

Curso Angular 9

Leonardo Moura Leitão (Coder)

Vídeos:

<https://www.youtube.com/watch?v=NCrWXZtIc7Q&list=PLdPPE0hUktOrPyAkdhHIIquKbwrGUkvw3>

Curso Grátis e Certificado:

https://bit.ly/cod3r-angular9_gratis

GitHub:

<https://github.com/cod3rcursos/angular-crud>

Aula 01 - Introdução

npm init -y

```
C:\coder\angular\crud\backend>npm init -y
Wrote to C:\coder\angular\crud\backend\package.json:

{
  "name": "backend",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \\"Error: no test specified\\\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}
```

Instalando o pacote json-server

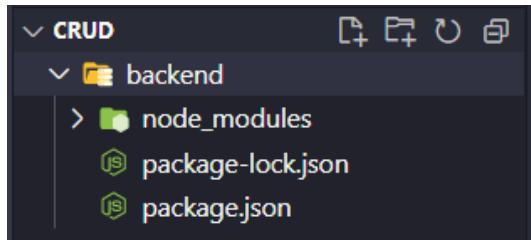
npm install json-server

```
C:\coder\angular\crud\backend>npm install json-server
npm notice created a lockfile as package-lock.json. You should commit this file
-
npm WARN backend@1.0.0 No description
npm WARN backend@1.0.0 No repository field.

+ json-server@0.16.3
added 182 packages from 80 contributors and audited 182 packages in 42.952s

13 packages are looking for funding
  run `npm fund` for details

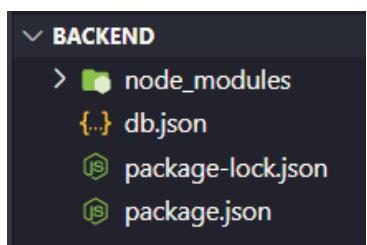
found 0 vulnerabilities
```



backend\package.json

```
{  
  "name": "backend",  
  "version": "1.0.0",  
  "description": "",  
  "main": "index.js",  
  "scripts": {  
    "test": "echo \\\"Error: no test specified\\\" && exit 1"  
  },  
  "keywords": [],  
  "author": "",  
  "license": "ISC",  
  "dependencies": {  
    "json-server": "^0.16.3"  
  }  
}
```

- Na pasta raiz do projeto adicione um arquivo chamado "**db.json**"



db.json

```
{  
  "products": [  
    {  
      "id": 1,  
      "name": "Caneta BIC Preta",  
      "price": 5.89  
    },  
    {  
      "id": 2,  
      "name": "Notebook Mac Pro",  
      "price": 12000.89  
    },  
    {  
      "id": 3,  
      "name": "Samsung S20+",  
      "price": 5000.89  
    }  
  ]  
}
```

package.json

```
{  
  "name": "backend",  
  "version": "1.0.0",  
  "description": "",  
  "main": "index.js",  
  "scripts": {  
    "test": "echo \\\"Error: no test specified\\\" && exit 1",  
    "start": "json-server --watch db.json --port 3001"  
  },  
  "keywords": [],  
  "author": "",  
  "license": "ISC",  
  "dependencies": {  
    "json-server": "^0.16.3"  
  }  
}
```

```
npm start
```

```
C:\coder\angular\crud\backend>npm start
> backend@1.0.0 start C:\coder\angular\crud\backend
> json-server --watch db.json --port 3001

\{^_^\}/ hi!

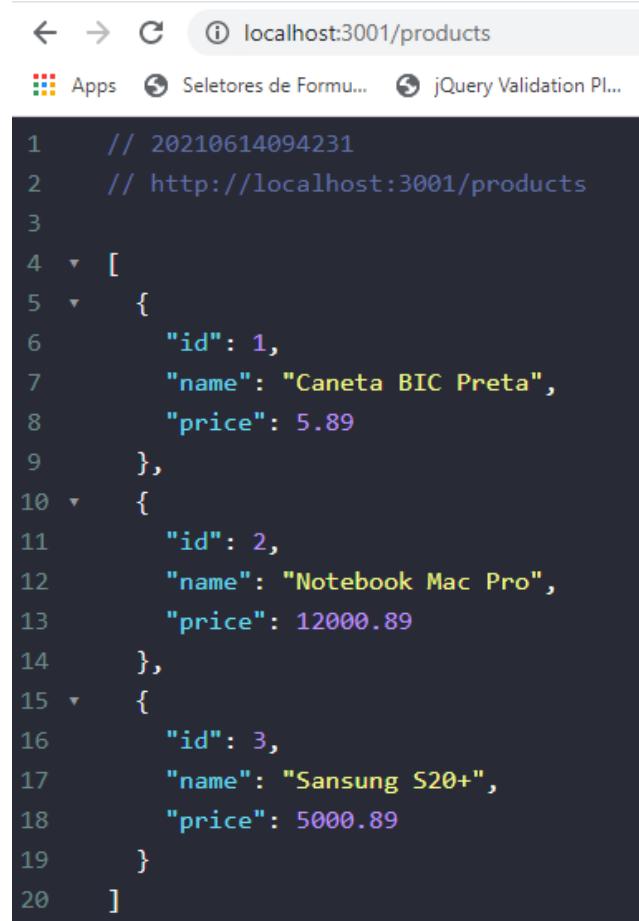
Loading db.json
Done

Resources
http://localhost:3001/products
Home
http://localhost:3001

Type s + enter at any time to create a snapshot of the database
Watching...
```

- No browser:

<http://localhost:3001/products>



A screenshot of a browser window displaying the JSON data at <http://localhost:3001/products>. The browser's address bar shows the URL. Below the address bar, there are tabs for 'Apps', 'Seletores de Formu...', and 'jQuery Validation Pl...'. The main content area displays the following JSON array:

```
1 // 20210614094231
2 // http://localhost:3001/products
3
4 [
5   {
6     "id": 1,
7     "name": "Caneta BIC Preta",
8     "price": 5.89
9   },
10  {
11    "id": 2,
12    "name": "Notebook Mac Pro",
13    "price": 12000.89
14  },
15  {
16    "id": 3,
17    "name": "Samsung S20+",
18    "price": 5000.89
19  }
20 ]
```

