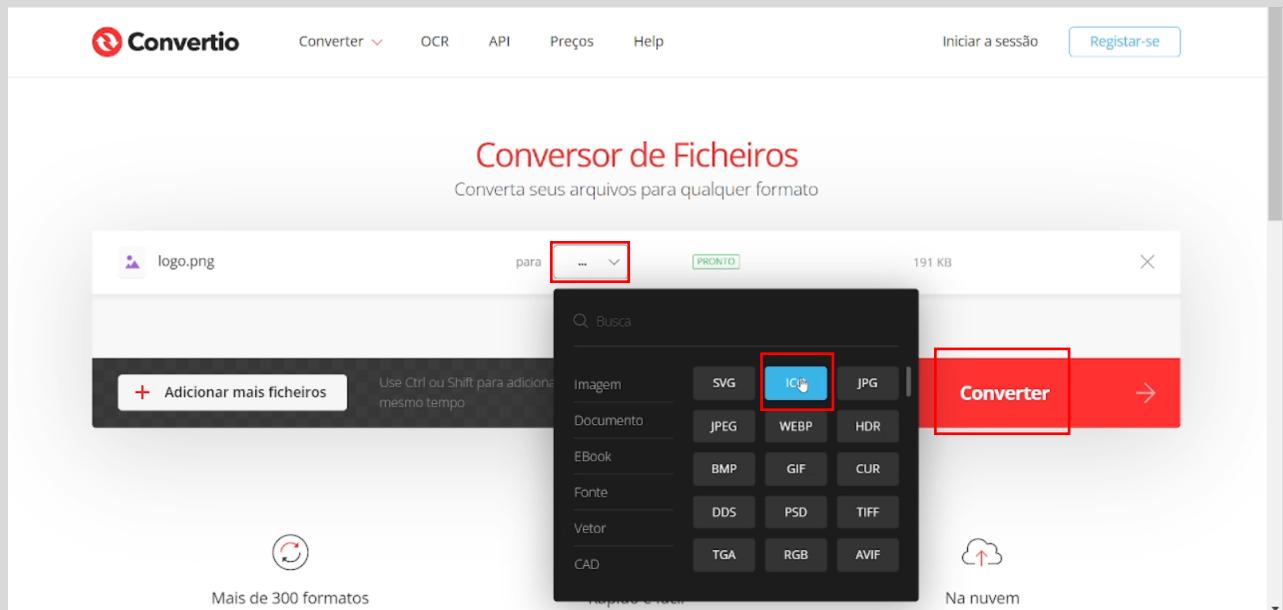
O ícone será a imagem **logo.png**, que está na pasta **src/assets/img**.



1. Acesse o site **convertio.co/pt**, clique em **Selecionar arquivos** e selecione a imagem **logo.png**.



2. Na janela seguinte, clique no **menu dropdown**, selecione a terminação **ico** e depois clique em **Converter**.



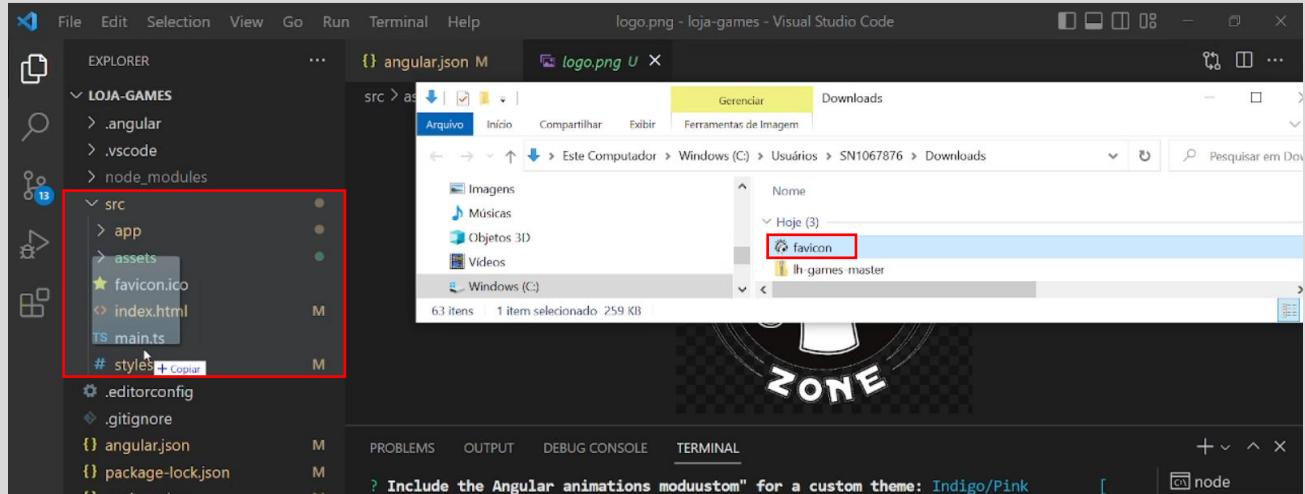
3. Aguarde a conversão ser concluída e clique em **Descarregar**.

The screenshot shows the Convertio website with a success message: "Conversão concluída!". A file named "logo.ico" is listed, marked as "CONCLUIDO" (Completed) with a size of "ICO / 260 KB". A red box highlights the blue "Descarregar" (Download) button. Below the file listing, a note states: "Os arquivos ficarão armazenados por 24 horas. Vá em Meus Arquivos para excluí-los manualmente." At the bottom, there's a link to "Converta mais arquivos" and a rating section with a 4.6 star average from 16,503,649 votes.

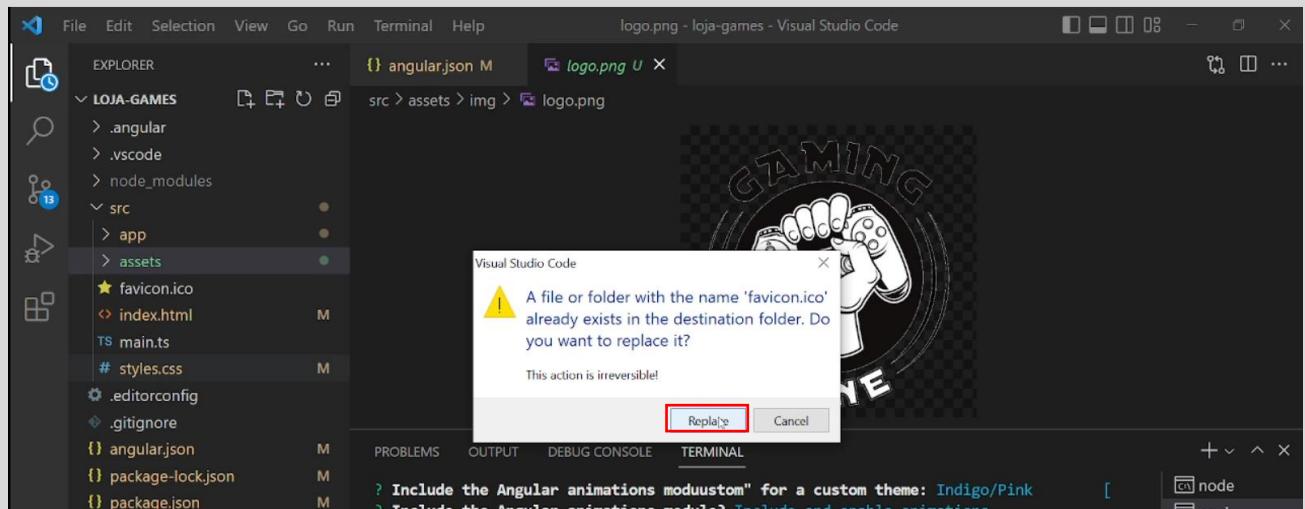
4. Renomeie o arquivo baixado como **favicon.ico**.

The screenshot shows a Windows File Explorer window with the title bar "Downloads". The left sidebar shows "Arquivo", "Início", "Compartilhar", "Exibir", "Gerenciar", and "Ferramentas de Imagem". The address bar shows the path: "Este Computador > Windows (C:) > Usuários > SN1067876 > Downloads". The main area displays a list of files: "Imagens", "Músicas", "Objetos 3D", "Vídeos", and a folder "Windows (C:)". A file named "logo" is selected, and its name is highlighted with a red box. Other visible files include "lh-games-master" and "lh-games-master". At the bottom, it says "63 itens" and "1 item selecionado 259 KB".

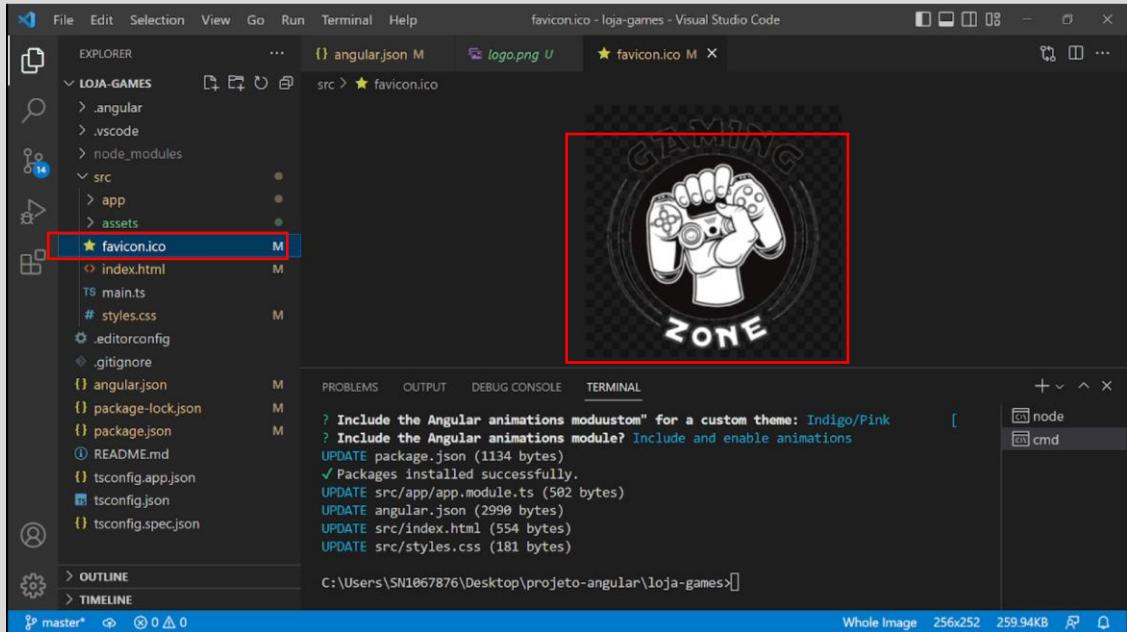
5. Clique e arraste o arquivo renomeado para dentro da pasta **src** do VSCode.



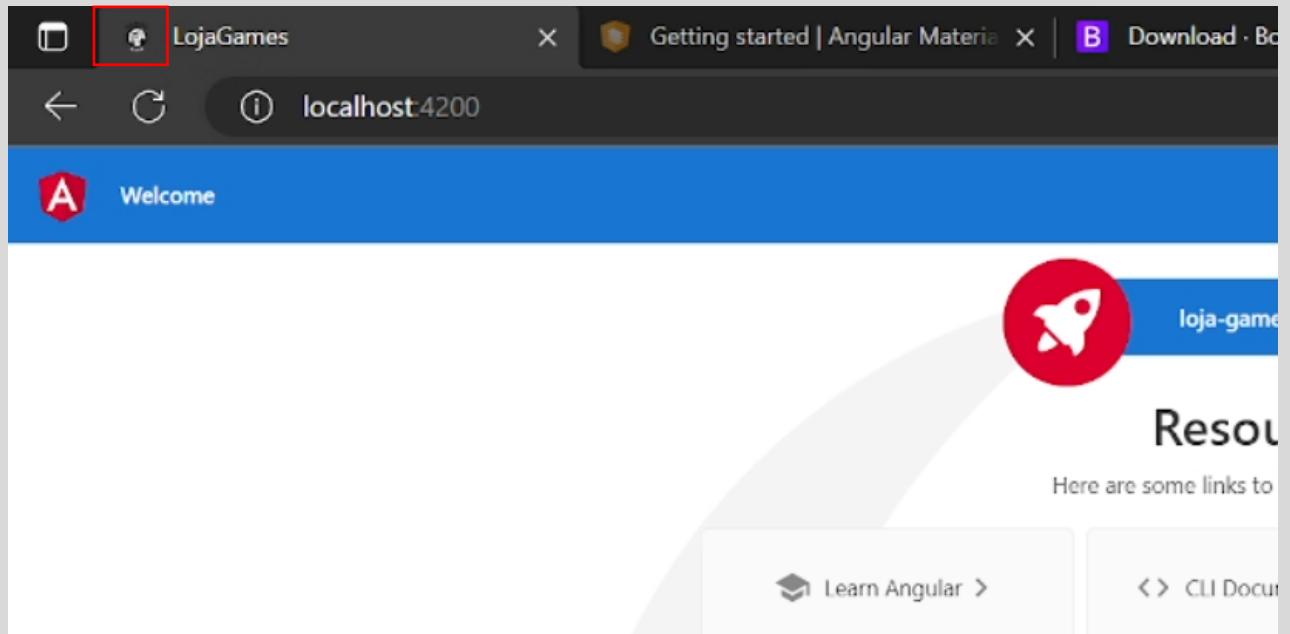
6. Clique em **Replace** dentro da caixa de alerta.



7. Verifique se a imagem do ícone foi alterado corretamente.

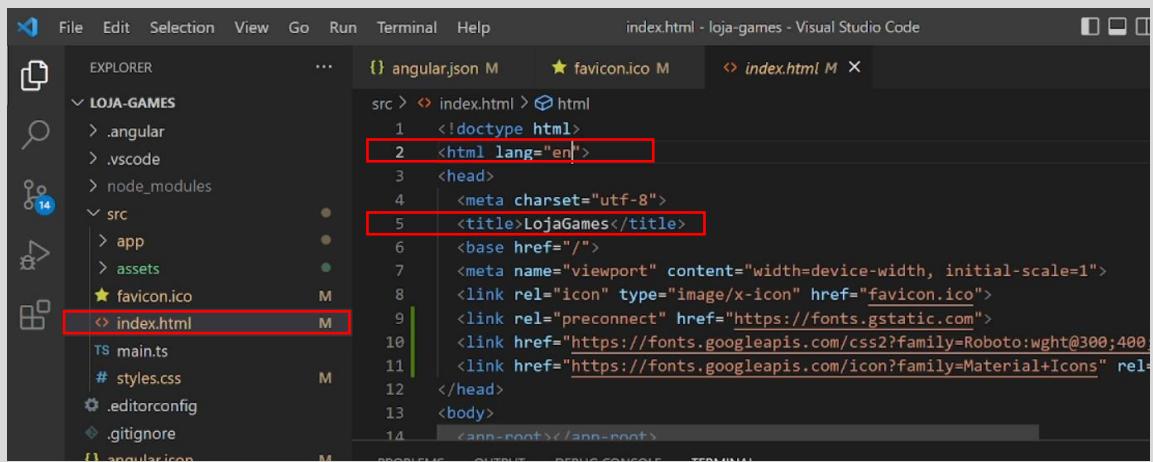


8. Atualize a visualização no navegador para alterar o ícone na aba da janela.



Ajustes antes de iniciar

1. Troque a língua e o título no início da `index.html`. Na segunda linha, altere `<html lang="en">` para `<html lang="pt-br">`. Na linha 5, tire **LojaGames** e digite **LH Games**.

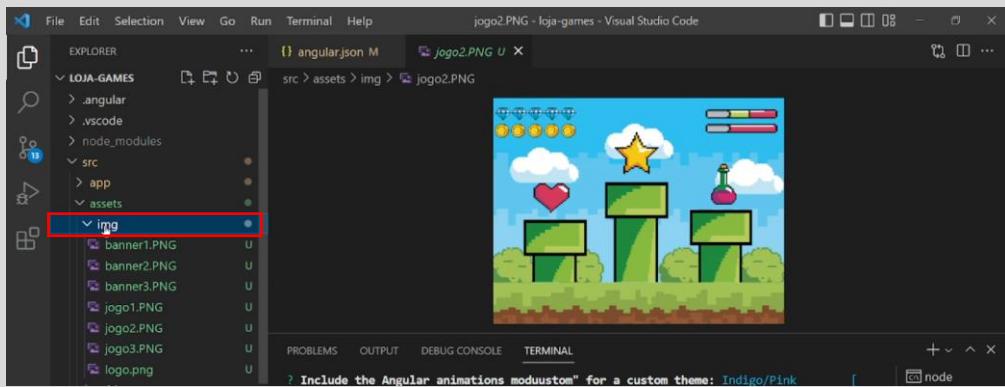


```
File Edit Selection View Go Run Terminal Help index.html - loja-games - Visual Studio Code

EXPLORER          angular.json M    favicon.ico M    index.html M X
LOJA-GAMES
  .angular
  .vscode
  node_modules
  src
    app
    assets
    favicon.ico
    index.html M
  TS main.ts
  # styles.css
  .editorconfig
  .gitignore
  angular.json

src > index.html > html
1  <!doctype html>
2  <html lang="en">
3  |<head>
4  |  <meta charset="utf-8">
5  |  <title>LojaGames</title>
6  |<base href="/">
7  |<meta name="viewport" content="width=device-width, initial-scale=1">
8  |<link rel="icon" type="image/x-icon" href="favicon.ico">
9  |<link rel="preconnect" href="https://fonts.gstatic.com">
10 |<link href="https://fonts.googleapis.com/css?family=Roboto:wght@300;400" rel="stylesheet">
11 |<link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">
12 |</head>
13 |<body>
14 |<app-root></app-root>
```

2. Coloque as imagens na pasta img.

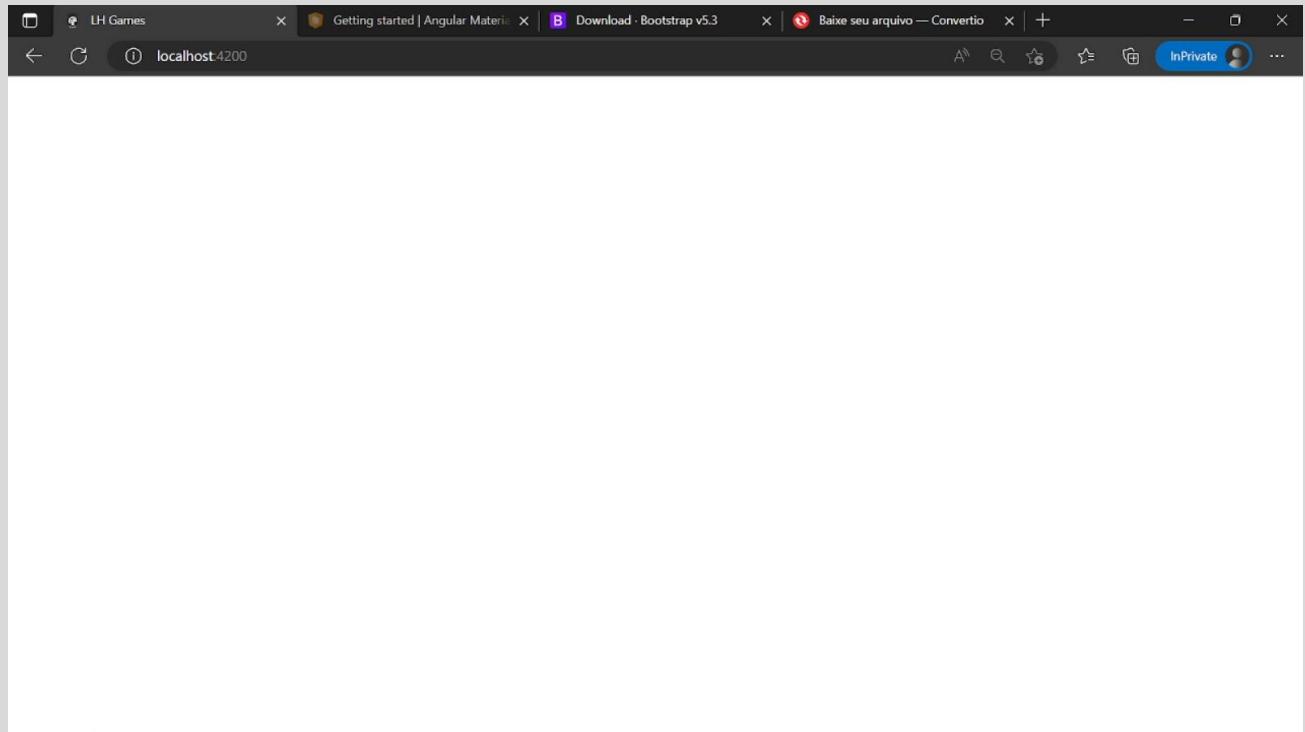


3. Na pasta app, no arquivo **app.componente.html**, selecione e apague todo conteúdo.

4. Na pasta app, no arquivo **app.component.ts**, mude o título alterando na linha 9: **Title = 'LH Games';**

```
File Edit Selection View Go Run Terminal Help • app.component.ts - loja-games - Visual Studio Code
EXPLORER ... jar.json M ★ favicon.ico M ◊ index.html M ◊ app.component.html M TS app.component.ts • □ ...
LOJA-GAMES
src > app > TS app.components.ts > AppComponent > title
1 import { Component } from '@angular/core';
2
3 @Component({
4   selector: 'app-root',
5   templateUrl: './app.component.html',
6   styleUrls: ['./app.component.css']
7 })
8 export class AppComponent {
9   title = 'LH Games';
10 }
11
```

5. Verifique o projeto no navegador.



Componentes do Angular Material

Nessa etapa, você deve escolher quais componentes do Angular Material usar, fazer a importação de cada um deles, depois copiar o código html e css de cada um, e, depois, fazer os devidos ajustes.