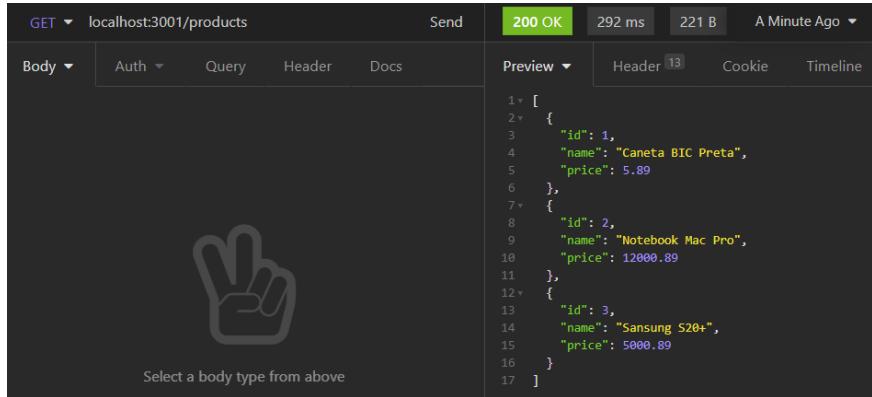
- No Insomnia:

Listando produtos

GET - localhost:3001/products

GET ▾ localhost:3001/products Send **200 OK** 292 ms 221 B A Minute Ago ▾

Body ▾ Auth ▾ Query Header Docs Preview ▾ Header 13 Cookie Timeline



```
1+ [
2+   {
3+     "id": 1,
4+     "name": "Caneta BIC Preta",
5+     "price": 5.89
6+   },
7+   {
8+     "id": 2,
9+     "name": "Notebook Mac Pro",
10+    "price": 12000.89
11+   },
12+   {
13+     "id": 3,
14+     "name": "Samsung S20+",
15+     "price": 5000.89
16+   }
17+ ]
```

Select a body type from above

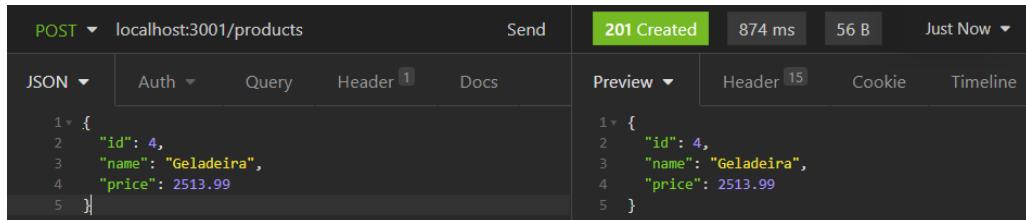
Cadastrando um produto

- No Insomnia:

POST - localhost:3001/products

POST ▾ localhost:3001/products Send **201 Created** 874 ms 56 B Just Now ▾

JSON ▾ Auth ▾ Query Header 1 Docs Preview ▾ Header 15 Cookie Timeline



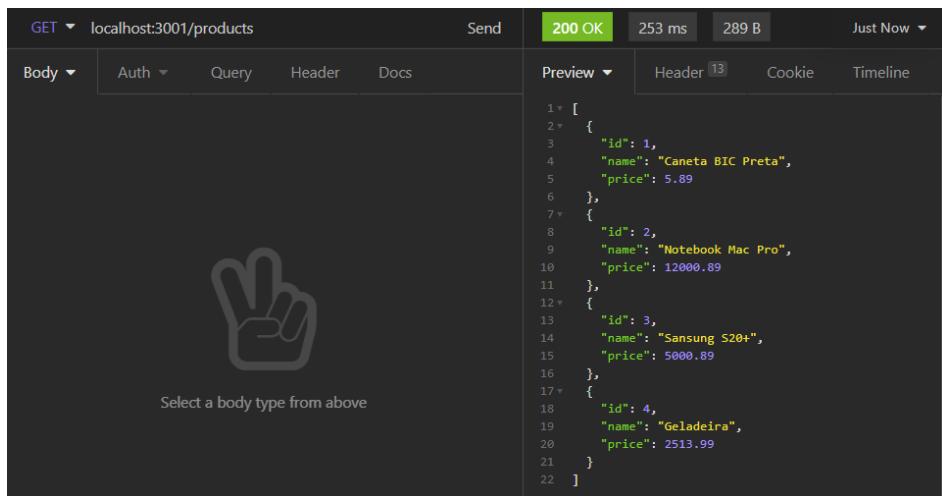
```
1+ {
2+   "id": 4,
3+   "name": "Geladeira",
4+   "price": 2513.99
5+ }
```

```
1+ {
2+   "id": 4,
3+   "name": "Geladeira",
4+   "price": 2513.99
5+ }
```

GET - localhost:3001/products

GET ▾ localhost:3001/products Send **200 OK** 253 ms 289 B Just Now ▾

Body ▾ Auth ▾ Query Header Docs Preview ▾ Header 13 Cookie Timeline



```
1+ [
2+   {
3+     "id": 1,
4+     "name": "Caneta BIC Preta",
5+     "price": 5.89
6+   },
7+   {
8+     "id": 2,
9+     "name": "Notebook Mac Pro",
10+    "price": 12000.89
11+   },
12+   {
13+     "id": 3,
14+     "name": "Samsung S20+",
15+     "price": 5000.89
16+   },
17+   {
18+     "id": 4,
19+     "name": "Geladeira",
20+     "price": 2513.99
21+   }
22+ ]
```