Os componentes que serão importados são:

- button
- card
- form field
- icon
- input
- menu
- toolbar

1. Acesse o endereço <https://material.angular.io/> e clique em Components.



2. Encontre o componente no lado esquerdo (**button**, por exemplo), clique em **API** e selecione a **referência** para copiar.

The screenshot shows the Angular Material Components documentation. The sidebar on the left lists various components: Autocomplete, Badge, Bottom Sheet, **Button** (which is highlighted with a red box), Button toggle, Card, Checkbox, Chips, Core, Datepicker, and Dialog. The main content area has tabs for OVERVIEW, **API** (which is highlighted with a red box), and EXAMPLES. The API tab displays the 'API reference for Angular Material button'. It includes the import statement: `import {MatButtonModule} from '@angular/material/button';`. To the right, there is a sidebar with a tree view of the Button component's structure, including MatButton, Directives, MatAnchor, MatFabButton, and others.

3. Dentro da pasta app, no arquivo **app.module.ts**, abra um comentário para marcar as importações do Angular Material (**/*Importações do Angular Material*/**) e cole a linha copiada abaixo do comentário.

```
/*Importações Angular Material*/
import {MatButtonModule} from '@angular/material/button';
```

The screenshot shows the Visual Studio Code interface with the file 'app.module.ts' open. The code editor shows the following content:

```
/*Importações Angular Material*/
import {MatButtonModule} from '@angular/material/button';

@NgModule({
  declarations: [
    AppComponent
  ],
  imports: [
    BrowserModule,
    AppRoutingModule,
    BrowserAnimationsModule,
    BrowserAnimationsModule,
    MatButtonModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
```

The line `import {MatButtonModule} from '@angular/material/button';` is highlighted with a red box. The Explorer sidebar on the left shows the project structure with files like 'index.html', 'app-routing.module.ts', 'app.component.css', etc. The bottom status bar indicates the file is saved in 'master' branch.

4. Ainda no arquivo **app.module.ts**, encontre o bloco de imports e adicione o módulo importado (**MatButtonModule**, no exemplo).

```
src > app > ts app.module.ts > AppModule
1
2  /* Importações Angular Material*/
3  import {MatButtonModule} from '@angular/material/button';
4
5
6
7
8  import { AppRoutingModule } from './app-routing.module';
9  import { AppComponent } from './app.component';
10 | import { BrowserAnimationsModule } from '@angular/platform-browser/animations';
11
12 @NgModule({
13   declarations: [
14     AppComponent
15   ],
16   imports: [
17     BrowserModule,
18     AppRoutingModule,
19     BrowserAnimationsModule,
20 |     MatButtonModule
21   ],
22   providers: [],
23   bootstrap: [AppComponent]
```

5. Repita os passos para cada componentes escolhido até que o arquivo **app.module.ts** contenha os seguintes blocos:

```
/* Importações Angular Material*/
import {MatButtonModule} from '@angular/material/button';
import {MatCardModule} from '@angular/material/card';
import {MatFormFieldModule} from '@angular/material/form-field';
import {MatGridListModule} from '@angular/material/grid-list';
import {MatIconModule} from '@angular/material/icon';
import {MatInputModule} from '@angular/material/input';
import {MatMenuModule} from '@angular/material/menu';
import {MatToolbarModule} from '@angular/material/toolbar';
```

```
imports: [
  MatButtonModule,
  MatCardModule,
  MatFormFieldModule,
  MatGridListModule,
  MatIconModule,
  MatInputModule,
  MatMenuModule,
  MatToolbarModule
],
```

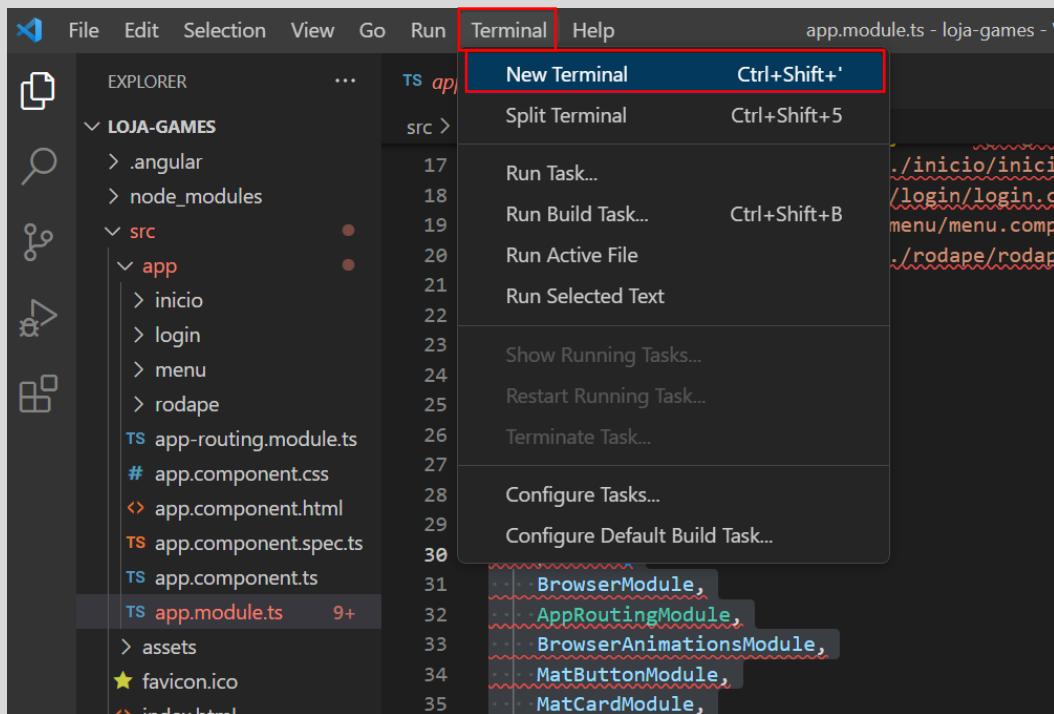
Componentes do Site

Agora você vai criar os componentes das páginas do site:

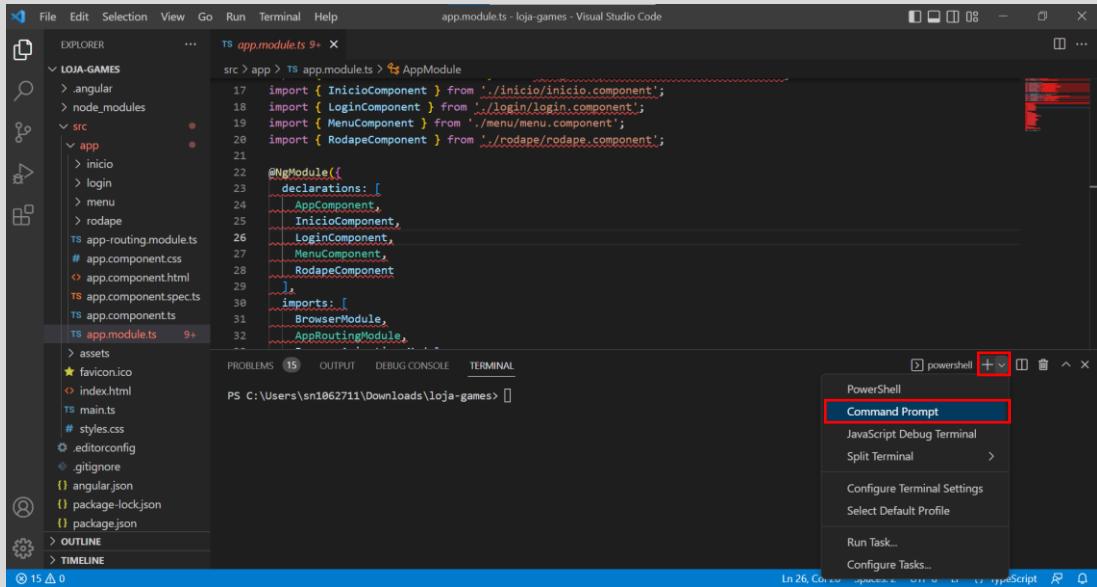
- inicio
- login
- menu
- rodapé

Dentro dos componentes do site, você vai inserir os componentes do Angular Material da etapa anterior.

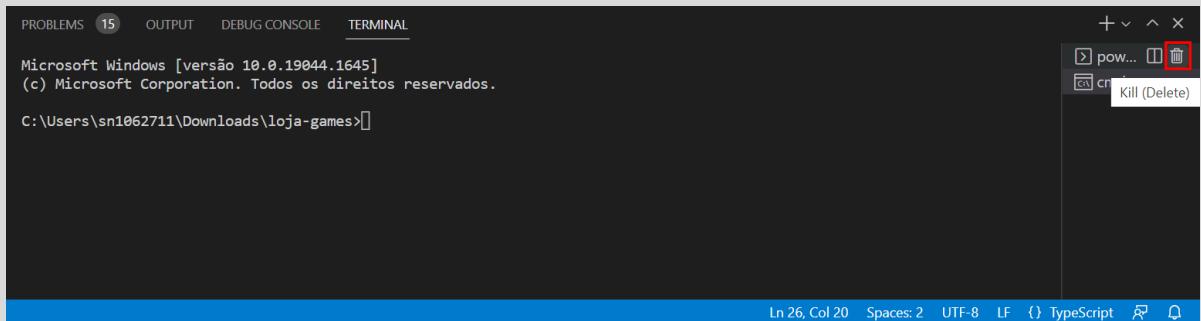
1. Abra o terminal do VSCode clicando no menu superior em **Terminal/New Terminal** ou usando o atalho **Ctrl+Shift+'**



2. Antes de criar os componentes, verifique o tipo de terminal.
Clique no ícone de + e selecione o **Prompt de comando**.



3. Apague os demais prompts, clicando no ícone de lata de lixo.



4. No prompt de comando, digite **ng generate component inicio** e dê **Enter**.

The screenshot shows the Visual Studio Code interface. The terminal tab is active, displaying the command `C:\Users\SN1067876\Desktop\projeto-angular\loja-games>ng generate component inicio`. The code editor shows the `app.module.ts` file with the following code:

```

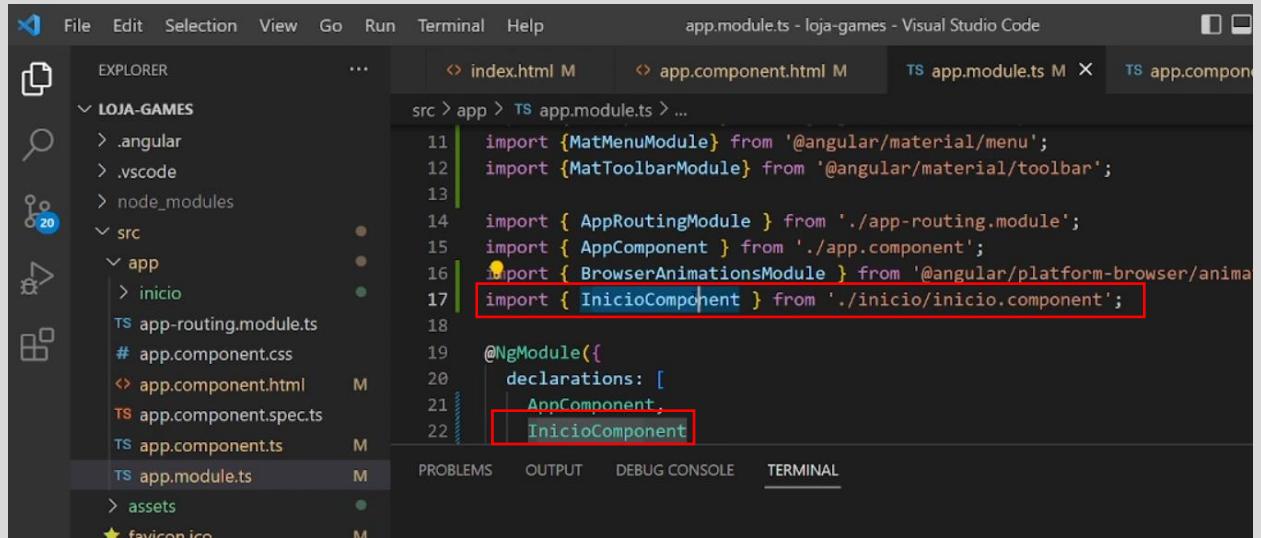
src > app > TS app.module.ts > AppModule
  declarations: [
    AppComponent
  ],
  imports: [
    BrowserModule,
    AppRoutingModule,
    BrowserAnimationsModule,
    MatButtonModule,
    MatCardModule,
    MatFormFieldModule,
    MatGridListModule,
    MatIconModule,
    MatInputModule
  ]

```

5. Aguarde a criação da pasta inicio e seus respectivos arquivos, dentro da pasta app.

The screenshot shows the Visual Studio Code interface after the command has been run. The Explorer sidebar now includes a new folder named `inicio`, which contains four files: `inicio.component.css`, `inicio.component.html`, `inicio.component.spec.ts`, and `inicio.component.ts`. The code editor still shows the `app.module.ts` file with the same code as before. The terminal shows the output of the command: `C:\Users\SN1067876\Desktop\projeto-angular\loja-games>ng generate component inicio` followed by `CREATE src/app/inicio/inicio_component.html (21 bytes)`.

6. No arquivo **app.module.ts**, a importação e a declaração do **InícioComponent** foram criadas automaticamente.



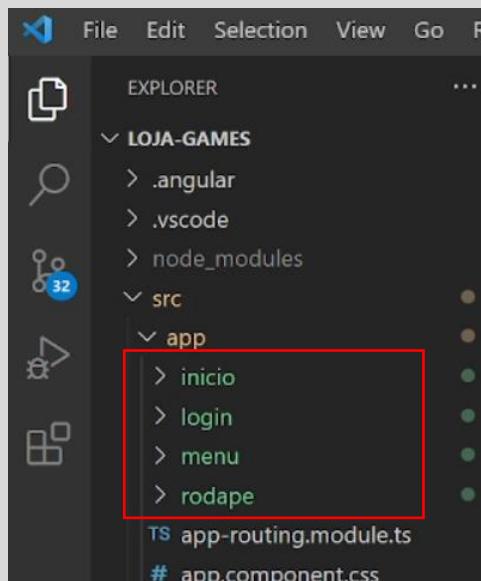
```

File Edit Selection View Go Run Terminal Help
app.module.ts - loja-games - Visual Studio Code
EXPLORER index.html M app.component.html M TS app.module.ts M X TS app.compon...
src > app > TS app.module.ts ...
11 import { MatMenuModule } from '@angular/material/menu';
12 import { MatToolbarModule } from '@angular/material/toolbar';
13
14 import { AppRoutingModule } from './app-routing.module';
15 import { AppComponent } from './app.component';
16 import { BrowserAnimationsModule } from '@angular/platform-browser/animations';
17 import { InicioComponent } from './inicio/inicio.component';

18 @NgModule({
19   declarations: [
20     AppComponent,
21     InicioComponent
22 ]

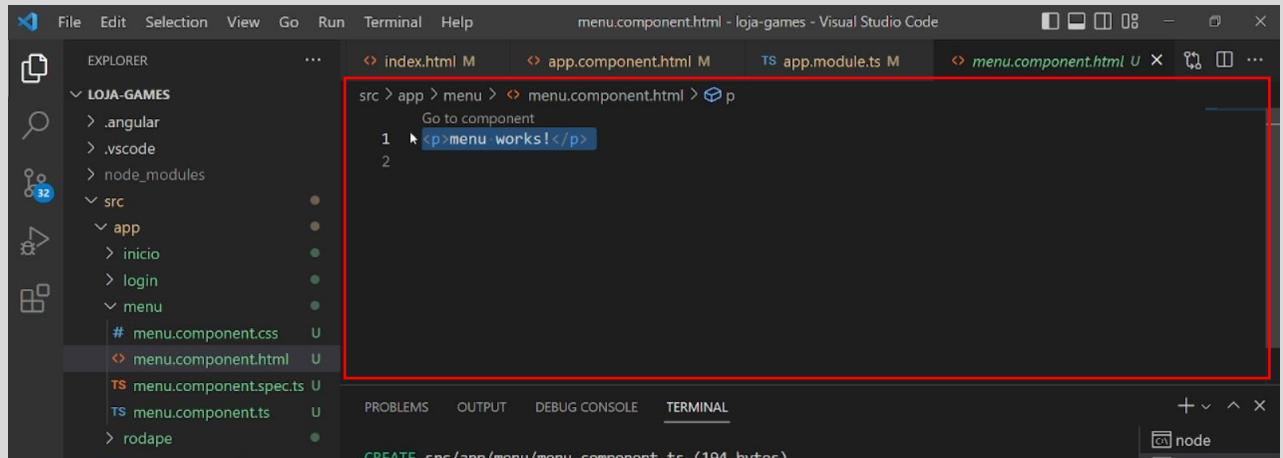
```

Agora, crie os demais componentes: login, menu e rodapé. Digite **ng g c login**, dê **Enter** e aguarde a criação do componente. Digite **ng g c menu**, dê **Enter** e aguarde. Digite **ng g c rodape**, dê **Enter** e aguarde. As pastas dos componentes devem aparecer do lado esquerdo, dentro da pasta app.



Menu

1. Selecione todo o conteúdo do arquivo **menu.componente.html** e delete

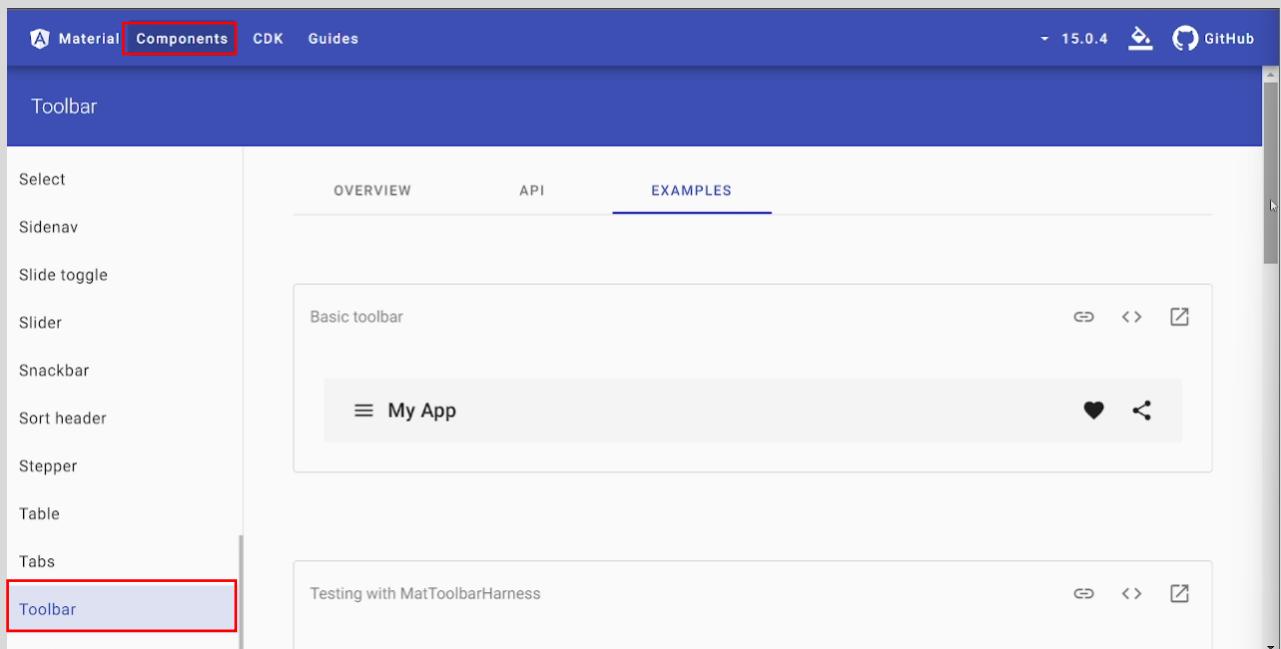


A screenshot of Visual Studio Code showing the file structure of a project named "LOJA-GAMES". The "src" folder contains "app", "menu", and "rodape". Inside "menu", there are files for "menu.component.css", "menu.component.html", "menu.component.spec.ts", and "menu.component.ts". The "menu.component.html" file is open in the editor, displaying the code:

```
src > app > menu > menu.component.html > p
Go to component
1 <p>menu works!</p>
2
```

 The line `<p>menu works!</p>` is highlighted with a red box.

2. Acesse o Angular Material e clique em **Toolbar**, dentro de Components.



A screenshot of the Angular Material Components documentation page. The top navigation bar has "Components" selected, which is highlighted with a red box. The left sidebar lists components: Select, Sidenav, Slide toggle, Slider, Snackbar, Sort header, Stepper, Table, Tabs, and Toolbar. The "Toolbar" item is also highlighted with a red box. The main content area shows examples for the Toolbar component, including a "Basic toolbar" example with a "My App" title and a "Testing with MatToolbarHarness" example. The "EXAMPLES" tab is active.

3. Encontre o último exemplo de toolbar e clique no ícone <>.

The screenshot shows the Angular Material Components documentation. The left sidebar has 'Toolbar' selected. The main content area displays a 'Toolbar overview' example titled 'My Application'. This example consists of four stacked toolbar components. The third toolbar contains the text 'My App'. The fourth toolbar contains the text 'Second Line'. A red box highlights the third toolbar. Another red box highlights the copy icon (<>) in the top right corner of the preview area.

4. Clique em **HTML**, selecione o terceiro <mat-toolbar> e copie.

The screenshot shows the Angular Material Components documentation. The left sidebar has 'Toolbar' selected. The main content area displays a 'Toolbar overview' example. Below the toolbar examples, there is a code editor tab labeled 'HTML' which is highlighted with a red box. The code editor contains the HTML template for the toolbar examples. A red box highlights the third `<mat-toolbar>` element in the code, which corresponds to the third toolbar shown in the previous screenshot.

```

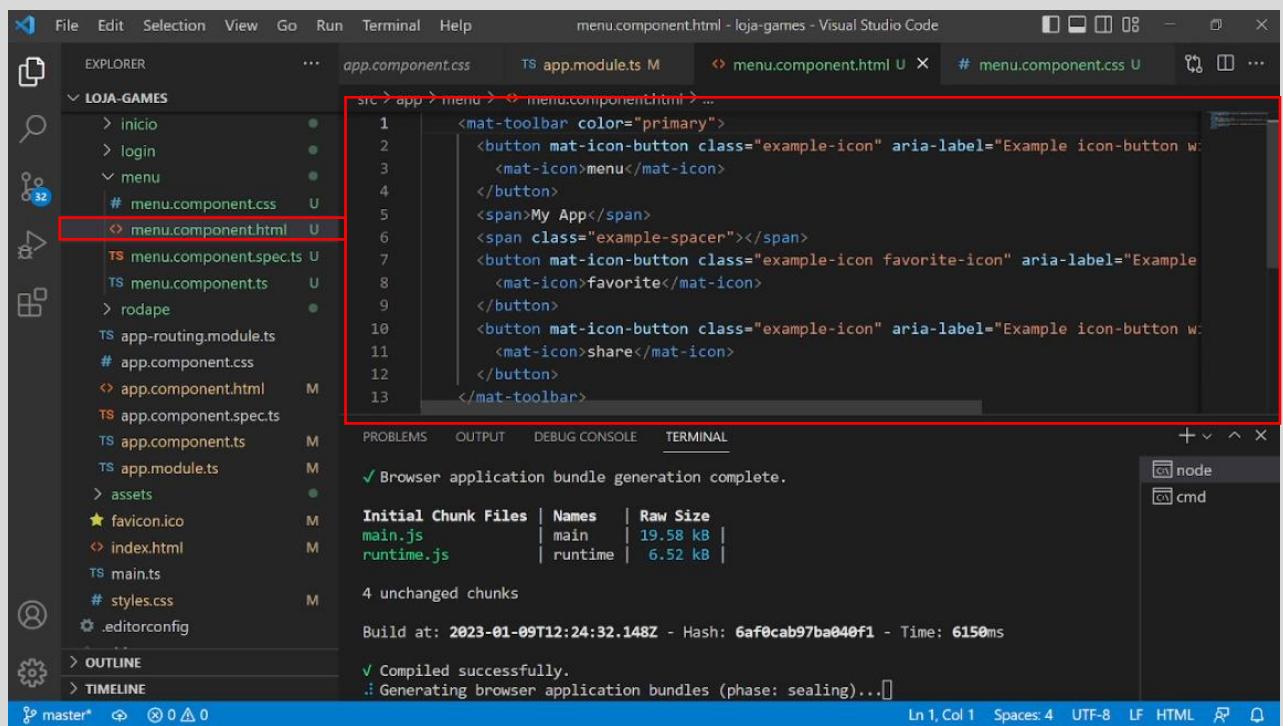
<p>
  <mat-toolbar>
    <span>My Application</span>
  </mat-toolbar>
</p>

<p>
  <mat-toolbar>
    <button mat-icon-button class="example-icon" aria-label="Example icon-button with menu icon">
      <mat-icon>menu</mat-icon>
    </button>
    <span>My App</span>
    <span class="example-spacer"></span>
    <button mat-icon-button class="example-icon favorite-icon" aria-label="Example icon-button with heart icon">
      <mat-icon>favorite</mat-icon>
    </button>
    <button mat-icon-button class="example-icon" aria-label="Example icon-button with share icon">
      <mat-icon>share</mat-icon>
    </button>
  </mat-toolbar>
</p>

```

5. Cole o código do terceiro <mat-toolbar> dentro do arquivo menu.componente.html

```
<mat-toolbar color="primary">
  <button mat-icon-button class="example-icon" aria-label="Example icon-button with menu icon">
    <mat-icon>menu</mat-icon>
  </button>
  <span>My App</span>
  <span class="example-spacer"></span>
  <button mat-icon-button class="example-icon favorite-icon" aria-label="Example icon-button with heart icon">
    <mat-icon>favorite</mat-icon>
  </button>
  <button mat-icon-button class="example-icon" aria-label="Example icon-button with share icon">
    <mat-icon>share</mat-icon>
  </button>
</mat-toolbar>
```



6. Delete o **primeiro botão** (código a seguir):

```
<button mat-icon-button class="example-icon" aria-label="Example icon-button with menu icon">
    <mat-icon>menu</mat-icon>
</button>
```

E insira a **imagem do logo** no lugar, digitando o código a seguir:

```
<span></span>
```

7. Troque o título **My App** por **Produtos** e delete os **botões depois de example-spacer**, tirando o código a seguir:

```
<button mat-icon-button class="example-icon favorite-icon" aria-label="Example icon-button with heart icon">
    <mat-icon>favorite</mat-icon>
</button>
<button mat-icon-button class="example-icon" aria-label="Example icon-button with share icon">
    <mat-icon>share</mat-icon>
</button>
```

E insira um texto de **login**, digitando no mesmo lugar o código a seguir:

```
<span>Login</span>
```

8. Retorne ao Angular Material e clique em **CSS**. Selecione todo o css e copie.

The screenshot shows the Angular Material documentation interface. On the left, there's a sidebar with various components listed: Slide toggle, Slider, Snackbar, Sort header, Stepper, Table, Tabs, **Toolbar**, and Tooltip. The **Toolbar** item is highlighted with a light blue background. The main content area displays the 'Toolbar overview' page. At the top of this page, there are tabs for HTML, TS, and CSS. The CSS tab is selected and highlighted with a red box. Below the tabs, there is some sample CSS code:

```
.example-spacer { flex: 1 1 auto; }
```

 This code is also enclosed in a red box. The overall URL in the browser bar is `https://material.angular.io/components/toolbar/overview`.

9. Cole o css copiado em **menu.componentes.css**

```
.example-spacer {
  flex: 1 1 auto;
}
```

The screenshot shows a Visual Studio Code window with the file `# menu.component.css` open. The code editor contains the following CSS rule:

```
src > app > menu > # menu.component.css > .example-spacer
1 .example-spacer [
2   flex: 1 1 auto;
3 ]
```

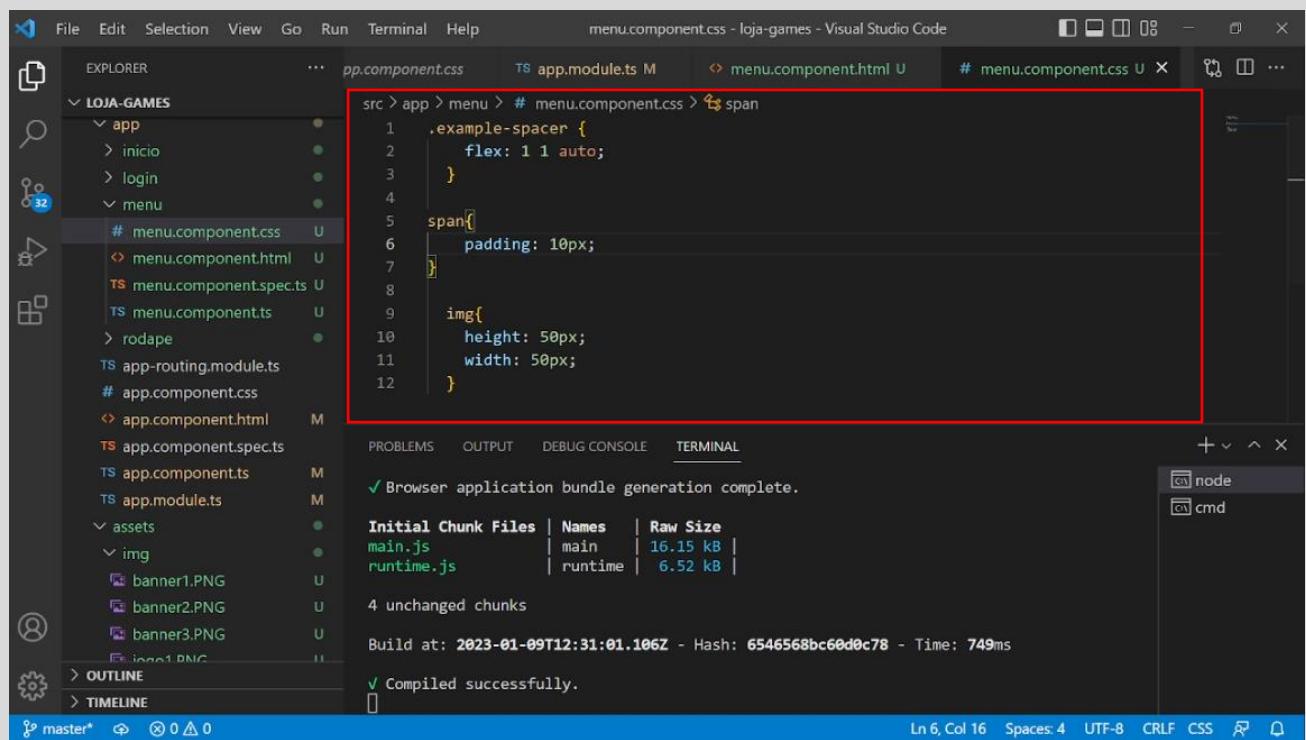
A red box highlights the entire CSS rule. The Explorer sidebar on the left shows the project structure with a folder named `LOJA-GAMES` containing files like `.angular`, `.vscode`, and `node_modules`. Inside `src/app`, there are `inicio`, `login`, and `menu` components. The `menu.component.css` file is currently selected.

O que estiver antes do elemento **.example-spacer** no html ficará à esquerda na tela. O que estiver depois do **.example-spacer** ficará à esquerda.

10. Acrescente no css o **tamanho da imagem** e um **espacamento** entre os elementos, digitando o seguinte código no arquivo menu.componente.css

```
span{  
    padding: 10px;  
}
```

```
img{  
    height: 50px;  
    width: 50px;  
}
```



The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure under "LOJA-GAMES".
- Code Editor:** Displays the CSS code for "menu.component.css". A red box highlights the entire code block.
- Terminal:** Shows the build process output:
 - ✓ Browser application bundle generation complete.
 - Initial Chunk Files | Names | Raw Size

main.js	main	16.15 KB
runtime.js	runtime	6.52 KB

 - 4 unchanged chunks
 - Build at: 2023-01-09T12:31:01.106Z - Hash: 6546568bc60d0c78 - Time: 749ms
 - ✓ Compiled successfully.
- Status Bar:** Shows "Ln 6, Col 16" and other file statistics.

11. Retorne ao Angular Material e clique em **TS**. Verifique se há códigos a serem copiados entre a linha do **import** e do **@Component**. Verifique também se há métodos a serem copiados entre as **chaves da última linha**.

```

import {Component} from '@angular/core';

/**
 * @title Toolbar overview
 */
@Component({
  selector: 'toolbar-overview-example',
  templateUrl: 'toolbar-overview-example.html',
  styleUrls: ['./toolbar-overview-example.css'],
})
export class ToolbarOverviewExample {}

```

Note que a estrutura está parecida com o arquivo **menu.componentes.ts**. Não há ajustes a serem feitos.

```

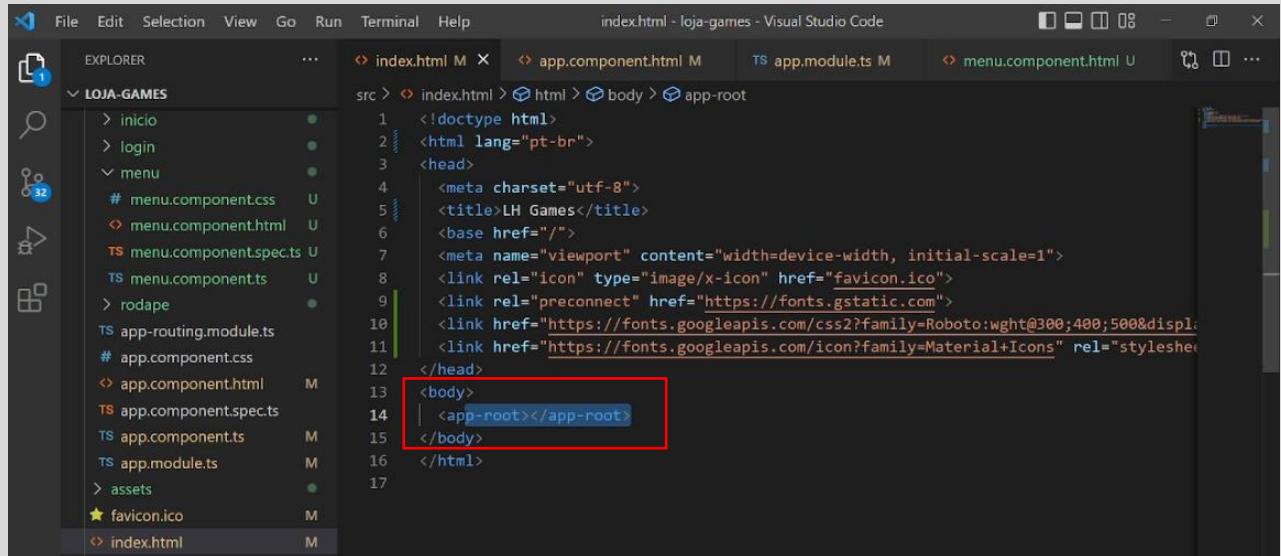
import { Component } from '@angular/core';

@Component({
  selector: 'app-menu',
  templateUrl: './menu.component.html',
  styleUrls: ['./menu.component.css']
})
export class MenuComponent {
}

```

Visualização no navegador

- O **index.html** está chamando o app root com o trecho
<body><app-root></app-root></body>.

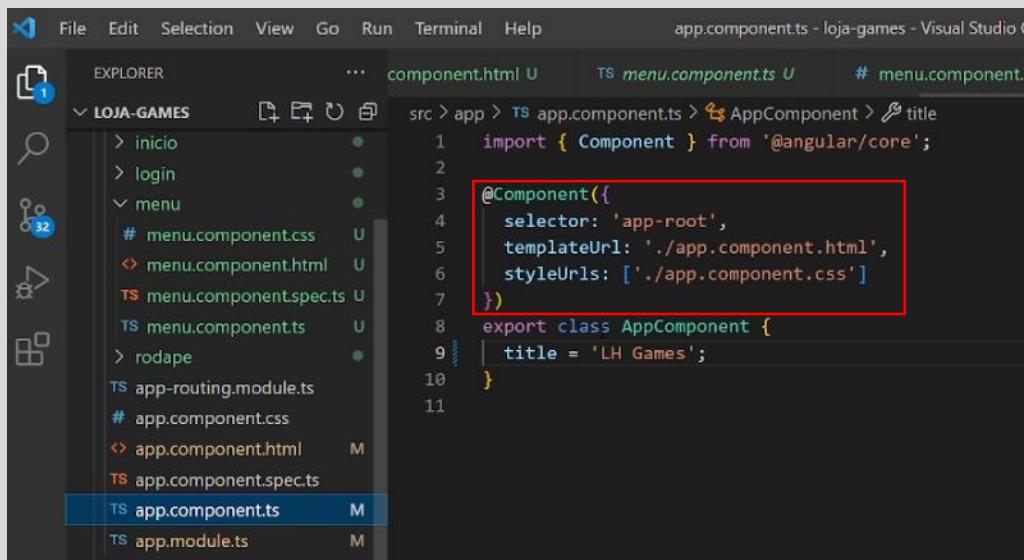


```

File Edit Selection View Go Run Terminal Help index.html - loja-games - Visual Studio Code
EXPLORER index.html M X app.component.html M app.module.ts menu.component.html U
src > index.html > html > body > app-root
1  <!doctype html>
2  <html lang="pt-br">
3  <head>
4    <meta charset="utf-8">
5    <title>LH Games</title>
6    <base href="/">
7    <meta name="viewport" content="width=device-width, initial-scale=1">
8    <link rel="icon" type="image/x-icon" href="favicon.ico">
9    <link rel="preconnect" href="https://fonts.gstatic.com">
10   <link href="https://fonts.googleapis.com/css2?family=Roboto:wght@300;400;500&display=block">
11   <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">
12 </head>
13 <body>
14 | <app-root></app-root>
15 </body>
16 </html>
17

```

- No arquivo **app.componentes.ts**, o **app root** puxa o arquivo **app.componente.html** e o **app.componente.css**

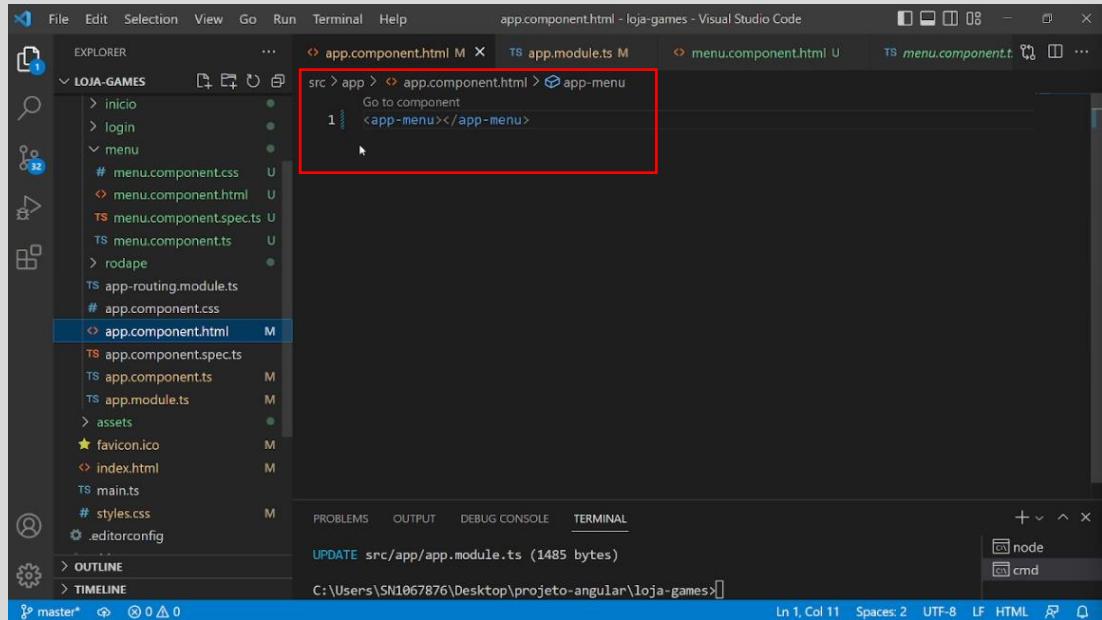


```

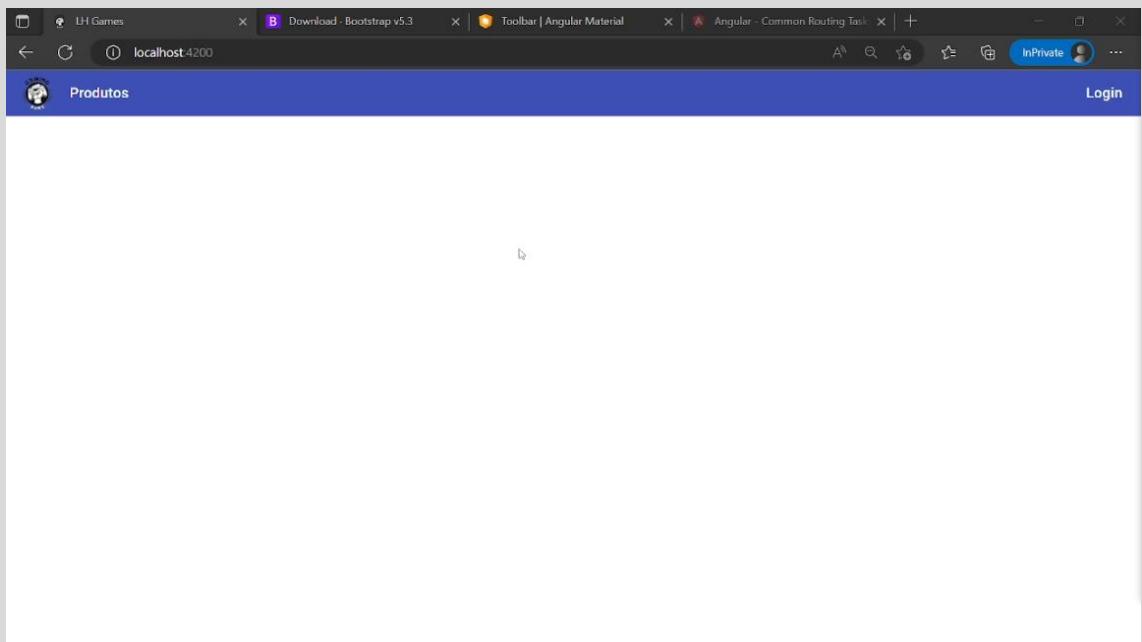
File Edit Selection View Go Run Terminal Help app.component.ts - loja-games - Visual Studio Code
EXPLORER component.html U menu.component.ts # menu.component.html
src > app > app.component.ts > AppComponent > title
1 import { Component } from '@angular/core';
2
3 @Component({
4   selector: 'app-root',
5   templateUrl: './app.component.html',
6   styleUrls: ['./app.component.css']
7 })
8 export class AppComponent {
9   title = 'LH Games';
10 }
11

```

3. No arquivo **app.component.html**, digite **<app-menu></app-menu>** e salve as alterações.



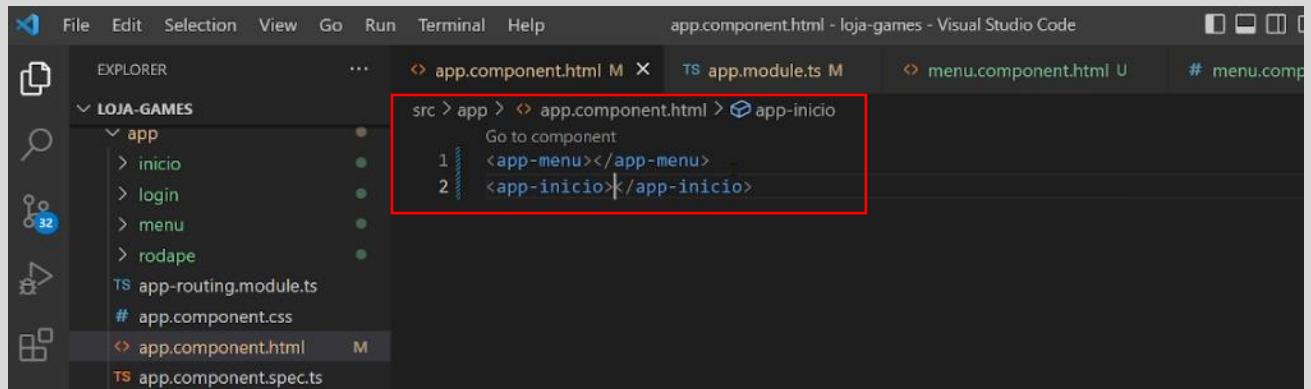
4. Atualize o site no navegador para verificar as alterações.