Select a body type from above

Editando um produto

PUT - localhost:3001/products/4

PUT ▾	localhost:3001/products/4	Send	200 OK	303 ms	58 B	Just Now ▾		
JSON ▾	Auth ▾	Query	Header 13	Docs	Preview ▾	Header 13	Cookie	Timeline
1 * { 2 "id": 4, 3 "name": "Refrigerador", 4 "price": 2721.90 5 }				1 * { 2 "id": 4, 3 "name": "Refrigerador", 4 "price": 2721.9 5 }				

GET - localhost:3001/products

GET ▾	localhost:3001/products	Send	200 OK	226 ms	291 B	Just Now ▾		
Body ▾	Auth ▾	Query	Header	Docs	Preview ▾	Header 13	Cookie	Timeline
 Select a body type from above					1 * [2 * { 3 "id": 1, 4 "name": "Caneta BIC Preta", 5 "price": 5.89 6 }, 7 * { 8 "id": 2, 9 "name": "Notebook Mac Pro", 10 "price": 12000.89 11 }, 12 * { 13 "id": 3, 14 "name": "Samsung S20+", 15 "price": 5000.89 16 }, 17 * { 18 "id": 4, 19 "name": "Refrigerador", 20 "price": 2721.9 21 } 22]			

Excluindo um produto

DELETE - localhost:3001/products/4

DELETE ▾	localhost:3001/products/4	Send	200 OK	247 ms	2 B	Just Now ▾		
Body ▾	Auth ▾	Query	Header	Docs	Preview ▾	Header 13	Cookie	Timeline
					1 * {}			

GET - localhost:3001/products

GET ▾	localhost:3001/products	Send	200 OK	254 ms	221 B	Just Now ▾		
Body ▾	Auth ▾	Query	Header	Docs	Preview ▾	Header 13	Cookie	Timeline
<pre>1 [2 { 3 "id": 1, 4 "name": "Caneta BIC Preta", 5 "price": 5.89 6 }, 7 { 8 "id": 2, 9 "name": "Notebook Mac Pro", 10 "price": 12000.89 11 }, 12 { 13 "id": 3, 14 "name": "Samsung S20+", 15 "price": 5000.89 16 } 17]</pre>								
Select a body type from above								

Aula 02 - Visão geral Angular

O que é o Angular?

É um framework JavaScript desenvolvido pelo Google para criação de aplicações Web SPA baseada em componentes.

Instalando o Command Line Interface (CLI)

```
npm i -g @angular/cli
```

Criando um aplicação

```
ng new <nome-app>
```

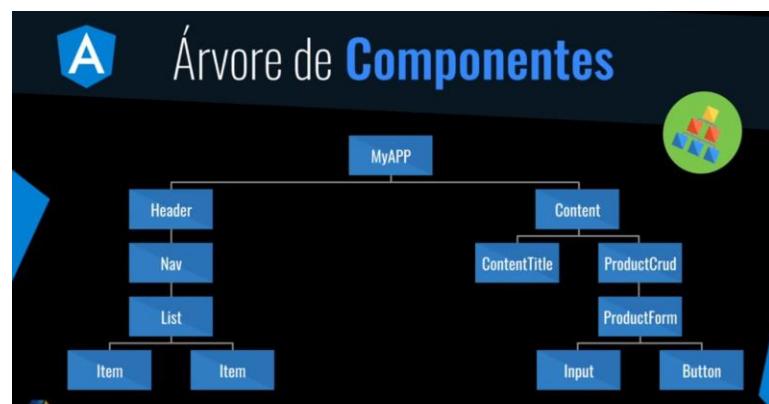
TypeScript

- Linguagem criada pela Microsoft.
- O código escrito em TypeScript é compilado para JavaScript.
- Orientada a objetos e possui tipagem forte.
- Superset do JavaScript
- Possui funcionalidades adicionais ao JavaScript

Outros recursos da linguagem:

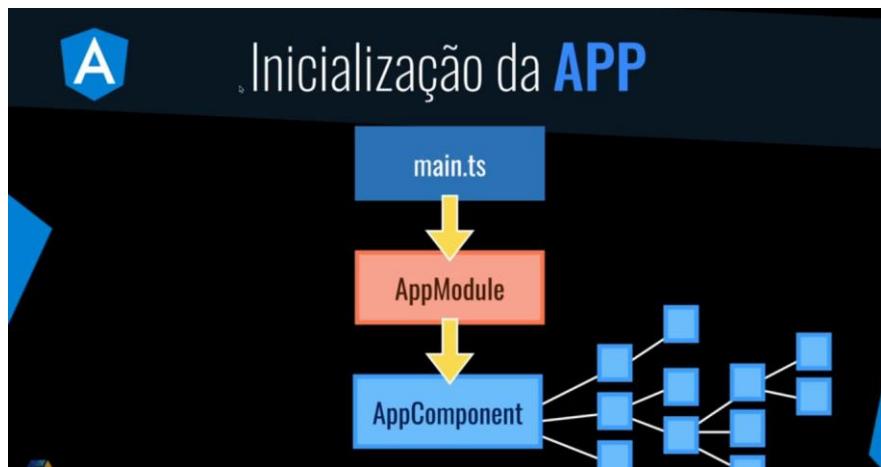
- decorator
- interface

Árvore de componentes



Aula 03 - Conceitos essenciais do Angular

Fluxo da Inicialização da APP



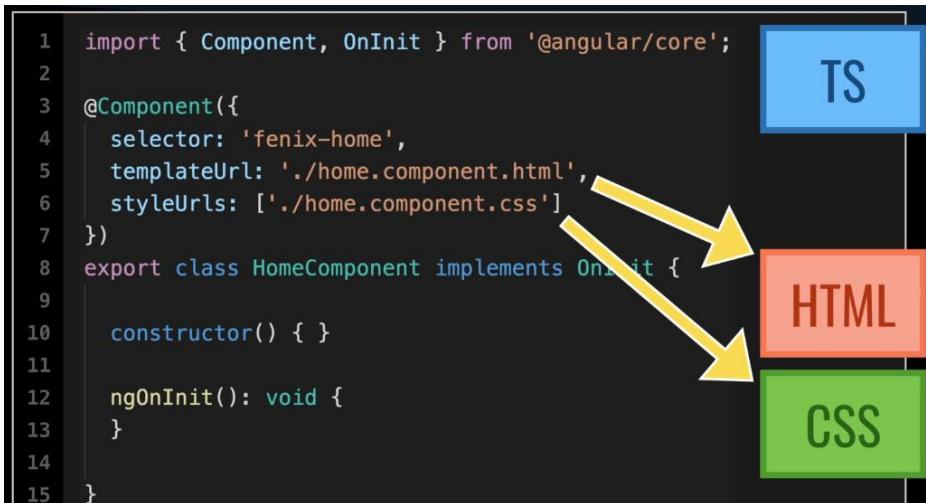
O que é um componente?

Exemplo: Componente home:



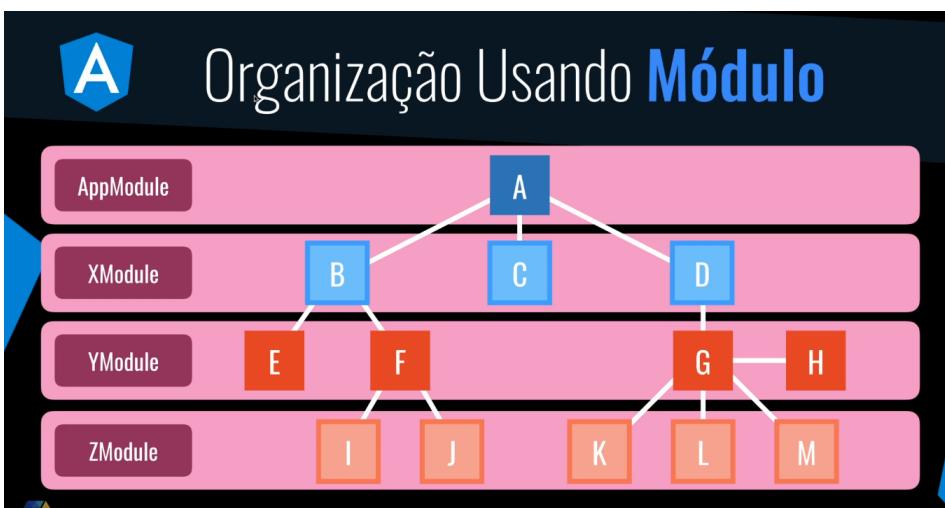
tag personalizada:

```
<app-home></app-home>
```

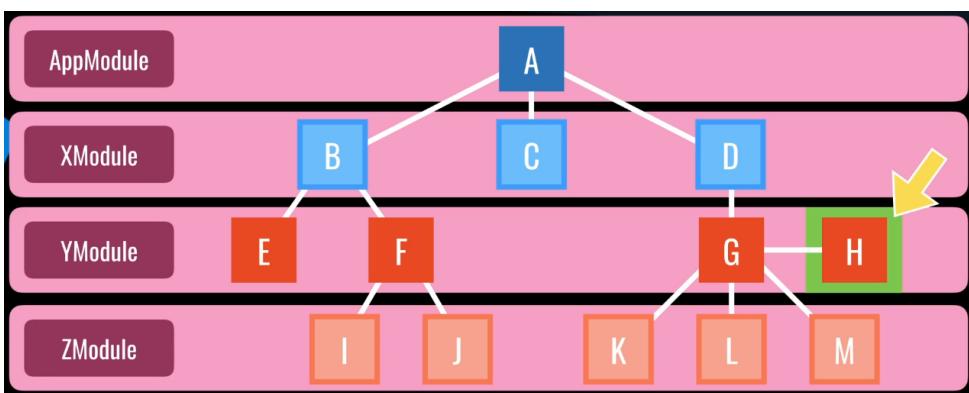


Organização usando módulo

- Componentes ficam dentro de módulos.