Início

1. No arquivo **app.component.html**, digite o seguinte código após `<app-menu></app-menu>`:

```
<app-inicio></app-inicio>
```



2. Acesse <https://getbootstrap.com/> e clique em **Docs**

New in v5.3 Color mode support, expanded color palette, and more!

Build fast, responsive sites with Bootstrap

Powerful, extensible, and feature-packed frontend toolkit.
Build and customize with Sass, utilize prebuilt grid system

3. Desça a barra de rolagem do lado esquerdo até encontrar Components. Clique em Carousel.

The screenshot shows the Bootstrap documentation interface. The top navigation bar includes links for Docs, Examples, Icons, Themes, Blog, a search bar, and version v5.3. On the left, a sidebar titled 'Components' lists various UI components like Accordion, Alerts, Badge, etc., with 'Carousel' highlighted. The main content area features a title 'Carousel' and a description: 'A slideshow component for cycling through elements—images or slides of text—like a carousel.' Below this is an advertisement for Adobe Stock. A 'How it works' section follows, containing a bulleted list about the carousel's functionality. To the right, a vertical 'On this page' sidebar lists additional topics such as 'How it works', 'Basic examples', and 'Autoplaying carousels'.

4. Role a tela até encontrar Autoplaying carousels e copie o código html do primeiro exemplo, clicando no ícone de prancheta.

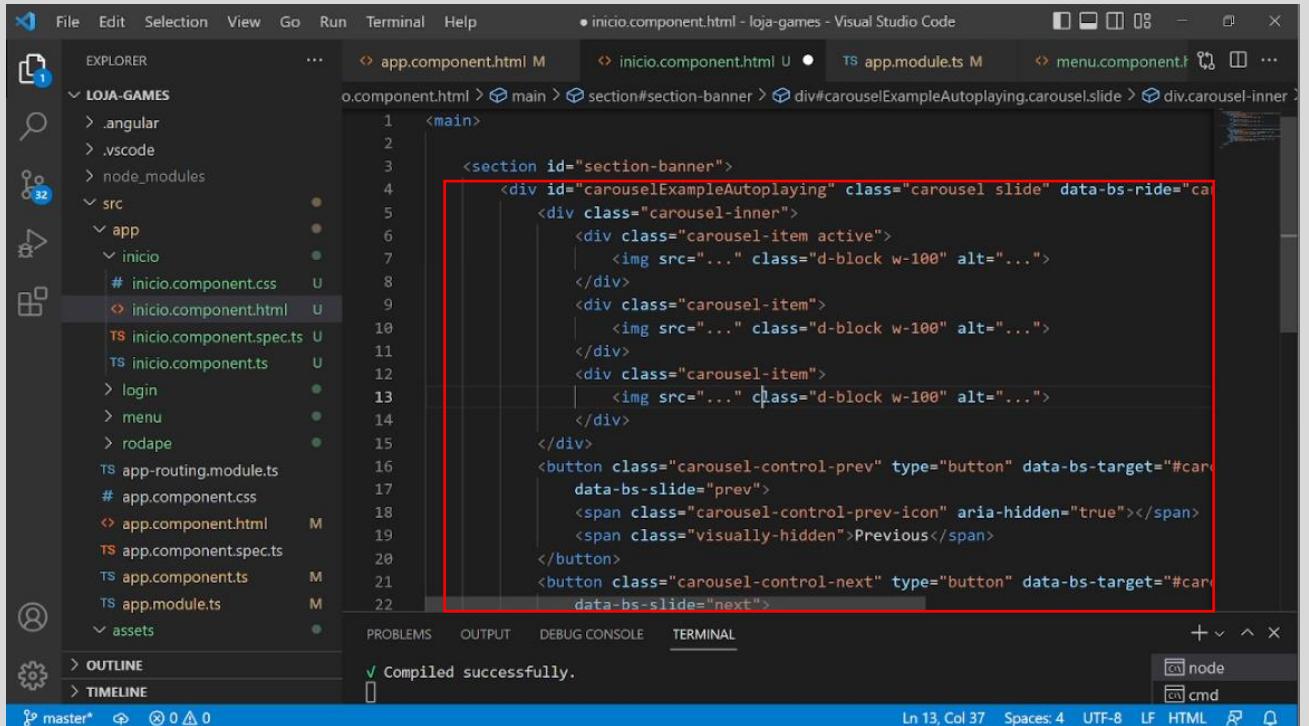
This screenshot shows the same documentation interface as the previous one, but the 'Autoplaying carousels' section is now visible in the 'On this page' sidebar. The 'Components' sidebar still has 'Carousel' selected. The main content area shows an example of an Autoplaying carousel with its corresponding HTML code. A red box highlights the copy icon (a clipboard symbol) next to the code snippet.

5. Volte ao VS Code, abra o arquivo **inicio.component.html** e apague todo o conteúdo. No lugar, digite:

```
<main>
  <section id="section-banner">

  </section>
</main>
```

Cole o código **html do carousel** entre as tags **<section>** e **</section>**.



The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure under "LOJA-GAMES". The "src/app/inicio" folder contains files: # inicio.component.css, inicio.component.html, inicio.component.spec.ts, inicio.component.ts, login, menu, rodape, TS app-routing.module.ts, # app.component.css, app.component.html, TS app.components.spec.ts, TS app.component.ts, TS app.module.ts, and assets.
- Editor:** The "inicio.component.html" file is open. The copied HTML code is pasted into the editor between the <section id="section-banner"> and </section> tags. This copied code is highlighted with a red border.
- Terminal:** Shows the message "Compiled successfully."
- Status Bar:** Displays "Ln 13, Col 37 Spaces: 4 UTF-8 LF HTML" and icons for node and cmd.

6. Altere os valores das três tags , alterando os atributos **src** e **alt** dessas tags, conforme as linhas a seguir:

```

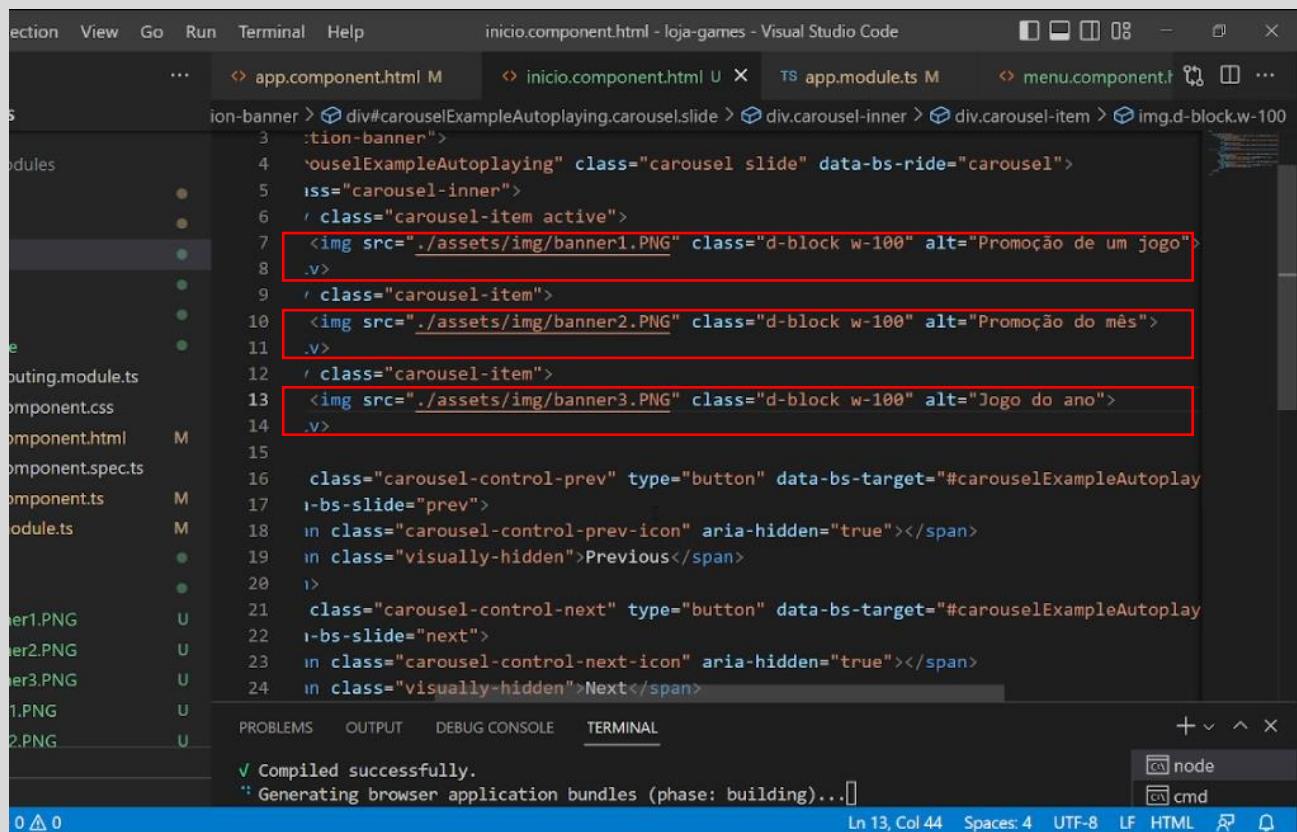
```

```

```

```

```



The screenshot shows the Visual Studio Code interface with the file `inicio.component.html` open. The code editor displays the following HTML code:

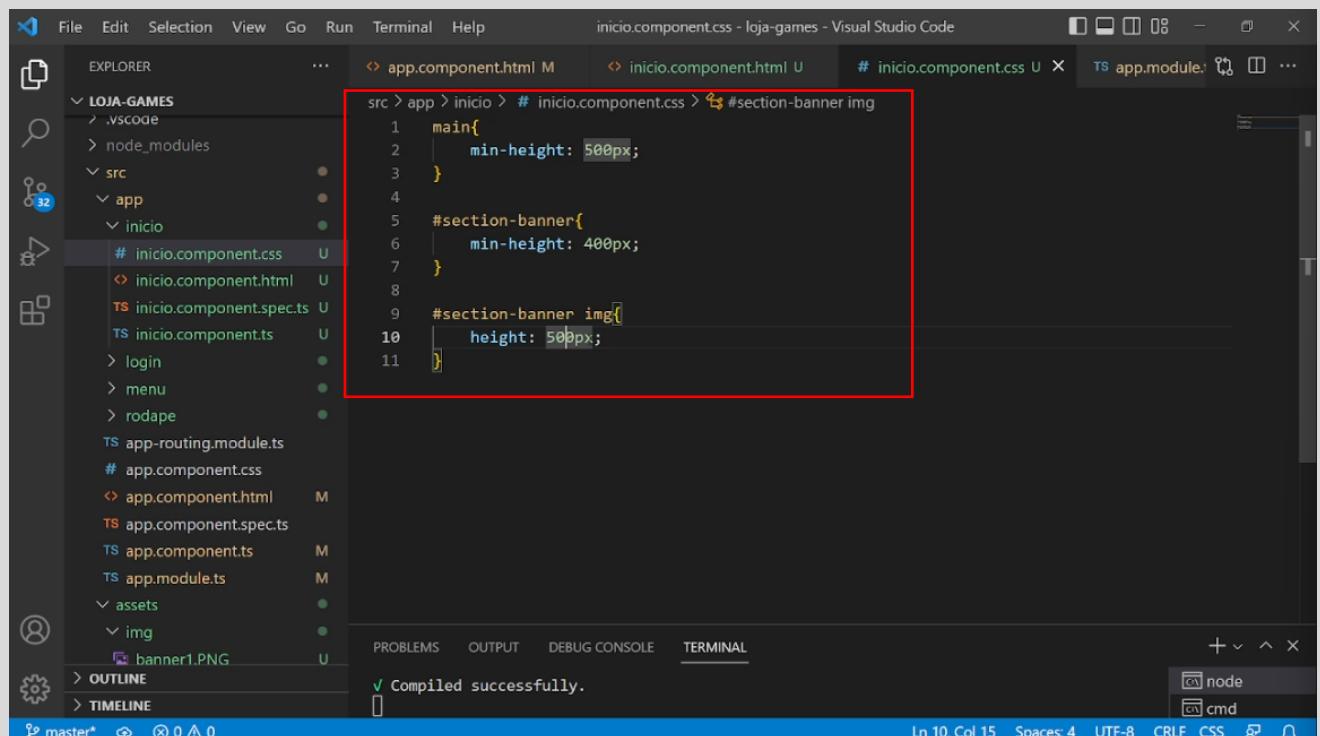
```
ion-banner > div#carouselExampleAutoplaying.carousel.slide > div.carousel-inner > div.carousel-item > img.d-block.w-100  
3   :tion-banner">  
4     'ouselExampleAutoplaying" class="carousel slide" data-bs-ride="carousel">  
5       ss="carousel-inner">  
6         lass="carousel-item active">  
7             
8             .v>  
9           lass="carousel-item">  
10              
11              .v>  
12            lass="carousel-item">  
13                
14                .v>  
15  
16      class="carousel-control-prev" type="button" data-bs-target="#carouselExampleAutoplaying">  
17        i-bs-slide="prev">  
18          n class="carousel-control-prev-icon" aria-hidden="true"></span>  
19          n class="visually-hidden">Previous</span>  
20    >  
21      class="carousel-control-next" type="button" data-bs-target="#carouselExampleAutoplaying">  
22        i-bs-slide="next">  
23          n class="carousel-control-next-icon" aria-hidden="true"></span>  
24          n class="visually-hidden">Next</span>
```

Three `img` tags are highlighted with red boxes, corresponding to the instructions in the exercise. The terminal at the bottom shows the output of a successful compilation:

```
✓ Compiled successfully.  
⠼ Generating browser application bundles (phase: building)...
```

7. No arquivo **inicio.component.css**, digite o seguinte código:

```
main{  
    min-height: 500px;  
}  
  
#section-banner{  
    min-height: 400px;  
}  
  
#section-banner img{  
    height: 500px;  
}
```

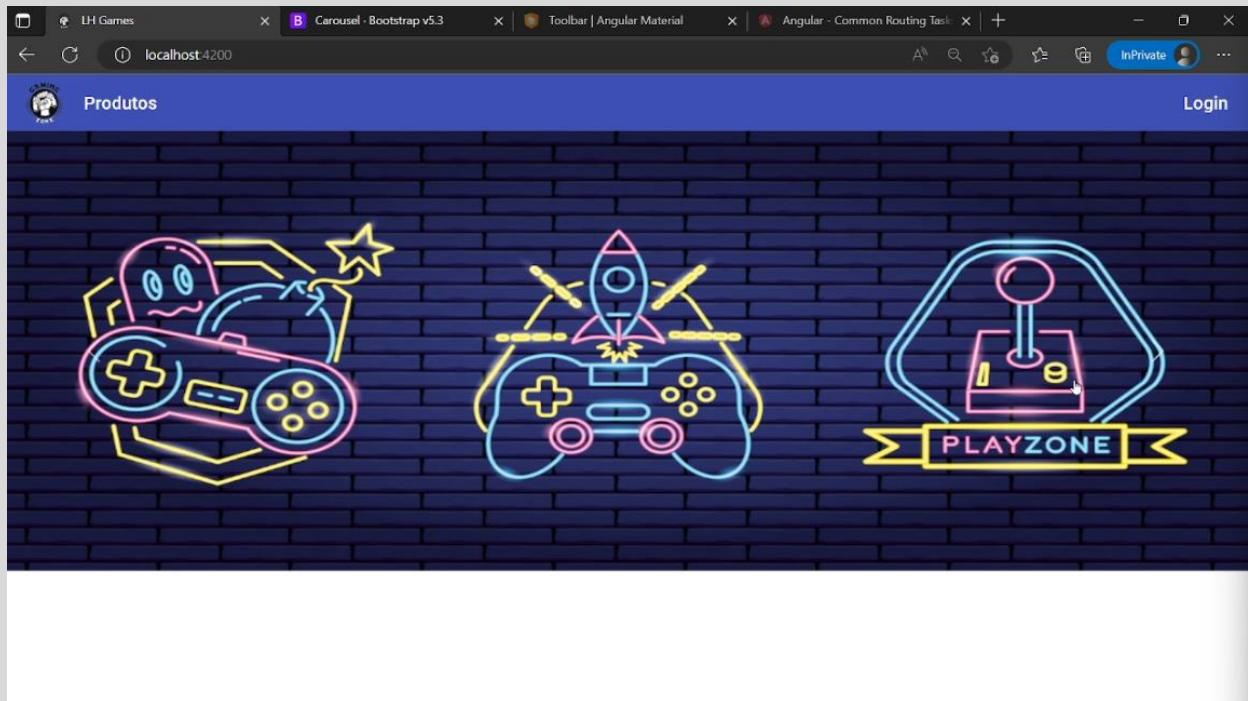


The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure under "LOJA-GAMES". The file "# inicio.component.css" is selected and highlighted.
- Code Editor:** Displays the CSS code from the previous block. A red rectangular box highlights the entire code block.
- Terminal:** Shows the message "Compiled successfully." at the bottom.
- Status Bar:** Shows "Ln 10, Col 15" and other terminal-related information.

Assim as imagens do carousel terão o mesmo tamanho.

8. Atualize o navegador para conferir as alterações.



Importante

Lembre-se sempre de salvar todas as alterações feitas em todos os arquivos.



9. Retorne ao Angular Material e clique em **Card**, dentro de **Components**. Depois clique em **Exemplos**

The screenshot shows the Angular Material Components page. The 'Components' tab is selected, highlighted with a red box. On the left sidebar, the 'Card' component is also highlighted with a red box. The main content area displays the 'OVERVIEW' tab, which contains a brief description of the `<mat-card>` component and a preview of a basic card. The 'EXAMPLES' tab is also highlighted with a red box. A sidebar on the right provides an 'Overview Content' section with links to various card-related sections.

10. Clique no ícone <> do primeiro exemplo.

The screenshot shows the same Angular Material Components page as before, but now the first example under 'Basic cards' is expanded. The 'Simple card' example is shown in a larger preview window. The icon with two arrows (the 'View code' icon) is highlighted with a red box. The rest of the interface remains the same, with the 'Card' component selected in the sidebar.

11. Clique em **TS** e confira se precisa de ajustes. Como não há nada a ser ajustado, não precisa copiar.

```

Material Components CDK Guides
Badge Card with multiple sections 15.1.0 GitHub
Bottom Sheet
Button
Button toggle
Card
Checkbox
Chips
Core
Datepicker
Dialog

HTML TS CSS
import {Component} from '@angular/core';

/**
 * @title Card with multiple sections
 */
@Component({
  selector: 'card-fancy-example',
  templateUrl: 'card-fancy-example.html',
  styleUrls: ['card-fancy-example.css'],
})
export class CardFancyExample {}

```

The screenshot shows the Angular Material Components page with the 'Card' component selected. The 'TS' tab is highlighted with a red box. The code editor displays the TypeScript code for the 'CardFancyExample' component, which defines a component with a specific selector, template URL, and style URLs. Below the code, a preview window shows a card with a header image of a Shiba Inu dog and the text 'Shiba Inu Dog Breed'.