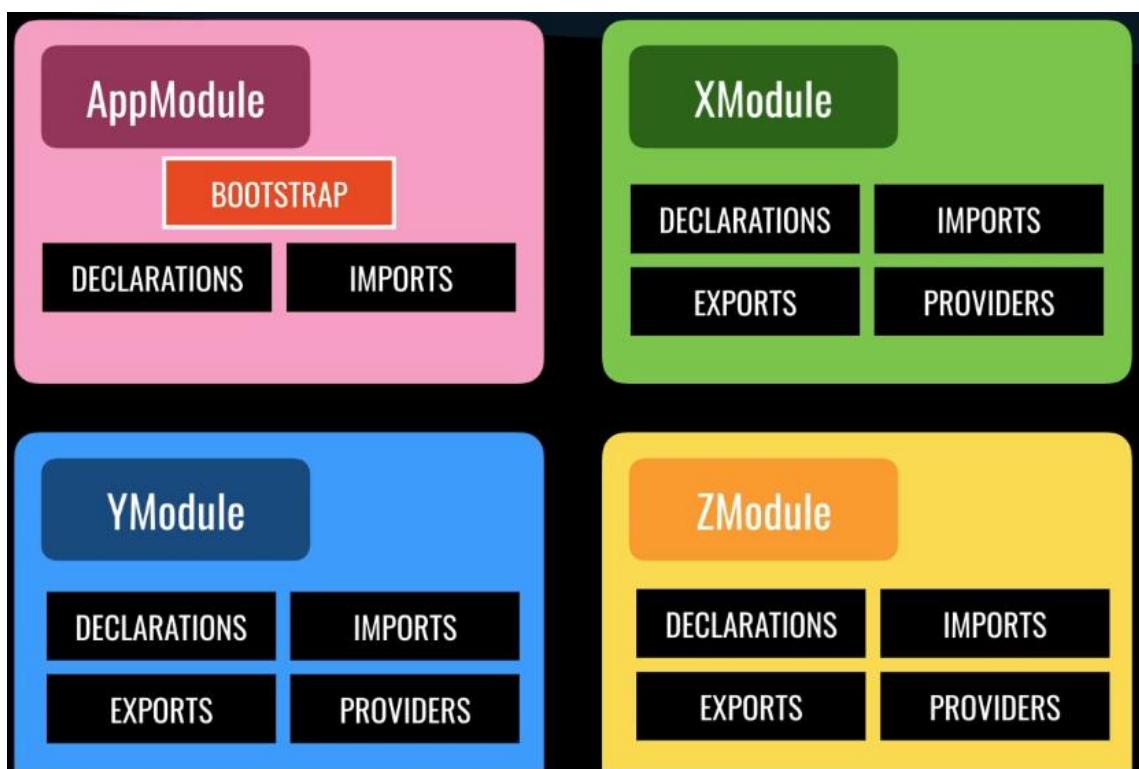
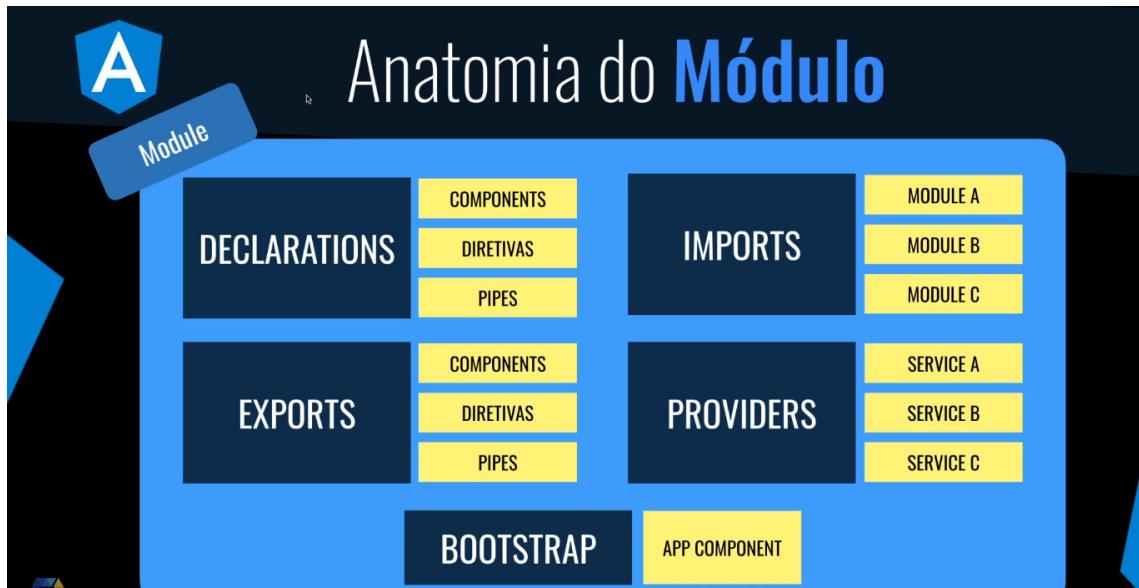


- O componente H é visível apenas dentro do módulo (YModule):



Anatomia do módulo

O Angular possui 5 atributos dentro do módulo: declarations, imports, exports, provider e bootstrap.



- O AppModule não precisa exportar para nenhum outro módulo.

Aula 04 - Criando app com o Angular CLI

Instalando o pacote angular/cli

```
npm i -g @angular/cli
```

- Execute este comando na pasta CRUD:

```
C:\coder\angular\crud>npm i -g @angular/cli
npm WARN deprecated request@2.88.2: request has been deprecated, see https://github
.com/request/request/issues/3142
npm WARN deprecated har-validator@5.1.5: this library is no longer supported
npm WARN deprecated uuid@3.4.0: Please upgrade to version 7 or higher. Older vers
ions may use Math.random() in certain circumstances, which is known to be problemat
ic. See https://v8.dev/blog/math-random for details.
C:\Users\beto1\AppData\Roaming\npm\ng -> C:\Users\beto1\AppData\Roaming\npm\node_mo
dules\@angular\cli\bin\ng

> @angular/cli@12.0.4 postinstall C:\Users\beto1\AppData\Roaming\npm\node_modules\@
angular\cli
> node ./bin/postinstall/script.js

? Would you like to share anonymous usage data with the Angular Team at Google unde
r
Google's Privacy Policy at https://policies.google.com/privacy? For more details an
d
how to change this setting, see https://angular.io/analytics. No
+ @angular/cli@12.0.4
added 234 packages from 179 contributors in 217.377s
```