12. Clique em **CSS**, selecione e copie o **código**, que é o mesmo do código a seguir:

```

.example-card {
  max-width: 400px;
}

.example-header-image {
  background-image: url('https://material.angular.io/assets/img/examples/shiba1.jpg');
  background-size: cover;
}

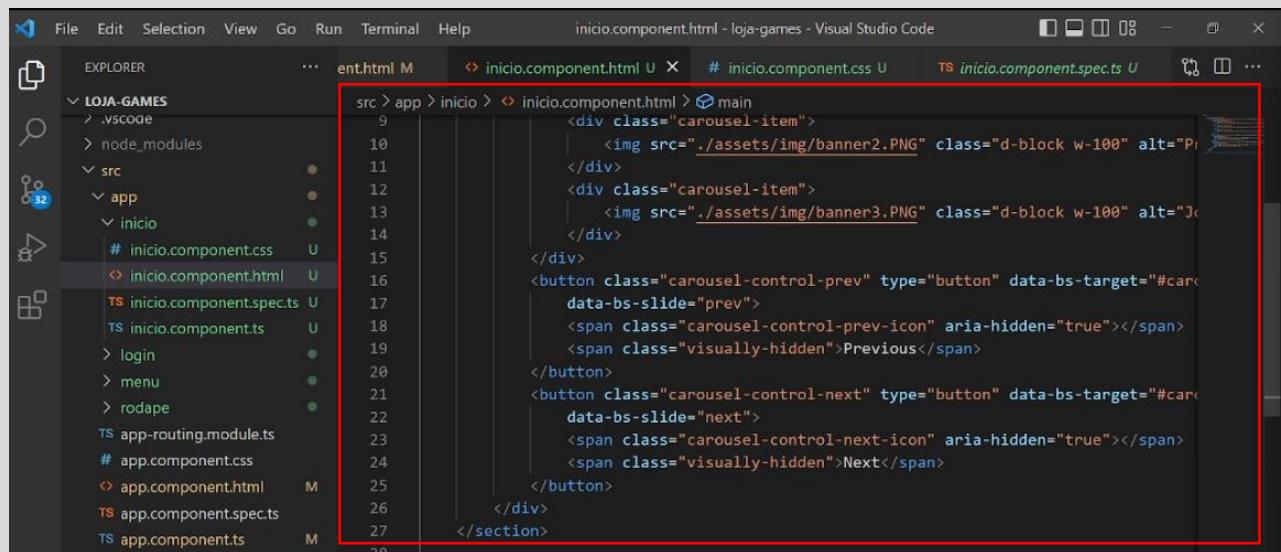
```

E cole no arquivo **inicio.component.css**. Altere a linha **background-size: cover;** para **background-size: auto;**

13. Clique em HTML, selecione e copie o código, que é o mesmo do código a seguir:

```
<mat-card class="example-card">
  <mat-card-header>
    <div mat-card-avatar class="example-header-image"></div>
    <mat-card-title>Shiba Inu</mat-card-title>
    <mat-card-subtitle>Dog Breed</mat-card-subtitle>
  </mat-card-header>
  
  <mat-card-content>
    <p>
      The Shiba Inu is the smallest of the six original and distinct spitz breeds of dog from Japan.
      A small, agile dog that copes very well with mountainous terrain, the Shiba Inu was originally bred for hunting.
    </p>
  </mat-card-content>
  <mat-card-actions>
    <button mat-button>LIKE</button>
    <button mat-button>SHARE</button>
  </mat-card-actions>
</mat-card>
```

E cole o código html no arquivo **inicio.component.html**, entre as tags **</section>** e **</main>**.



```
File Edit Selection View Go Run Terminal Help inicio.component.html - loja-games - Visual Studio Code
EXPLORER ... ent.html M inicio.component.html U # inicio.component.css U TS inicio.component.spec.ts U ...
LOJA-GAMES .vscode
  node_modules
  src
    app
      inicio
        # inicio.component.css U
        <!--> inicio.component.html U
        # inicio.component.spec.ts U
        inicio.component.ts U
        login
        menu
        rodape
        TS app-routing.module.ts
        # app.component.css
        <!--> app.component.html M
        TS app.component.spec.ts
        TS app.component.ts M
src > app > inicio > inicio.component.html > main
  <div class="carousel-item">
    
  </div>
  <div class="carousel-item">
    
  </div>
  <button class="carousel-control-prev" type="button" data-bs-target="#caro
    data-bs-slide="prev">
    <span class="carousel-control-prev-icon" aria-hidden="true"></span>
    <span class="visually-hidden">Previous</span>
  </button>
  <button class="carousel-control-next" type="button" data-bs-target="#caro
    data-bs-slide="next">
    <span class="carousel-control-next-icon" aria-hidden="true"></span>
    <span class="visually-hidden">Next</span>
  </button>
</div>
</section>
```

14. Ajuste o código html conforme a seguir:

```
<mat-card class="example-card">
  
  <mat-card-header>
    <mat-card-title>Jogo 1</mat-card-title>
  </mat-card-header>
    <mat-card-content>
      <p>
        Descrição do jogo.
      </p>
    </mat-card-content>
    <mat-card-actions>
      <p>R$ 300,00</p>
      <button mat-button>Comprar</button>
    </mat-card-actions>
  </mat-card>
```

Esse é o html de **um card**. Você precisa de **três cards seguidos**, alinhados na vertical. Para isso, você precisa de um **Grid list**.

15. Acesse o Angular Material e clique em **Grid List**. O css e o TS não precisam de ajustes. Selecione apenas o código **HTML** e copie.

Angular Material Components CDK Guides

mat-grid-list is a two-dimensional list view that arranges cells into grid-based layout. See Material Design spec [here](#).

Grid list

HTML TS CSS

```
<mat-grid-list cols="2" rowHeight="2:1">
  <mat-grid-tile></mat-grid-tile>
  <mat-grid-tile>2</mat-grid-tile>
  <mat-grid-tile>3</mat-grid-tile>
  <mat-grid-tile>4</mat-grid-tile>
</mat-grid-list>
```

1 2

Code copied

16. Cole o código no **app.component.html** entre as tags **</section>** e **<mat-card>**.

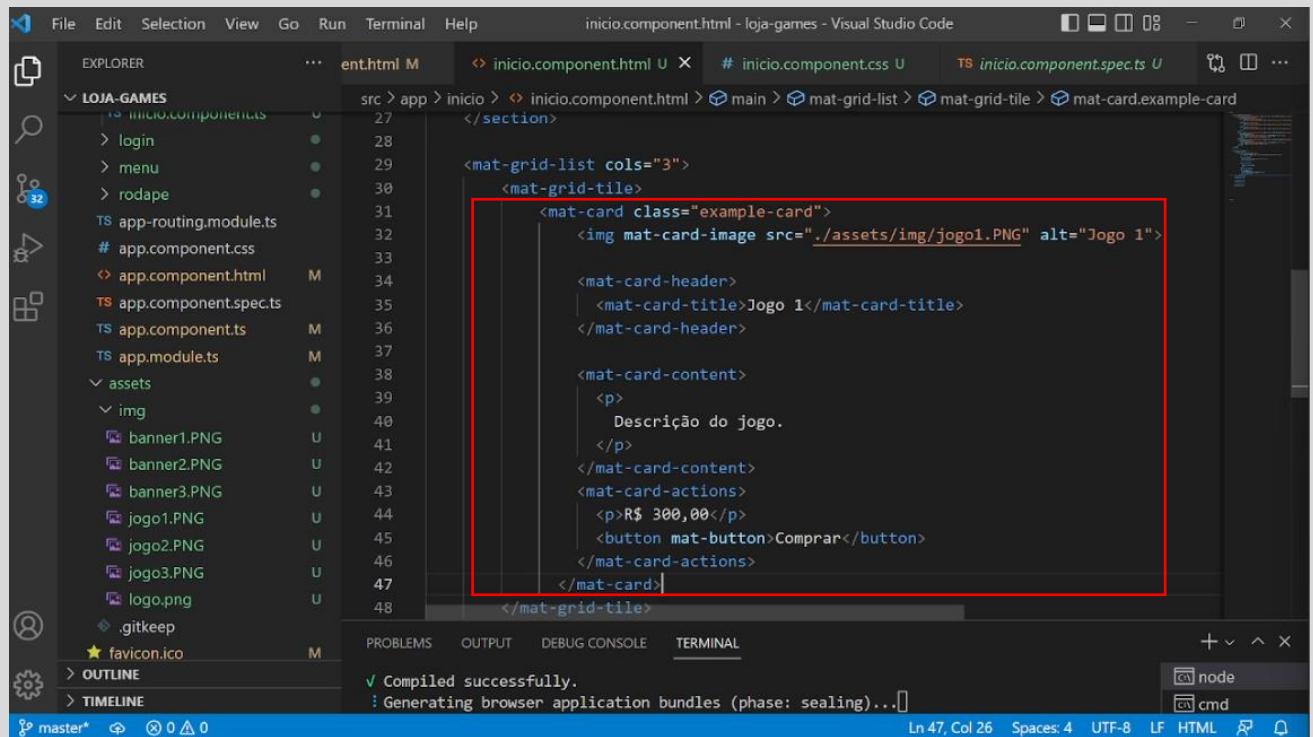
```
</section>
<mat-grid-list cols="2" rowHeight="2:1">
  <mat-grid-tile>1</mat-grid-tile>
  <mat-grid-tile>2</mat-grid-tile>
  <mat-grid-tile>3</mat-grid-tile>
  <mat-grid-tile>4</mat-grid-tile>
</mat-grid-list>

<mat-card class="example-card">
  
  <mat-card-header>
    <mat-card-title>Jogo 1</mat-card-title>
  </mat-card-header>
  <mat-card-content>
    <p>Descrição do jogo...</p>
  </mat-card-content>
</mat-card>
```

17. Altere o código colado conforme a seguir:

```
<mat-grid-list cols="3">
  <mat-grid-tile>1</mat-grid-tile>
  <mat-grid-tile>2</mat-grid-tile>
  <mat-grid-tile>3</mat-grid-tile>
</mat-grid-list>
```

18. Selecione o código do card, entre as tags <mat-card> e </mat-card>, e cole substituindo os numerais 1,2 e 3 do Grid List.



The screenshot shows the Visual Studio Code interface with the file `inicio.component.html` open. The code editor displays the following HTML structure:

```
src > app > inicio > inicio.component.html > main > mat-grid-list > mat-grid-tile > mat-card.example-card
<mat-grid-list cols="3">
  <mat-grid-tile>
    <mat-card class="example-card">
      
      <mat-card-header>
        <mat-card-title>Jogo 1</mat-card-title>
      </mat-card-header>
      <mat-card-content>
        <p>
          Descrição do jogo.
        </p>
      </mat-card-content>
      <mat-card-actions>
        <p>R$ 300,00</p>
        <button mat-button>Comprar</button>
      </mat-card-actions>
    </mat-card>
  </mat-grid-tile>
</mat-grid-list>
```

A red rectangular box highlights the entire content of the first `mat-card` element, starting from its opening tag `<mat-card class="example-card">` and ending at its closing tag `</mat-card>`. This indicates that the user should copy this specific block of code and replace the corresponding numbered tiles in the grid list.

19. Altere o código colado conforme a seguir:

```
<mat-grid-list cols="3">
  <mat-grid-tile>
    <mat-card class="example-card">
      
      <mat-card-header>
        <mat-card-title>Jogo 1</mat-card-title>
      </mat-card-header>
      <mat-card-content>
        <p>Descrição do jogo.</p>
      </mat-card-content>
      <mat-card-actions>
        <p>R$ 300,00</p>
        <button mat-button>Comprar</button>
      </mat-card-actions>
    </mat-card>
  </mat-grid-tile>
  <mat-grid-tile>
    <mat-card class="example-card">
      
      <mat-card-header>
        <mat-card-title>Jogo 2</mat-card-title>
      </mat-card-header>
      <mat-card-content>
        <p>Descrição do jogo.</p>
      </mat-card-content>
      <mat-card-actions>
        <p>R$ 200,00</p>
        <button mat-button>Comprar</button>
      </mat-card-actions>
    </mat-card>
  </mat-grid-tile>
  <mat-grid-tile>
    <mat-card class="example-card">
      
      <mat-card-header>
        <mat-card-title>Jogo 3</mat-card-title>
      </mat-card-header>
      <mat-card-content>
        <p>Descrição do jogo.</p>
      </mat-card-content>
      <mat-card-actions>
        <p>R$ 400,00</p>
        <button mat-button>Comprar</button>
      </mat-card-actions>
    </mat-card>
  </mat-grid-tile>
</mat-grid-list>
```

20. Insira um **título** para o Grid List, digitando antes da tag **<mat-grid-list>**:

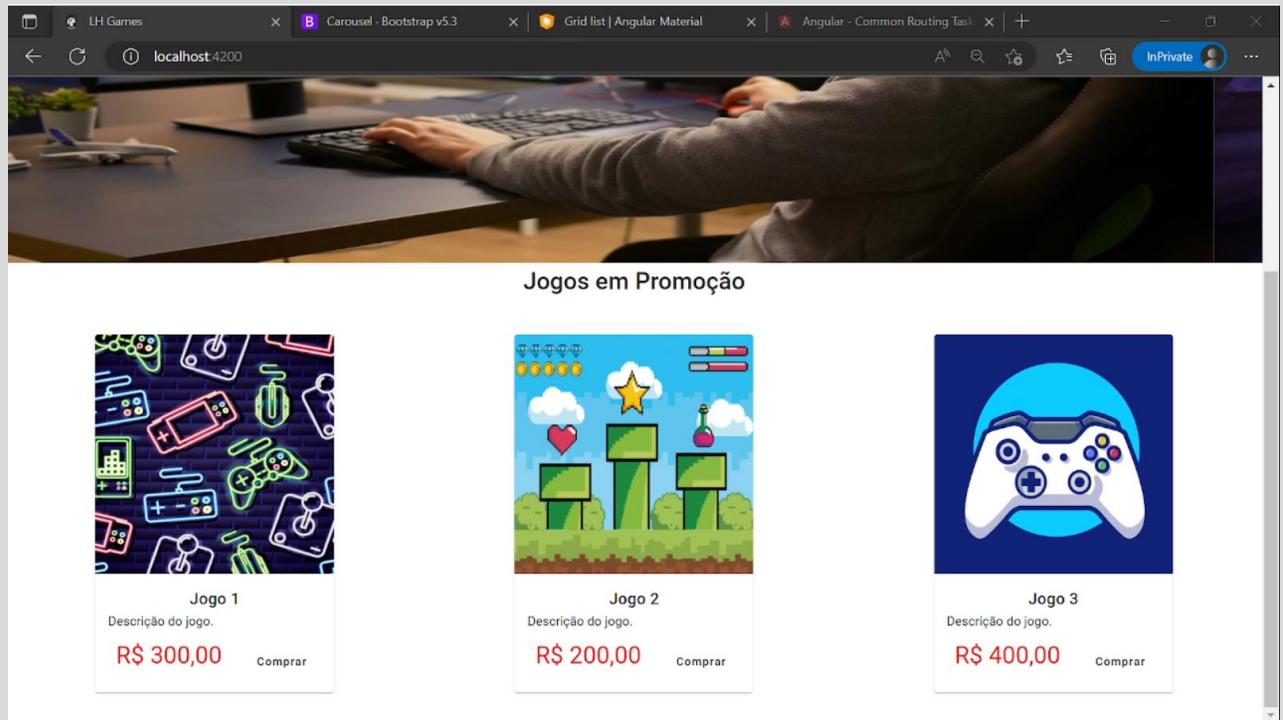
```
<h2>Jogos em Promoção</h2>
```

21. Ajuste o css, colando o código a seguir em **inicio.component.css**:

```
mat-grid-list{  
  margin: 10px 0px;  
}  
mat-grid-tile{  
  padding: 2px;  
  border-radius: 30px;  
t-card-header{  
  display: flex;  
  align-items: center;  
  justify-content: center;  
t-card-actions{  
  display: flex;  
  justify-content: space-around;  
}  
mat-card-actions p{  
  color: red;  
  font-size: 30px;  
}  
mat-card img{  
  height: 300px;  
  width: 300px;  
}
```

Assim as imagens dos cards terão o mesmo tamanho, haverá um espaçamento entre os cards e também entre os elementos dentro do card e os títulos dentro dos cards ficarão centralizados, assim como o título **Jogos em Promoção**

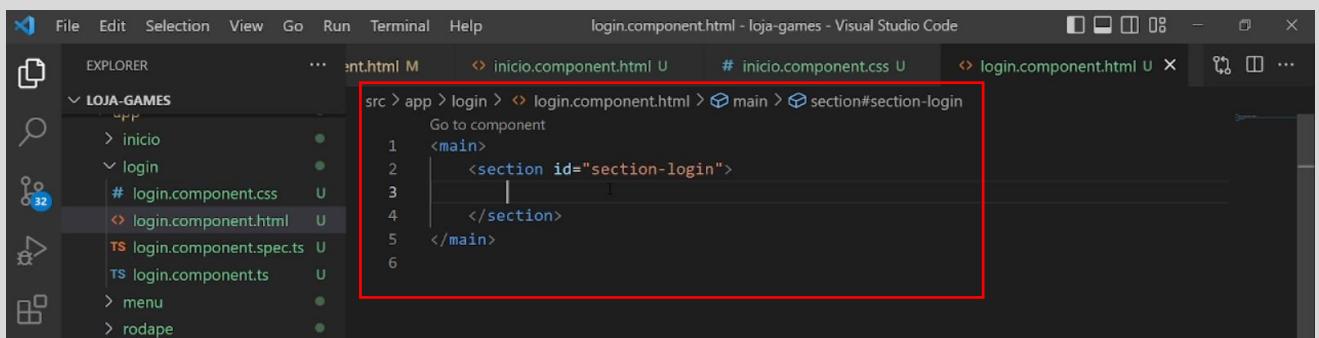
22. Salve todos os arquivos e confira a renderização no navegador.



Login

1. No arquivo **login.component.html**, digite o seguinte código:

```
<main>
  <section id="section-login">
    </section>
</main>
```



2. Acesse <https://getbootstrap.com/> , clique em **Docs** e depois role o menu do lado esquerdo até encontrar **Form Control**.

The screenshot shows the Bootstrap documentation website with the following details:

- Header:** Includes links for 'Docs' (highlighted with a red box), 'Examples', 'Icons', 'Themes', and 'Blog'. There is also a search bar and social media links.
- Sidebar:** A navigation menu on the left side with sections like 'Images', 'Tables', 'Figures', 'Forms' (which is expanded), 'Components' (which is expanded), and others like 'Accordion', 'Alerts', 'Badge', 'Breadcrumb', etc.
- Main Content:** The 'Forms' section is currently active. Under 'Form control', there is sample HTML code for a carousel with controls. The code includes various Bootstrap classes like 'carousel slide', 'carousel-inner', 'carousel-item', and 'carousel-control-prev'.
- Right Sidebar:** A 'On this page' sidebar on the right lists various Bootstrap components and features, such as 'How it works', 'Basic examples', 'Indicators', 'Captions', 'Crossfade', 'Autoplaying carousels', 'Individual .carousel-item interval', 'Autoplaying carousels without controls', 'Disable touch swiping', 'Dark variant', 'Custom transition', 'Sass', 'Variables', 'Usage', 'Via data attributes', 'Via JavaScript', 'Options', and 'Methods'.

3. Escolha o **sexto exemplo**, com email de exemplo, campo de entrada de dados e botão para enviar. Clique no ícone de prancheta para copiar todo o código html.

The screenshot shows the Bootstrap documentation page for 'Form control'. On the left, there's a sidebar with navigation links like Reboot, Typography, Images, Tables, Figures, and a 'Forms' section that includes 'Overview', 'Form control' (which is currently selected), 'Select', 'Checks & radios', 'Range', and 'Input group'. The main content area displays a login form with two input fields: 'email@example.com' and 'Password', followed by a blue button labeled 'Confirm identity'. A red box highlights the two input fields. Below the form, the HTML code is shown in a code editor:

```
<form class="row g-3">
  <div class="col-auto">
    <label for="staticEmail2" class="visually-hidden">Email</label>
    <input type="text" readonly class="form-control-plaintext" id="staticEmail2" value="email@example.com" placeholder="Email" aria-label="Email" aria-describedby="inputGroupPrepend" style="width: 100%; height: 100%; border: none; border-radius: 0; background-color: transparent; font-size: 1em; font-weight: bold; font-family: inherit; padding: 0; margin: 0; position: relative; z-index: 0; opacity: 1;">
  </div>
  <div class="col-auto">
    <label for="inputPassword2" class="visually-hidden">Password</label>
    <input type="password" class="form-control" id="inputPassword2" placeholder="Password" style="width: 100%; height: 100%; border: none; border-radius: 0; background-color: transparent; font-size: 1em; font-weight: bold; font-family: inherit; padding: 0; margin: 0; position: relative; z-index: 0; opacity: 1;">
  </div>
  <div class="col-auto">
    <button type="submit" class="btn btn-primary mb-3">Confirm identity</button>
  </div>
</form>
```

4. Cole dentro de **login.component.html**, entre as tags **<section>** e **</section>**.

The screenshot shows Visual Studio Code with the file 'login.component.html' open. The code editor displays the following HTML structure:

```
<main>
  <section id="section-login">
    <form class="row g-3">
      <div class="col-auto">
        <label for="staticEmail2" class="visually-hidden">Email</label>
        <input type="text" readonly class="form-control-plaintext" id="staticEmail2" value="email@example.com" placeholder="Email" aria-label="Email" aria-describedby="inputGroupPrepend" style="width: 100%; height: 100%; border: none; border-radius: 0; background-color: transparent; font-size: 1em; font-weight: bold; font-family: inherit; padding: 0; margin: 0; position: relative; z-index: 0; opacity: 1;">
      </div>
      <div class="col-auto">
        <label for="inputPassword2" class="visually-hidden">Password</label>
        <input type="password" class="form-control" id="inputPassword2" placeholder="Password" style="width: 100%; height: 100%; border: none; border-radius: 0; background-color: transparent; font-size: 1em; font-weight: bold; font-family: inherit; padding: 0; margin: 0; position: relative; z-index: 0; opacity: 1;">
      </div>
      <div class="col-auto">
        <button type="submit" class="btn btn-primary mb-3">Confirm identity</button>
      </div>
    </form>
  </section>
</main>
```

5. Ajuste o **html** conforme a seguir:

```
<main>
  <section id="section-login">
    <h2>Login</h2>
    <form class="row g-3">

      <label for="staticUsuario" class="visually-hidden">Usuário</label>
      <input type="text" class="form-control" id="staticUsuario"
placeholder="email@exemplo.com">

      <br>

      <label for="inputPassword2" class="visually-hidden">Senha</label>
      <input type="password" class="form-control" id="inputPassword2"
placeholder="senha">
      <br>
      <button type="submit" class="btn btn-primary mb-3">Login</button>

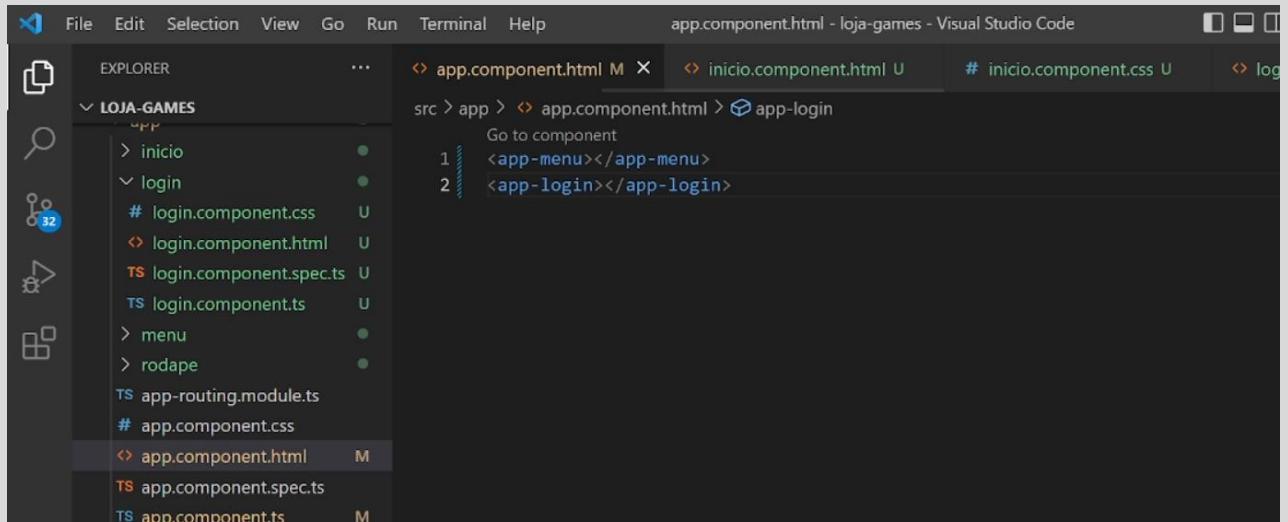
    </form>
  </section>
</main>
```

6. Abra o arquivo **login.component.css** e insira o código:

```
main{
  display: flex;
  justify-content: center;
  align-items: center;
  min-height: 700px;
}

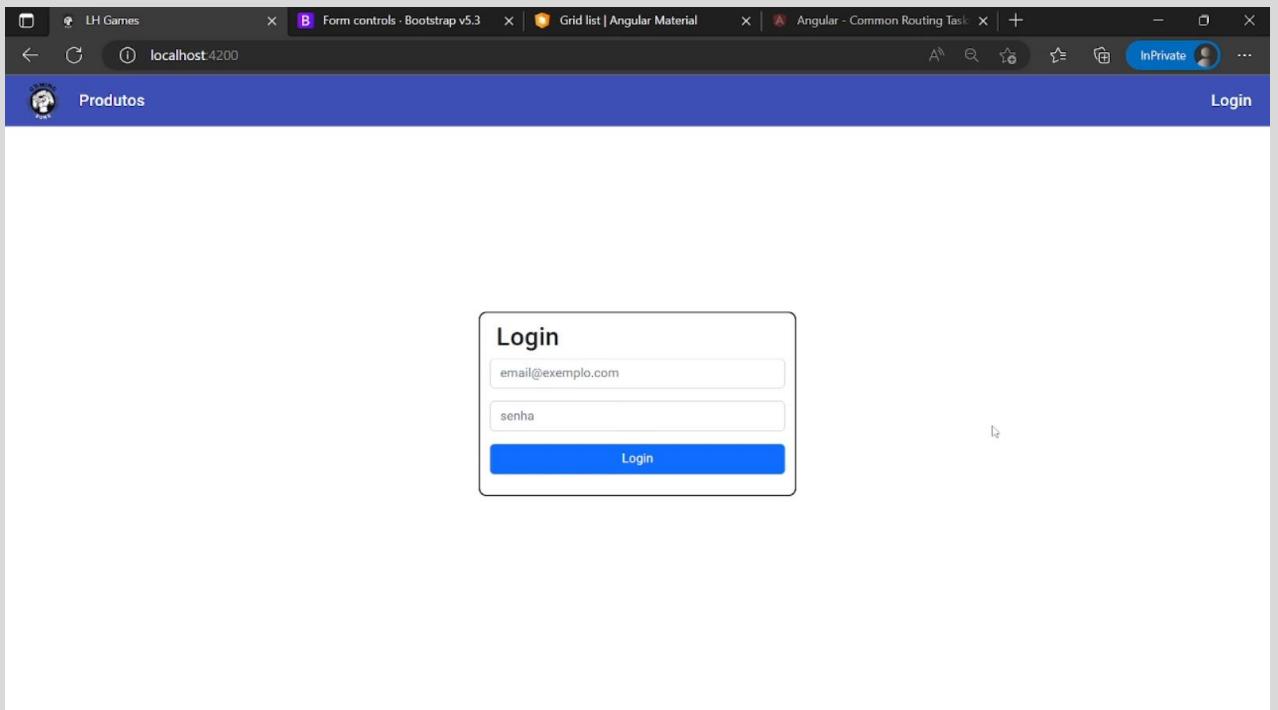
#section-login{
  height: 100%;
  width: 400px;
  border-radius: 10px;
  border: 2px solid black;
  padding: 10px 20px;
}
```

7. Abra o arquivo **app.component.html** e troque a tag **<ap-inicio>** e **</ap-inicio>** por **<app-login>** e **</app-login>**.



```
File Edit Selection View Go Run Terminal Help app.component.html - loja-games - Visual Studio Code
EXPLORER
LOJA-GAMES
  inicio
  login
    login.component.css
    login.component.html
    login.component.spec.ts
    login.component.ts
  menu
  rodape
  app-routing.module.ts
  app.component.css
  app.component.html
  app.component.spec.ts
  app.component.ts
src > app > app.component.html > app-login
  1 <app-menu></app-menu>
  2 <app-login></app-login>
```

8. Salve todos os arquivos e atualize o navegador.

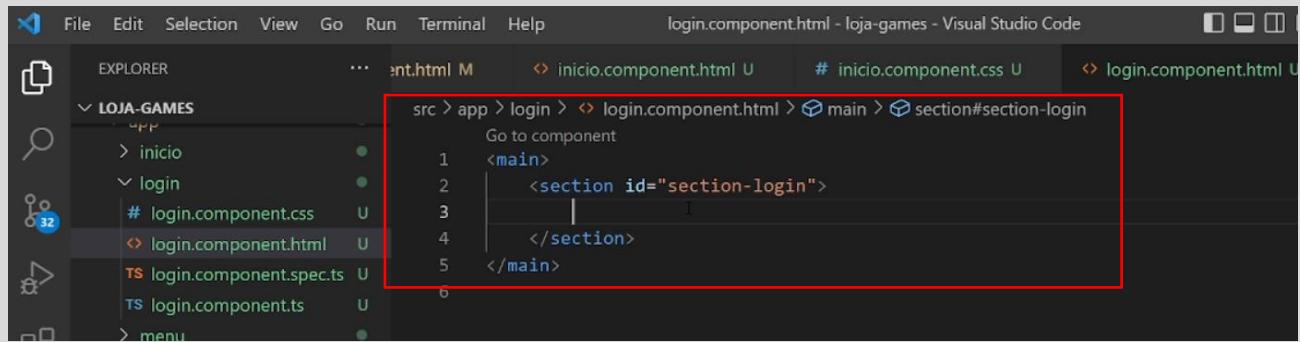


Rodapé

- No arquivo **rodape.component.html**, digite o seguinte código:

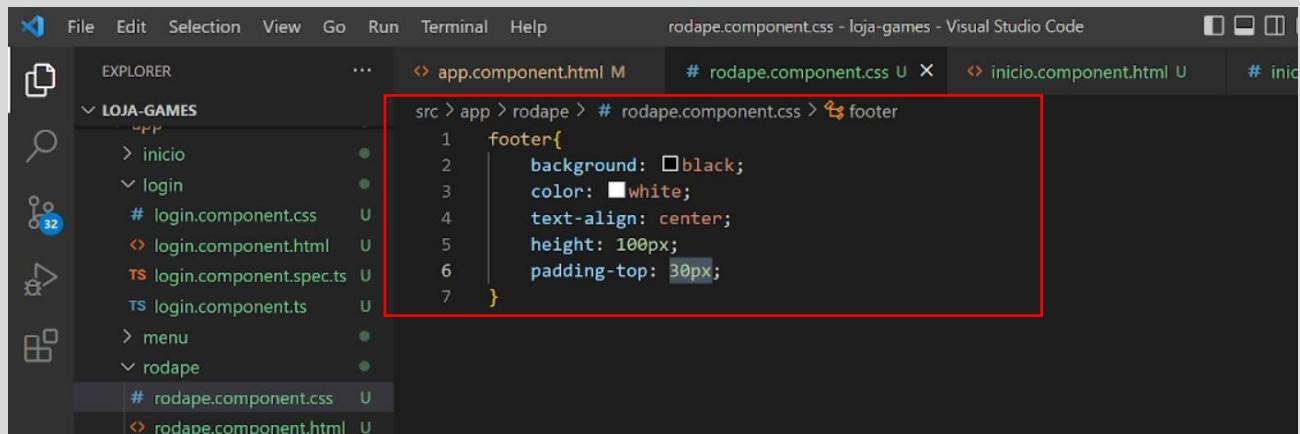
```
<footer>
  <p>Desenvolvido por Felipe</p>
</footer>
```

Obs.: Troque Felipe pelo seu nome.

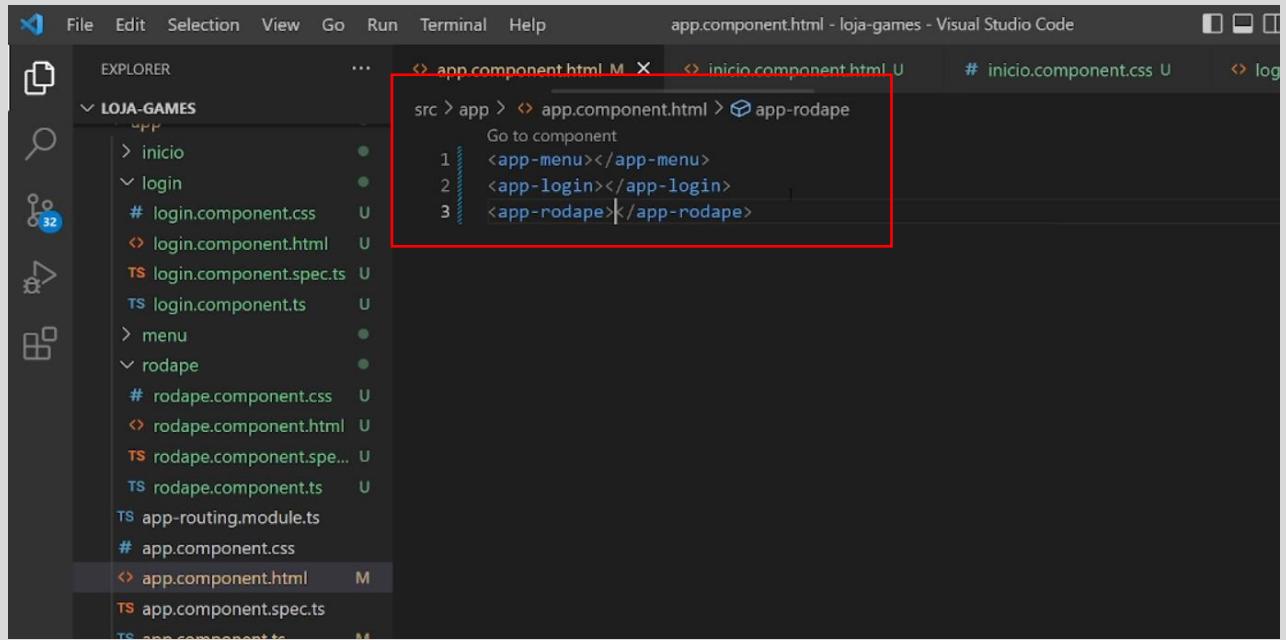


- No arquivo **rodape.component.css**, digite o seguinte código:

```
footer{
  background: black;
  color: white;
  text-align: center;
  height: 100px;
  padding-top: 30px;
}
```



3. No arquivo **app.component.html**, insira **<app-rodapé></app-rodapé>** na ultima linha.

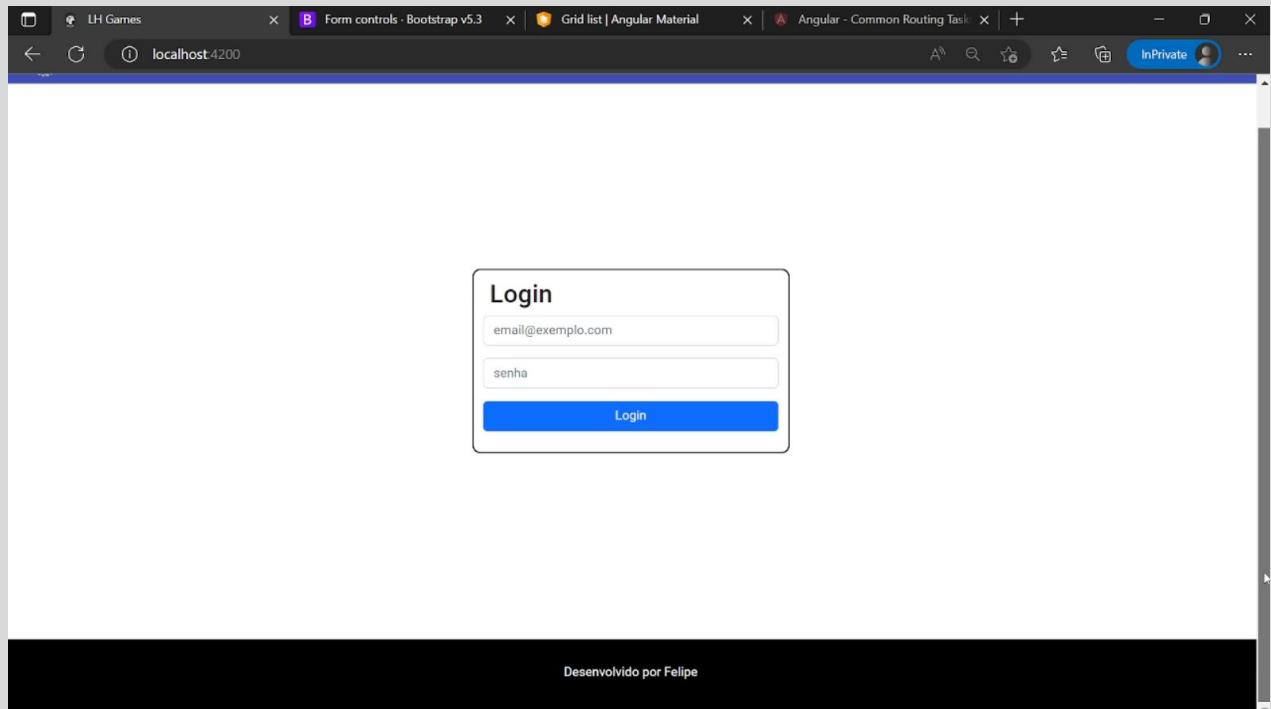


The screenshot shows the Visual Studio Code interface with the file `app.component.html` open. The code editor displays the following HTML structure:

```
<app-menu></app-menu>
<app-login></app-login>
<app-rodape></app-rodape>
```

A red box highlights the last line of the code, `<app-rodape></app-rodape>`. The left sidebar shows the project structure under the `LOJA-GAMES` folder, including files like `login.component.css`, `login.component.html`, and `rodape.component.css`.

4. Salve todos os arquivos e atualize o navegador.



Rotas

1. Acesse angular.io/docs e clique em **Developer guides** (Guias do desenvolvedor) e depois em **Routing and navigation** (roteamento e navegação) e então em **Common routing tasks** (tarefas comuns de roteamento). Role a tela até encontrar **AppRoutingModule**.

The screenshot shows the Angular documentation website. The left sidebar has a tree view with several sections. The 'Guías do desenvolvedor' section is expanded, and its 'Roteamento e navegação' child section is also expanded. The 'Tarefas comuns de roteamento' section is highlighted with a red box. The main content area displays the 'AppRoutingModule (excerpt)' code block, which contains the following code:

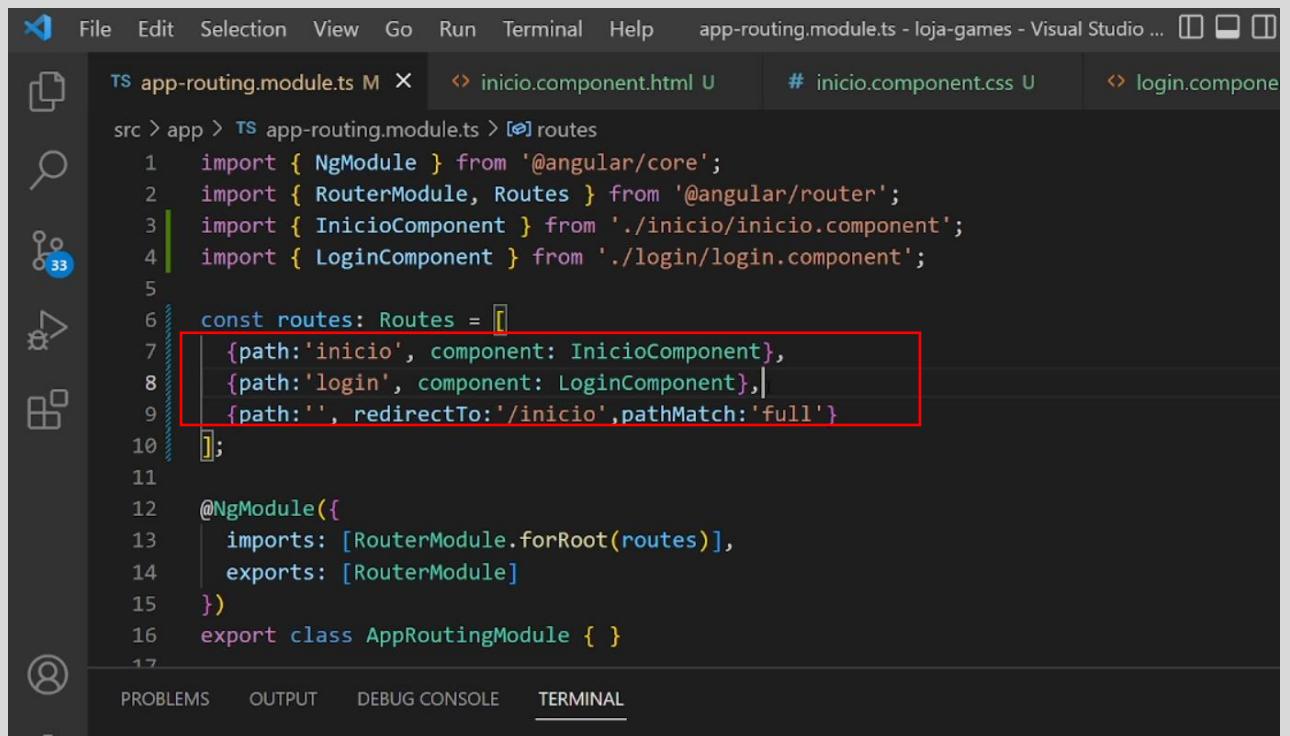
```
const routes: Routes = [
  { path: 'first-component', component: FirstComponent },
  { path: 'second-component', component: SecondComponent },
];
```

Below this, there is a note about adding routes to the application and a 'Modelo com routerLink e router-outlet' section.

Path é o caminho e pode ter qualquer nome. O componente é o elemento que será chamado pelo path.