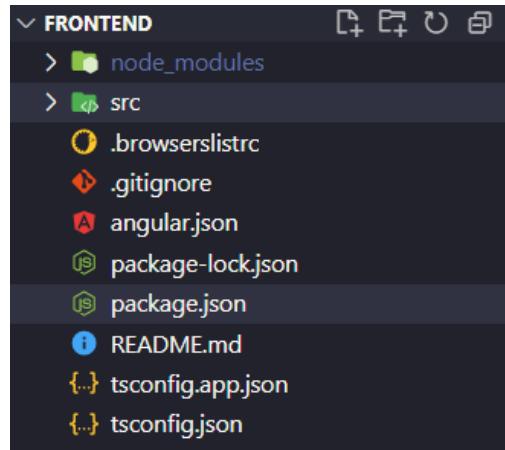
Criando um novo projeto

ng new frontend --minimal

```
C:\coder\angular\crud>ng new frontend --minimal
? Would you like to add Angular routing? Yes
? Which stylesheet format would you like to use? CSS
CREATE Frontend/angular.json (3188 bytes)
CREATE Frontend/package.json (815 bytes)
CREATE Frontend/README.md (1054 bytes)
CREATE Frontend/tsconfig.json (783 bytes)
CREATE Frontend/.gitignore (604 bytes)
CREATE Frontend/.browserslistrc (703 bytes)
CREATE Frontend/tsconfig.app.json (287 bytes)
CREATE Frontend/src/favicon.ico (948 bytes)
CREATE Frontend/src/index.html (294 bytes)
CREATE Frontend/src/main.ts (372 bytes)
CREATE Frontend/src/polyfills.ts (2820 bytes)
CREATE Frontend/src/styles.css (80 bytes)
CREATE Frontend/src/assets/.gitkeep (0 bytes)
CREATE Frontend/src/environments/environment.prod.ts (51 bytes)
CREATE Frontend/src/environments/environment.ts (658 bytes)
CREATE Frontend/src/app/app-routing.module.ts (245 bytes)
CREATE Frontend/src/app/app.module.ts (393 bytes)
CREATE Frontend/src/app/app.component.ts (1483 bytes)
/ Installing packages (npm)...

warning: LF will be replaced by CRLF in .browserslistrc.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in .gitignore.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in README.md.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in angular.json.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in package-lock.json.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in package.json.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/app/app-routing.module.ts.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/app/app.component.ts.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/app/app.module.ts.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/environments/environment.prod.ts.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/environments/environment.ts.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/index.html.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/main.ts.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/polyfills.ts.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/styles.css.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in tsconfig.app.json.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in tsconfig.json.
The file will have its original line endings in your working directory
  Successfully initialized git.
```

```
C:\coder\angular\crud>cd frontend  
C:\coder\angular\crud\frontend>code .
```



package.json

```
{
  "name": "frontend",
  "version": "0.0.0",
  "scripts": {
    "ng": "ng",
    "start": "ng serve",
    "build": "ng build",
    "watch": "ng build --watch --configuration development"
  },
  "private": true,
  "dependencies": {
    "@angular/animations": "~12.0.4",
    "@angular/common": "~12.0.4",
    "@angular/compiler": "~12.0.4",
    "@angular/core": "~12.0.4",
    "@angular/forms": "~12.0.4",
    "@angular/platform-browser": "~12.0.4",
    "@angular/platform-browser-dynamic": "~12.0.4",
    "@angular/router": "~12.0.4",
    "rxjs": "~6.6.0",
    "tslib": "^2.1.0",
    "zone.js": "~0.11.4"
  },
  "devDependencies": {
    "@angular-devkit/build-angular": "~12.0.4",
    "@angular/cli": "~12.0.4",
    "@angular/compiler-cli": "~12.0.4",
    "@types/node": "^12.11.1",
    "typescript": "~4.2.3"
  }
}
```

- No terminal, entre com o comando:

npm start

```
C:\coder\angular\crud\frontend>npm start

> frontend@0.0.0 start C:\coder\angular\crud\frontend
> ng serve

  Generating browser application bundles (phase: setup)...Compiling @angular/core : es2015 as esm2015
Compiling @angular/common : es2015 as esm2015
Compiling @angular/platform-browser : es2015 as esm2015
Compiling @angular/platform-browser-dynamic : es2015 as esm2015
Compiling @angular/router : es2015 as esm2015
✓ Browser application bundle generation complete.

Initial Chunk Files | Names           | Size
vendor.js           | vendor          | 2.38 MB
polyfills.js        | polyfills       | 508.81 kB
styles.css, styles.js | styles          | 381.00 kB
main.js              | main            | 12.85 kB
runtime.js          | runtime         | 6.57 kB

| Initial Total | 3.26 MB

Build at: 2021-06-15T00:20:53.848Z - Hash: 33b97f8b6f874da59ba5 - Time: 79174ms

** Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **

  ✓ Compiled successfully.
  ✓ Browser application bundle generation complete.

5 unchanged chunks

Build at: 2021-06-15T00:20:58.384Z - Hash: ebc9be96fa109082ed72 - Time: 2296ms

  ✓ Compiled successfully.
```

- Abra dois novos terminais: um para rodar o servidor backend e outro para digitarmos os comandos:

npm start

```
C:\coder\angular\crud\backend>npm start

> backend@1.0.0 start C:\coder\angular\crud\backend
> json-server --watch db.json --port 3001

  \{^_^\}/ hi!

  Loading db.json
  Done

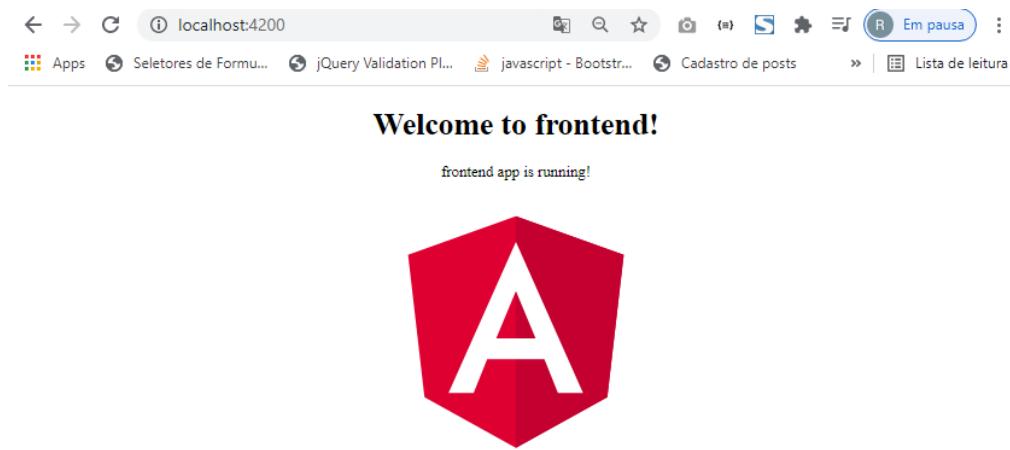
  Resources
  http://localhost:3001/products

  Home
  http://localhost:3001

  Type s + enter at any time to create a snapshot of the database
  Watching...
```