3. No arquivo **app-routing.module.ts**, insira o seguinte código entre os colchetes de **const routes: Routes =**

```
{path:'inicio', component: InicioComponent},  
{path:'login', component: LoginComponent},  
{path:'', redirectTo:'/inicio',pathMatch:'full'}
```

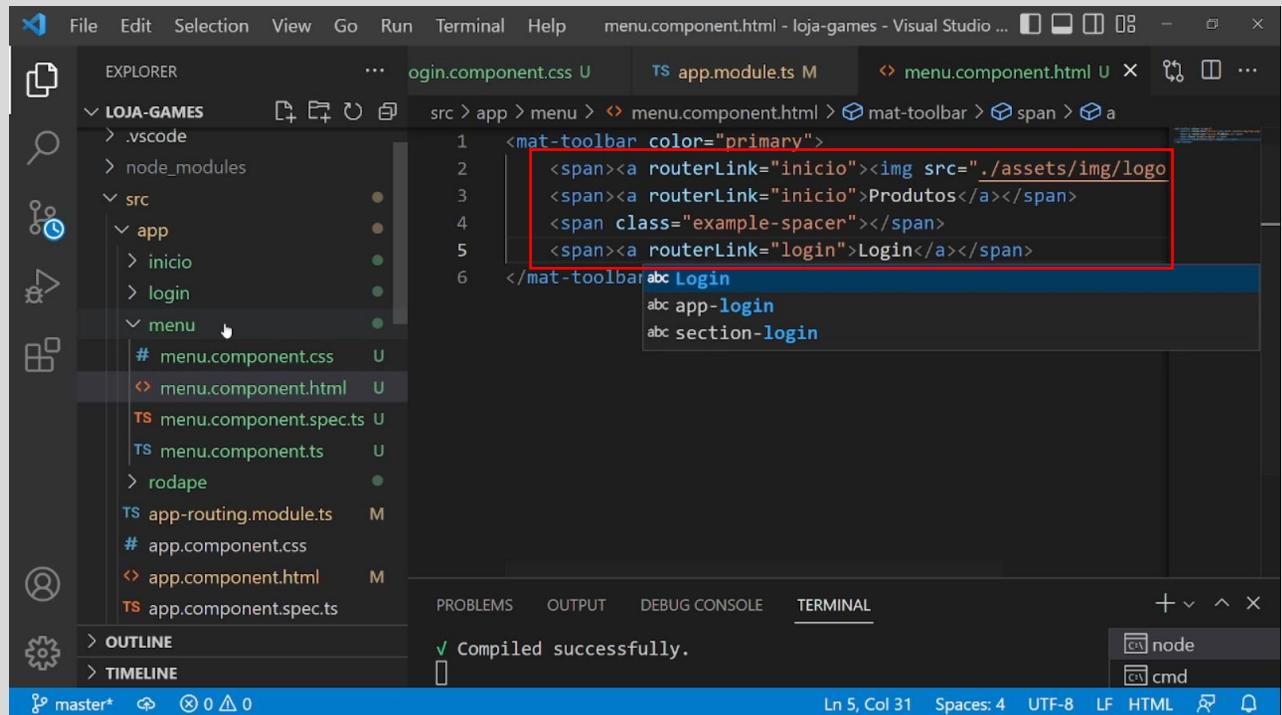


```
File Edit Selection View Go Run Terminal Help app-routing.module.ts - loja-games - Visual Studio ... □ □ □  
src > app > TS app-routing.module.ts M X ↳ inicio.component.html U # inicio.component.css U ↳ login.compone  
1 import { NgModule } from '@angular/core';  
2 import { RouterModule, Routes } from '@angular/router';  
3 import { InicioComponent } from './inicio/inicio.component';  
4 import { LoginComponent } from './login/login.component';  
5  
6 const routes: Routes = [  
7   {path:'inicio', component: InicioComponent},  
8   {path:'login', component: LoginComponent},  
9   {path:'', redirectTo:'/inicio',pathMatch:'full'}  
10];  
11  
12 @NgModule({  
13   imports: [RouterModule.forRoot(routes)],  
14   exports: [RouterModule]  
15 })  
16 export class AppRoutingModule { }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

4. No arquivo **menu.component.html**, insira os links conforme o código a seguir:

```
<mat-toolbar color="primary">
  <span><a routerLink="inicio"></a></span>
  <span><a routerLink="inicio">Produtos</a></span>
  <span class="example-spacer"></span>
  <span><a routerLink="login">Login</a></span>
</mat-toolbar>
```



The screenshot shows the Visual Studio Code interface with the following details:

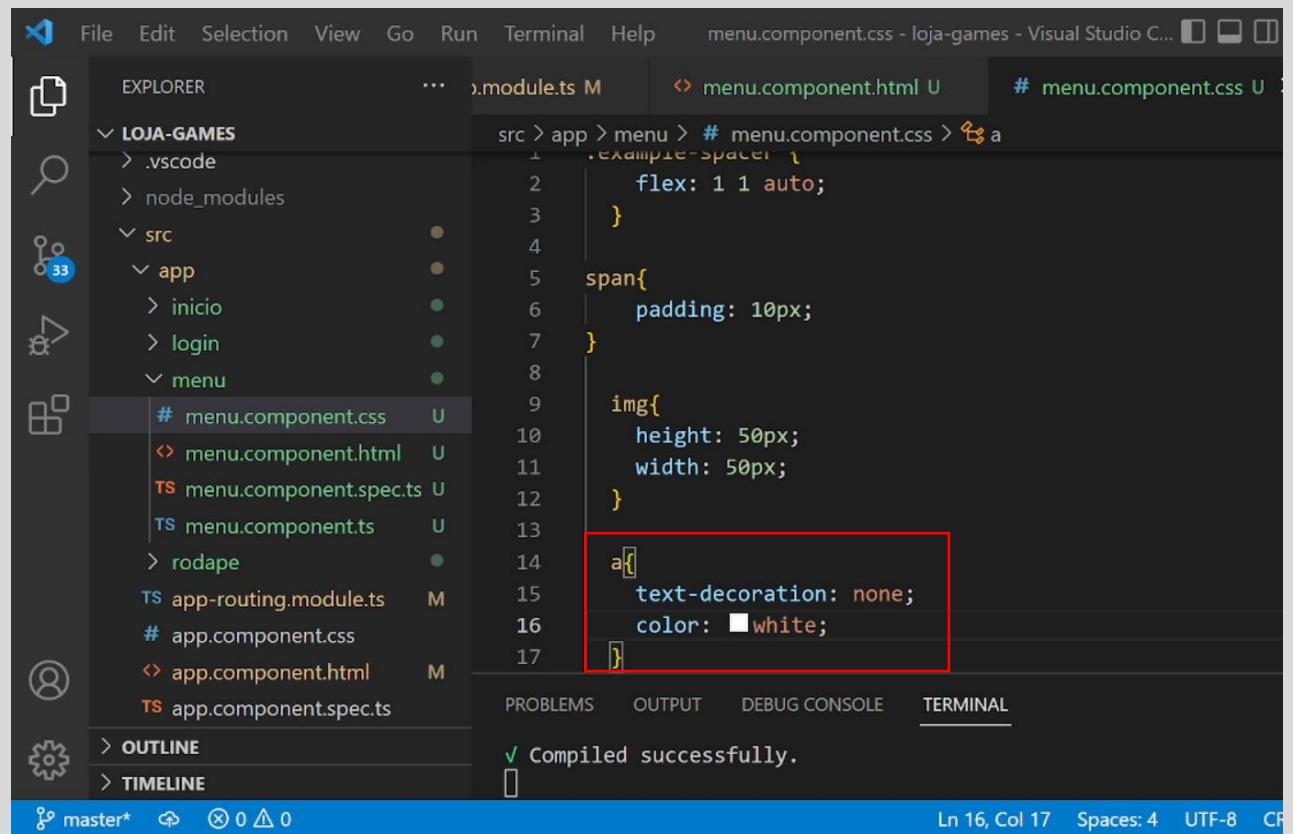
- File Explorer:** Shows the project structure under "LOJA-GAMES". The "menu" folder is selected, and its contents are listed: menu.component.css, menu.component.html, menu.component.spec.ts, and menu.component.ts.
- Editor:** The "menu.component.html" file is open. The code is displayed as follows:

```
<mat-toolbar color="primary">
  <span><a routerLink="inicio"></a></span>
  <span><a routerLink="inicio">Produtos</a></span>
  <span class="example-spacer"></span>
  <span><a routerLink="login">Login</a></span>
</mat-toolbar>
```

The lines from 1 to 5 are highlighted with a red rectangle.
- Terminal:** The terminal shows the message "Compiled successfully."
- Status Bar:** Shows the current branch as "master*", file count as "0", and other status information like "Ln 5, Col 31" and "Spaces: 4".

5. No arquivo **menu.component.css**, insira o seguinte código ao final:

```
a{  
  text-decoration: none;  
  color: white;  
}
```



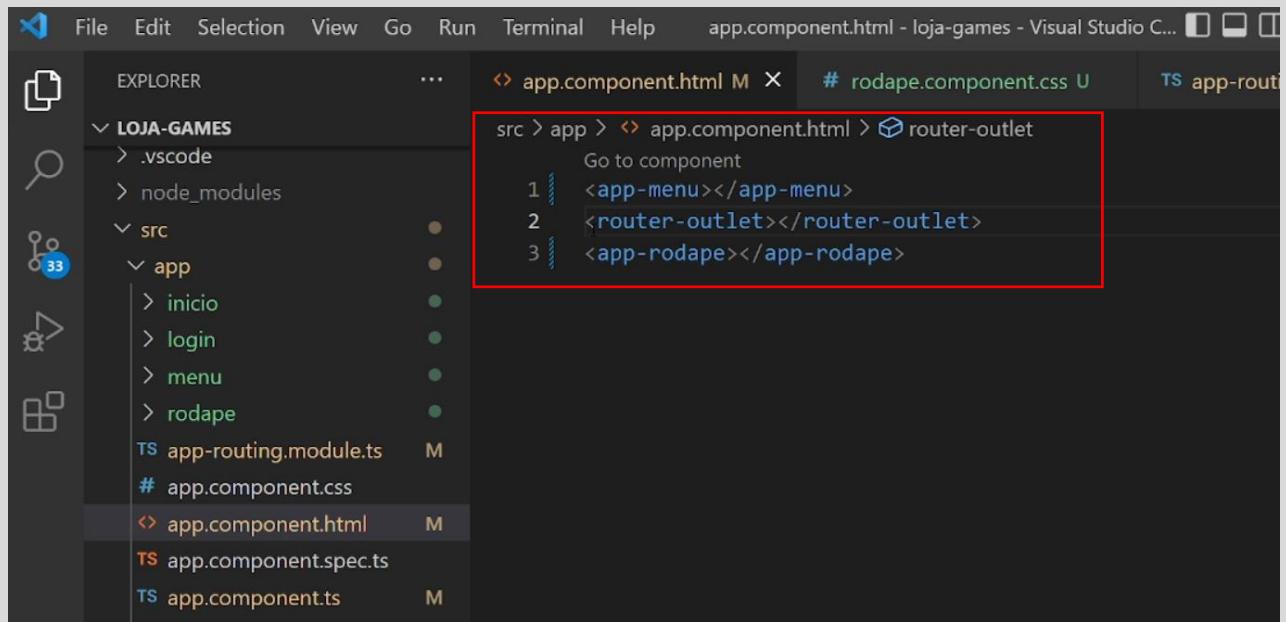
The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Editor:** The file `menu.component.css` is open, showing the following CSS code:

```
src > app > menu > # menu.component.css > a  
  flex: 1 1 auto;  
}  
span{  
  padding: 10px;  
}  
img{  
  height: 50px;  
  width: 50px;  
}  
a{  
  text-decoration: none;  
  color: white;  
}
```
- Explorer:** Shows the project structure under `LOJA-GAMES`, including `.vscode`, `node_modules`, `src` (with `app`, `menu`), and files like `module.ts`, `menu.component.html`, `menu.component.spec.ts`, `menu.component.ts`, `rodape`, `app-routing.module.ts`, `app.component.css`, `app.component.html`, and `app.component.spec.ts`.
- Terminal:** Shows the message `Compiled successfully.`
- Status Bar:** master*, 0 △ 0, Ln 16, Col 17, Spaces: 4, UTF-8, CR.

6. No arquivo **app.component.html**, ajuste o código conforme a seguir:

```
<app-menu></app-menu>
<router-outlet></router-outlet>
<app-rodape></app-rodape>
```



O **<router-outlet>** busca as rotas definidas no **app.routing.module.ts**. Lembre-se de fazer a importação desse módulo.

7. Salve todos os arquivos e verifique os links e rotas, atualizando o navegador.

Jogos em Promoção

Login

email@example.com

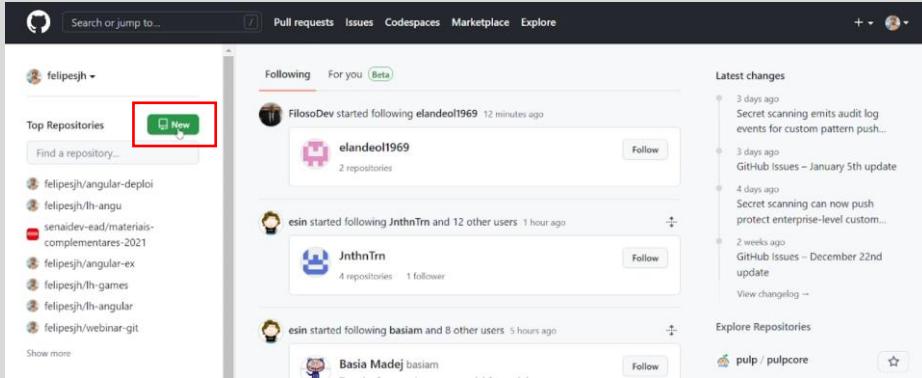
senha

Login

Desenvolvido por Felipe

Publicação no GitHub

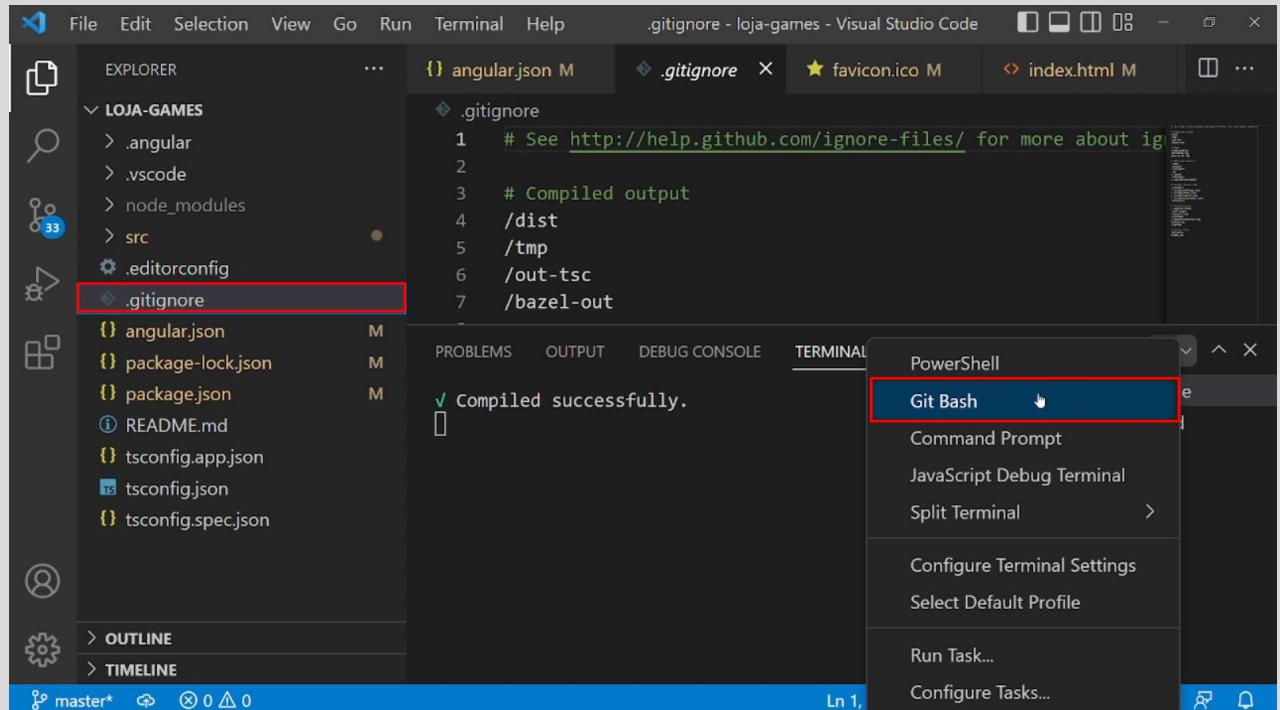
1. Acesse seu repositório em github.com, faça seu login e clique em **New**.



2. Nomeie o novo repositório e clique em **Create Repository**.

 A screenshot of the 'Create a new repository' form on GitHub. The form has fields for 'Repository template' (set to 'No template'), 'Owner' (set to 'felipesjh'), and 'Repository name' (set to 'lh-games-loja', which is highlighted with a red box). There's a note about repository names being available. The 'Description (optional)' field is empty. Below, there are radio buttons for 'Public' (selected) and 'Private'. Under 'Initialize this repository with:', there are options for 'Add a README file' (unchecked) and 'Add .gitignore' (unchecked). The 'Choose a license' section shows 'None' selected. At the bottom, the 'Create repository' button is highlighted with a red box.

3. Retorne ao VS Code e abra um terminal do Git Bash



Observe que há arquivos relacionados ao Git no lado esquerdo

Importante

Lembre-se que você precisa ter instalado o Git Bash em sua máquina.



4. Digite **git status** no terminal e dê **Enter**.

The screenshot shows the Visual Studio Code interface. In the top navigation bar, the tabs are File, Edit, Selection, View, Go, Run, Terminal, Help, and .gitignore - loja-games - Visual Studio Code. Below the tabs, there are several files listed: angular.json M, .gitignore X, favicon.ico M, index.html M, app.component.html M, and a file with a redacted path. On the left sidebar, there are icons for file, folder, search, and other project-related functions. The main workspace shows the .gitignore file open, containing the following content:

```
# See http://help.github.com/ignore-files/ for more about ignoring files.  
# Compiled output  
.dist  
.tmp  
.out-tsc  
.bazel-out
```

Below the workspace, there are tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, and TERMINAL. The TERMINAL tab is selected, showing a terminal session with the command \$ git status. The output of the command is visible in the terminal window.

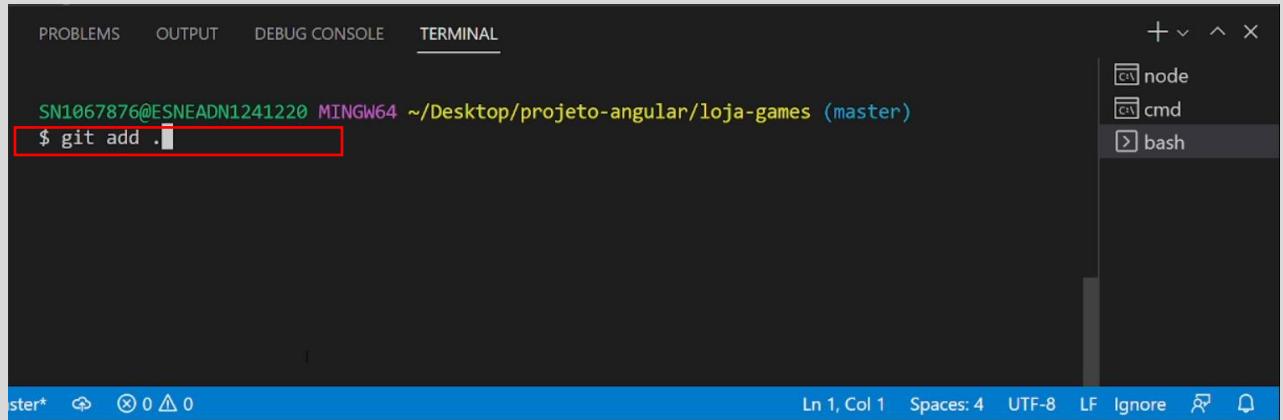
Os arquivos do projeto devem aparecer em vermelho.

The screenshot shows the Visual Studio Code interface with the TERMINAL tab selected. The terminal window displays the output of the git status command. The output shows untracked files:

```
Untracked files:  
(use "git add <file>..." to include in what will be committed)  
src/app/inicio/  
src/app/login/  
src/app/menu/  
src/app/rodape/  
src/assets/img/
```

A red box highlights the list of untracked files. Below this, the message "no changes added to commit (use "git add" and/or "git commit -a")" is shown. At the bottom of the terminal window, the command \$ is followed by a cursor, indicating the user can type more commands.

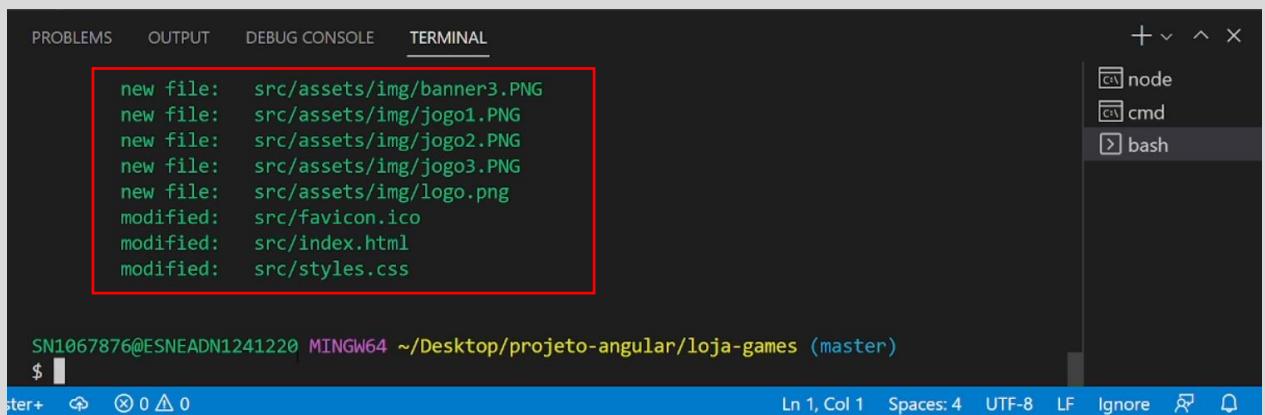
5. Digite **git add .** e dê **Enter**.



A screenshot of a terminal window in Visual Studio Code. The window title is 'TERMINAL'. The status bar at the bottom shows 'Ln 1, Col 1' and 'Spaces: 4'. The terminal output shows the command '\$ git add .' with a red box highlighting it. The right sidebar shows icons for node, cmd, and bash.

6. Digite **git status** novamente e dê **Enter**.

Os arquivos da atividade devem aparecer verdes agora.



A screenshot of a terminal window in Visual Studio Code. The window title is 'TERMINAL'. The status bar at the bottom shows 'Ln 1, Col 1' and 'Spaces: 4'. The terminal output shows the command '\$ git status' with a red box highlighting the output area. The output lists several files: 'new file: src/assets/img/banner3.PNG', 'new file: src/assets/img/jogo1.PNG', 'new file: src/assets/img/jogo2.PNG', 'new file: src/assets/img/jogo3.PNG', 'new file: src/assets/img/logo.png', 'modified: src/favicon.ico', 'modified: src/index.html', and 'modified: src/styles.css'. The right sidebar shows icons for node, cmd, and bash.

7. Digite `git commit -m "meu primeiro projeto em Angular"` e dê Enter.

The screenshot shows the VS Code interface with the Terminal tab selected. The terminal window displays the following text:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

new file: src/assets/img/jogo1.PNG
new file: src/assets/img/jogo2.PNG
new file: src/assets/img/jogo3.PNG
new file: src/assets/img/logo.png
modified: src/favicon.ico
modified: src/index.html
modified: src/styles.css

SN1067876@ESNEADN1241220 MINGW64 ~/Desktop/projeto-angular/loja-games (master)
$ git commit -m "Meu primeiro projeto em Angular"
```

The command `git commit -m "Meu primeiro projeto em Angular"` is highlighted with a red box. The status bar at the bottom of the terminal window shows "Ln 1, Col 1" and "Spaces: 4".