- No browser:

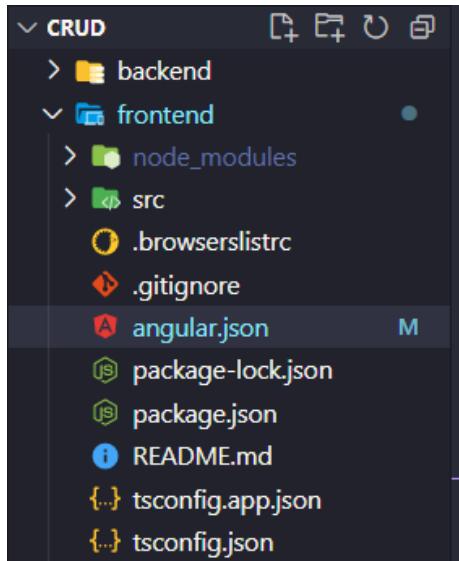
localhost:4200



Here are some links to help you start:

- [Tour of Heroes](#)
- [CLI Documentation](#)
- [Angular blog](#)

Aula 05 - Conhecendo os arquivos do projeto

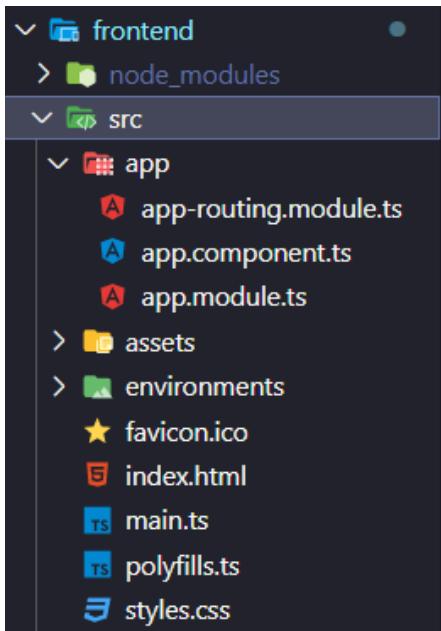


frontend\angular.json

```
{
  "$schema": "./node_modules/@angular/cli/lib/config/schema.json",
  "version": 1,
  "newProjectRoot": "projects",
  "projects": {
    "frontend": {
      "projectType": "application",
      "schematics": {
        "@schematics/angular:component": {
          "inlineTemplate": false,
          "inlineStyle": false,
          "skipTests": true
        },
        "@schematics/angular:class": {
          "skipTests": true
        },
        "@schematics/angular:directive": {
          "skipTests": true
        },
        "@schematics/angular:guard": {
          "skipTests": true
        },
        "@schematics/angular:interceptor": {
          "skipTests": true
        },
        "@schematics/angular:pipe": {
          "skipTests": true
        },
        "@schematics/angular:service": {
          "skipTests": true
        }
      }
    }
  }
}
```

```
"@schematics/angular:application": {
  "strict": true
},
"root": "",
"sourceRoot": "src",
"prefix": "app",
"architect": {
  "build": {
    "builder": "@angular-devkit/build-angular:browser",
    "options": {
      "outputPath": "dist/frontend",
      "index": "src/index.html",
      "main": "src/main.ts",
      "polyfills": "src/polyfills.ts",
      "tsConfig": "tsconfig.app.json",
      "assets": [
        "src/favicon.ico",
        "src/assets"
      ],
      "styles": [
        "src/styles.css"
      ],
      "scripts": []
    },
    "configurations": {
      "production": {
        "budgets": [
          {
            "type": "initial",
            "maximumWarning": "500kb",
            "maximumError": "1mb"
          },
          {
            "type": "anyComponentStyle",
            "maximumWarning": "2kb",
            "maximumError": "4kb"
          }
        ],
        "fileReplacements": [
          {
            "replace": "src/environments/environment.ts",
            "with": "src/environments/environment.prod.ts"
          }
        ],
        "outputHashing": "all"
      },
      "development": {
        "buildOptimizer": false,
        "optimization": false,
        "vendorChunk": true,
        "extractLicenses": false,
        "sourceMap": true,
        "namedChunks": true
      }
    },
    "defaultConfiguration": "production"
  }
}
```

```
        },
        "serve": {
          "builder": "@angular-devkit/build-angular:dev-server",
          "configurations": {
            "production": {
              "browserTarget": "frontend:build:production"
            },
            "development": {
              "browserTarget": "frontend:build:development"
            }
          },
          "defaultConfiguration": "development"
        },
        "extract-i18n": {
          "builder": "@angular-devkit/build-angular:extract-i18n",
          "options": {
            "browserTarget": "frontend:build"
          }
        }
      }
    },
    "defaultProject": "frontend"
  }
}
```



- Em "main.ts" é chamado o módulo que vai inicializar a aplicação (**AppModule**):

frontend\src\main.ts

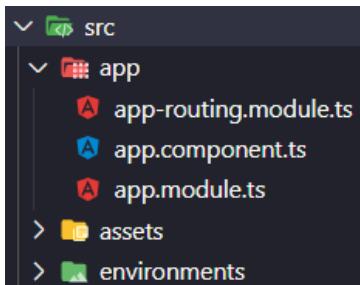
```
import { enableProdMode } from '@angular/core';
import { platformBrowserDynamic } from '@angular/platform-browser-dynamic';

import { AppModule } from './app/app.module';
import { environment } from './environments/environment';

if (environment.production) {
  enableProdMode();
}

platformBrowserDynamic().bootstrapModule(AppModule)
  .catch(err => console.error(err));
```