8. Retorne ao GitHub e copie a linha do repositório online `git remote add origin ...`

The screenshot shows a browser window with the URL <https://github.com/felipesjh/lh-games-loja>. The page provides instructions for setting up a new repository:

Quick setup — if you've done this kind of thing before

Set up in Desktop or **HTTPS** <https://github.com/felipesjh/lh-games-loja.git>

Get started by creating a new file or uploading an existing file. We recommend every repository include a `README`, `LICENSE`, and `.gitignore`.

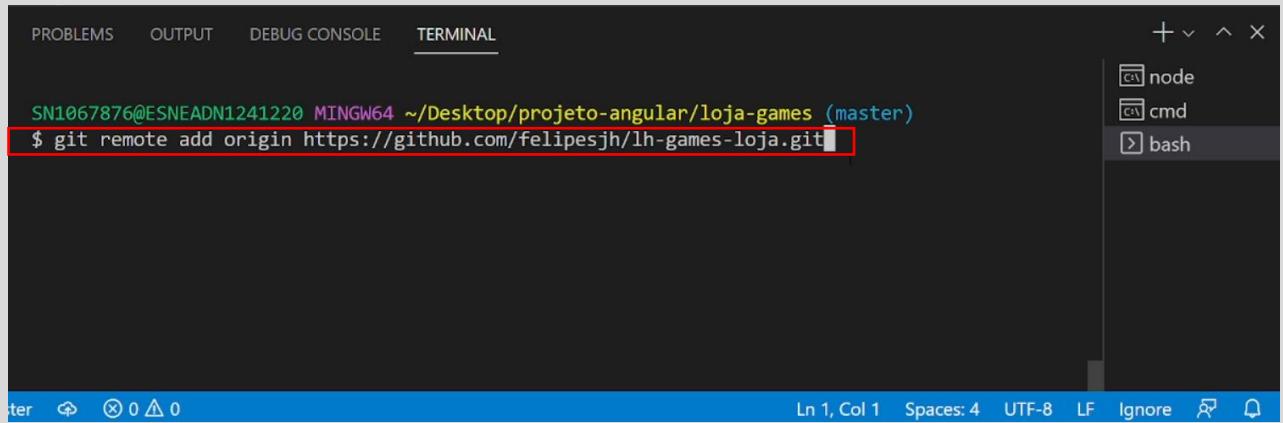
...or create a new repository on the command line

```
echo "# lh-games-loja" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/felipesjh/lh-games-loja.git
git push -u origin main
```

...or push an existing repository from the command line

```
git remote add origin https://github.com/felipesjh/lh-games-loja.git
git branch -M main
git push -u origin main
```

9. Cole a linha copiada no terminal do Git Bash e dê Enter.

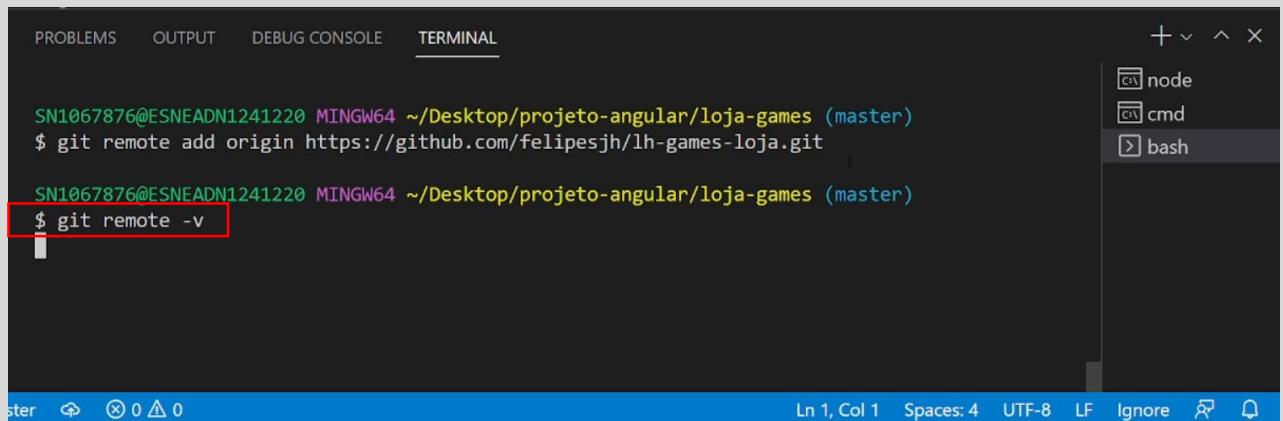


PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
SN1067876@ESNEADN1241220 MINGW64 ~/Desktop/projeto-angular/loja-games (master)
$ git remote add origin https://github.com/felipesjh/lh-games-loja.git
```

Ln 1, Col 1 Spaces: 4 UTF-8 LF Ignore ⚡ 📈

10. Para verificar se deu certo, digite `git remote -v` e dê Enter.



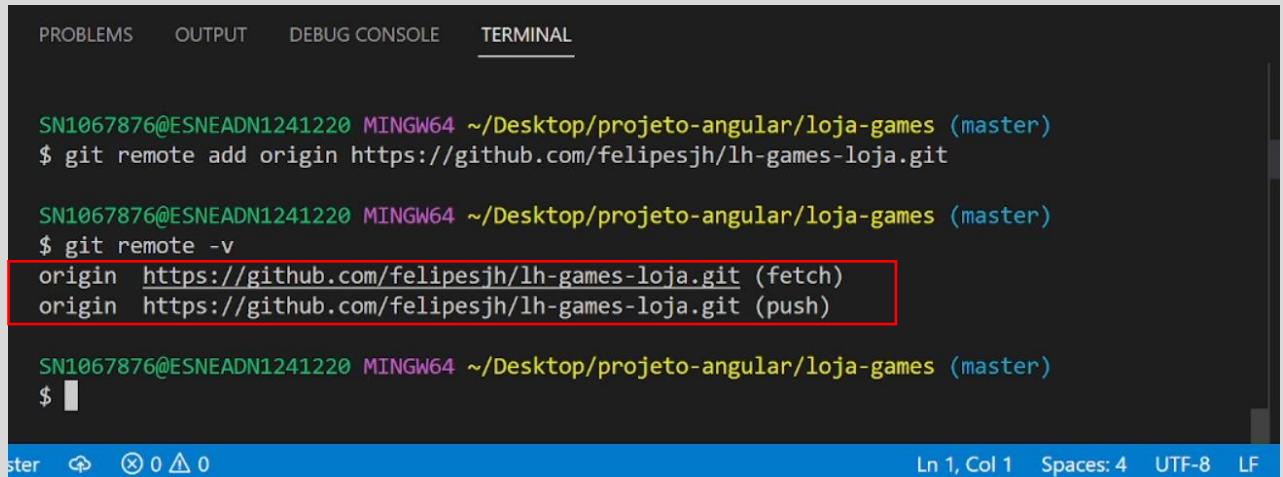
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
SN1067876@ESNEADN1241220 MINGW64 ~/Desktop/projeto-angular/loja-games (master)
$ git remote add origin https://github.com/felipesjh/lh-games-loja.git

SN1067876@ESNEADN1241220 MINGW64 ~/Desktop/projeto-angular/loja-games (master)
$ git remote -v
```

Ln 1, Col 1 Spaces: 4 UTF-8 LF Ignore ⚡ 📈

11. Se o link de origem aparecer com os comandos (**fetch**) e (**push**) está tudo certo.



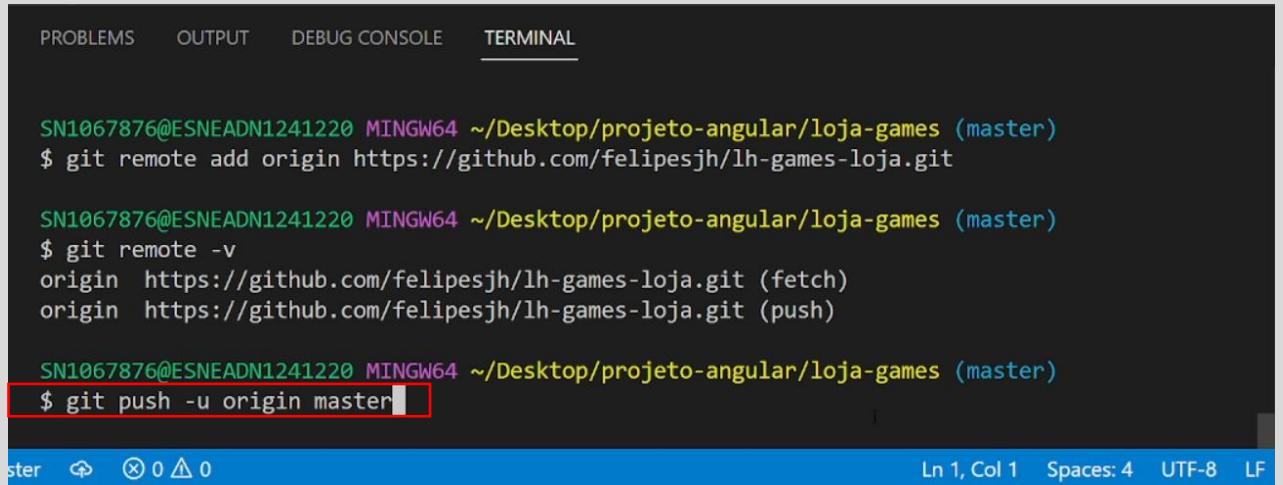
```
SN1067876@ESNEADN1241220 MINGW64 ~/Desktop/projeto-angular/loja-games (master)
$ git remote add origin https://github.com/felipesjh/lh-games-loja.git

SN1067876@ESNEADN1241220 MINGW64 ~/Desktop/projeto-angular/loja-games (master)
$ git remote -v
origin  https://github.com/felipesjh/lh-games-loja.git (fetch)
origin  https://github.com/felipesjh/lh-games-loja.git (push)

SN1067876@ESNEADN1241220 MINGW64 ~/Desktop/projeto-angular/loja-games (master)
$ 
```

ster 0 0 0 Ln 1, Col 1 Spaces: 4 UTF-8 LF

12. Para adicionar a atividade no repositório online, digite **git push -u origin master** e dê Enter.



```
SN1067876@ESNEADN1241220 MINGW64 ~/Desktop/projeto-angular/loja-games (master)
$ git remote add origin https://github.com/felipesjh/lh-games-loja.git

SN1067876@ESNEADN1241220 MINGW64 ~/Desktop/projeto-angular/loja-games (master)
$ git remote -v
origin  https://github.com/felipesjh/lh-games-loja.git (fetch)
origin  https://github.com/felipesjh/lh-games-loja.git (push)

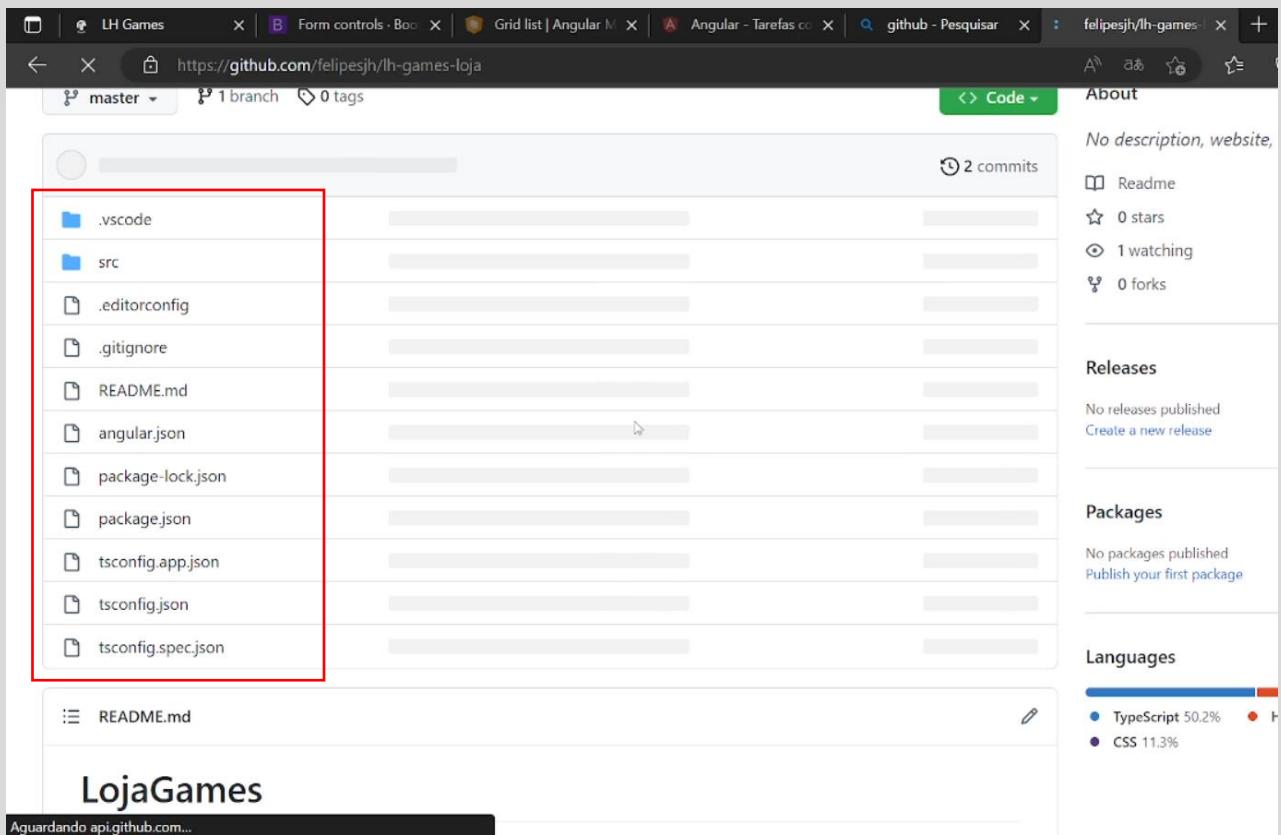
SN1067876@ESNEADN1241220 MINGW64 ~/Desktop/projeto-angular/loja-games (master)
$ git push -u origin master
```

ster 0 0 0 Ln 1, Col 1 Spaces: 4 UTF-8 LF

13. Para conferir se a atividade subiu, retorne ao GitHub e atualize a página.



14. As pastas e arquivos da atividade devem aparecer.



Códigos

src/index.htm

```
<!doctype html>
<html lang="pt-br">
<head>
  <meta charset="utf-8">
  <title>LH Games</title>
  <base href="/">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="icon" type="image/x-icon" href="favicon.ico">
  <link rel="preconnect" href="https://fonts.gstatic.com">
  <link
    href="https://fonts.googleapis.com/css2?family=Roboto:wght@300;400;500&display=swap"
    rel="stylesheet">
    <link href="https://fonts.googleapis.com/icon?family=Material+Icons"
    rel="stylesheet">
</head>
<body>
  <app-root></app-root>
</body>
</html>
```

src/app/app-routing.module.ts

```
import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';
import { InicioComponent } from './inicio/inicio.component';
import { LoginComponent } from './login/login.component';

const routes: Routes = [
  {path:'inicio', component: InicioComponent},
  {path:'login', component: LoginComponent},
  {path:'', redirectTo:'/inicio',pathMatch:'full'}
];

@NgModule({
  imports: [RouterModule.forRoot(routes)],
  exports: [RouterModule]
})
export class AppRoutingModule { }
```

src/app/app.component.html

```
<app-menu></app-menu>
<router-outlet></router-outlet>
<app-rodape></app-rodape>
```

src/app/app.component.ts

```
import { Component } from '@angular/core';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title = 'LH Games';
}
```

src/app/app.module.ts

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';

/*Importações Angular Material*/
import { MatButtonModule} from '@angular/material/button';
import { MatCardModule} from '@angular/material/card';
import { MatFormFieldModule} from '@angular/material/form-field';
import { MatGridListModule} from '@angular/material/grid-list';
import { MatIconModule} from '@angular/material/icon';
import { MatInputModule} from '@angular/material/input';
import { MatMenuModule} from '@angular/material/menu';
import { MatToolbarModule} from '@angular/material/toolbar';

import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { BrowserAnimationsModule } from '@angular/platform-browser/animations';
import { InicioComponent } from './inicio/inicio.component';
import { LoginComponent } from './login/login.component';
import { MenuComponent } from './menu/menu.component';
import { RodapeComponent } from './rodape/rodape.component';

@NgModule({
  declarations: [
    AppComponent,
    InicioComponent,
    LoginComponent,
    MenuComponent,
    RodapeComponent
  ],
  imports: [
    BrowserModule,
    AppRoutingModule,
    BrowserAnimationsModule,
    MatButtonModule,
    MatCardModule,
    MatFormFieldModule,
    MatGridListModule,
    MatIconModule,
    MatInputModule,
    MatMenuModule,
    MatToolbarModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

src/app/inicio/inicio.component.css

```
main{
  min-height: 500px;
}

#section-banner{
  min-height: 400px;
}

#section-banner img{
  height: 500px;
}

h2{
  text-align: center;
  font-size: 30px;
  margin-top: 5px;
}

.example-card {
  max-width: 400px;
}

.example-header-image {
  background-image:
url('https://material.angular.io/assets/img/examples/shiba1.jpg');
  background-size: auto;
}

mat-grid-list{
  margin: 10px 0px;
}

mat-grid-tile{
  padding: 2px;
  border-radius: 30px;
}

mat-card-header{
  display: flex;
  align-items: center;
  justify-content: center;
}

mat-card-actions{
  display: flex;
  justify-content: space-around;
}
```

```
mat-card-actions p{  
  color: red;  
  font-size: 30px;  
}  
  
mat-card img{  
  height: 300px;  
  width: 300px;  
}
```

src/app/inicio/inicio.component.html

```
<main>  
  
  <section id="section-banner">  
    <div id="carouselExampleAutoplaying" class="carousel slide" data-bs-  
ride="carousel">  
      <div class="carousel-inner">  
        <div class="carousel-item active">  
            
        </div>  
        <div class="carousel-item">  
            
        </div>  
        <div class="carousel-item">  
            
        </div>  
      </div>  
      <button class="carousel-control-prev" type="button" data-bs-  
target="#carouselExampleAutoplaying"  
data-bs-slide="prev">  
        <span class="carousel-control-prev-icon" aria-hidden="true"></span>  
        <span class="visually-hidden">Previous</span>  
      </button>  
      <button class="carousel-control-next" type="button" data-bs-  
target="#carouselExampleAutoplaying"  
data-bs-slide="next">  
        <span class="carousel-control-next-icon" aria-hidden="true"></span>  
        <span class="visually-hidden">Next</span>  
      </button>  
    </div>  
  </section>
```

Framework com consumo de API

```
<h2>Jogos em Promoção</h2>
<mat-grid-list cols="3">
  <mat-grid-tile>
    <mat-card class="example-card">
      

      <mat-card-header>
        <mat-card-title>Jogo 1</mat-card-title>
      </mat-card-header>

      <mat-card-content>
        <p>
          Descrição do jogo.
        </p>
      </mat-card-content>
      <mat-card-actions>
        <p>R$ 300,00</p>
        <button mat-button>Comprar</button>
      </mat-card-actions>
    </mat-card>
  </mat-grid-tile>
  <mat-grid-tile>
    <mat-card class="example-card">
      

      <mat-card-header>
        <mat-card-title>Jogo 2</mat-card-title>
      </mat-card-header>

      <mat-card-content>
        <p>
          Descrição do jogo.
        </p>
      </mat-card-content>
      <mat-card-actions>
        <p>R$ 200,00</p>
        <button mat-button>Comprar</button>
      </mat-card-actions>
    </mat-card>
  </mat-grid-tile>
  <mat-grid-tile>
    <mat-card class="example-card">
      

      <mat-card-header>
        <mat-card-title>Jogo 3</mat-card-title>
      </mat-card-header>
```

Framework com consumo de API

```
<mat-card-content>
    <p>Descrição do jogo.</p>
</mat-card-content>
<mat-card-actions>
    <p>R$ 400,00</p>
    <button mat-button>Comprar</button>
</mat-card-actions>
</mat-card>
</mat-grid-tile>
</mat-grid-list>
</main>
```

src/app/login/login.component.css

```
main{
    display: flex;
    justify-content: center;
    align-items: center;
    min-height: 700px;
}
#section-login{
    height: 100%;
    width: 400px;
    border-radius: 10px;
    border: 2px solid black;
    padding: 10px 20px;
}
```

src/app/login/login.component.html

```
<main>
    <section id="section-login">
        <h2>Login</h2>
        <form class="row g-3">
            <label for="staticUsuario" class="visually-hidden">Usuário</label>
            <input type="text" class="form-control" id="staticUsuario"
placeholder="email@example.com">
            <br>
            <label for="inputPassword2" class="visually-hidden">Senha</label>
            <input type="password" class="form-control" id="inputPassword2"
placeholder="senha">
            <br>
            <button type="submit" class="btn btn-primary mb-3">Login</button>
        </form>
    </section>
</main>
```

src/app/menu/menu.component.css

```
.example-spacer {  
  flex: 1 1 auto;  
}  
  
span{  
  padding: 10px;  
}  
  
img{  
  height: 50px;  
  width: 50px;  
}  
  
a{  
  text-decoration: none;  
  color: white;  
}
```

src/app/menu/menu.component.html

```
<mat-toolbar color="primary">  
  <span><a routerLink="inicio"></a></span>  
  <span><a routerLink="inicio">Produtos</a></span>  
  <span class="example-spacer"></span>  
  <span><a routerLink="login">Login</a></span>  
</mat-toolbar>
```

src/app/rodape/rodape.component.css

```
footer{  
  background: black;  
  color: white;  
  text-align: center;  
  height: 100px;  
  padding-top: 30px;  
}
```

src/app/rodape/rodape.component.html

```
<footer>  
  <p>Desenvolvido por Felipe</p>  
</footer>
```