- Dentro de "frontend\src\app" temos os arquivos do módulo criado:



frontend\src\app\app.component.ts

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

@Component({
  selector: 'app-root',
  template: `
    <!--The content below is only a placeholder and can be replaced.-->
    <div style="text-align:center" class="content">
      <h1>
        Welcome to {{title}}!
      </h1>
      <span style="display: block">{{ title }} app is running!</span>
      
    </div>
    <h2>Here are some links to help you start: </h2>
    <ul>
      <li>
        <h2><a target="_blank" rel="noopener" href="https://angular.io/tutorial">Tour of Heroes</a></h2>
      </li>
      <li>
        <h2><a target="_blank" rel="noopener" href="https://angular.io/cli">CLI Documentation</a></h2>
      </li>
      <li>
        <h2><a target="_blank" rel="noopener" href="https://blog.angular.io/">Angular blog</a></h2>
      </li>
    </ul>
    <router-outlet></router-outlet>
  `,
  styles: []
})
export class AppComponent {
  title = 'frontend';
}
```

- Em "bootstrap" é definido qual é o componente que será inicializado para carregar a aplicação:

frontend\src\app\app.module.ts

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

import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';

@NgModule({
  declarations: [
    AppComponent
  ],
  imports: [
    BrowserModule,
    AppRoutingModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

- Na pasta "assets" é permitido colocar, por exemplo, uma imagem ou um arquivo de fonte.
- A pasta "environments" é o local para inserir variáveis de ambiente.
- Por ser uma SPA (Single Page Application), O arquivo "index.html" aponta para o documento raiz de nossa aplicação.

frontend\src\index.html

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Frontend</title>
  <base href="/">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="icon" type="image/x-icon" href="favicon.ico">
</head>
<body>
  <app-root></app-root>
</body>
</html>
```

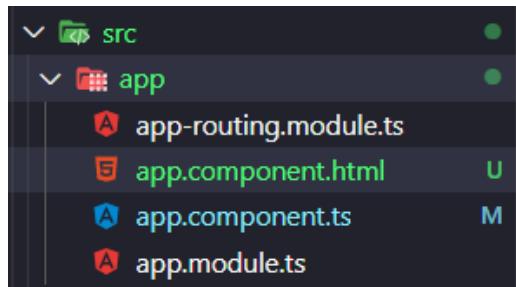
Aula 06 - HTML do AppComponent

frontend\src\app\app.component.ts

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

@Component({
  selector: 'app-root',
  templateUrl: 'app.component.html',
})
export class AppComponent {
  name = 'Maria';
}
```

- Adicione na pasta "frontend\src\app" o arquivo "app.component.html":



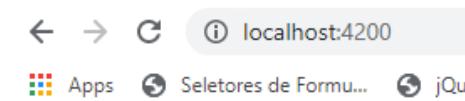
frontend\src\app\app.component.html

```
<h1>Componente APP</h1>
<h2>Seja bem-vinda, {{ name }}!</h2>
```

- Interrompa e reinicialize o servidor do Angular

- No browser:

localhost:4200



Componente APP

Seja bem-vinda, Maria!

Aula 07 - Instalar os componentes do Material

Instalando o Material

- Inicialmente interrompa a execução do servidor do Angular. Em seguida entre com o comando:

```
ng add @angular/material
```

```
c:\coder\angular\crud\frontend>ng add @angular/material
i  Using package manager: npm
✓ Found compatible package version: @angular/material@12.0.4.
✓ Package information loaded.

The package @angular/material@12.0.4 will be installed and executed.
Would you like to proceed? Yes
✓ Package successfully installed.
? Choose a prebuilt theme name, or "custom" for a custom theme: Indigo/Pink
  [ Preview: https://material.angular.io?theme=indigo-pink ]
? Set up global Angular Material typography styles? Yes
? Set up browser animations for Angular Material? Yes
UPDATE package.json (881 bytes)
✓ Packages installed successfully.

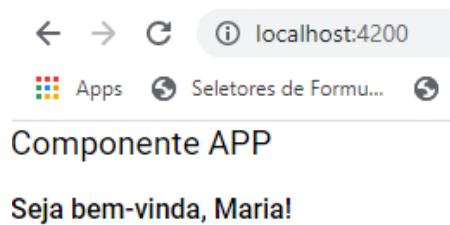
Your project is not using the default builders for "test". This means that we
cannot add the configured theme to the "test" target.
UPDATE src/app/app.module.ts (502 bytes)
UPDATE angular.json (3271 bytes)
UPDATE src/index.html (576 bytes)
UPDATE src/styles.css (181 bytes)
```

- Volte a rodar o servidor do Angular:

```
npm start
```

- No browser:

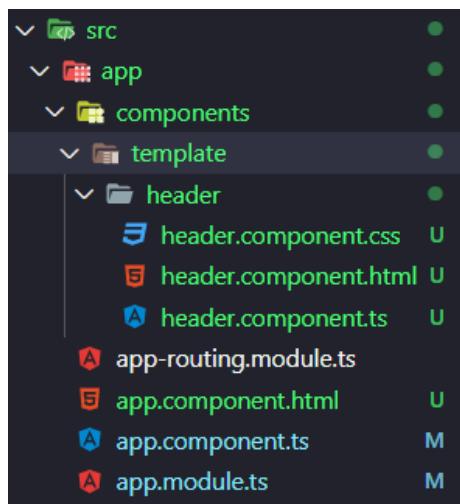
localhost:4200



Aula 08 - Componente Cabeçalho

ng generate component components/template/header

```
C:\coder\angular\crud\frontend>ng generate component components/template/header
CREATE  src/app/components/template/header/header.component.html (21 bytes)
CREATE  src/app/components/template/header/header.component.ts (275 bytes)
CREATE  src/app/components/template/header/header.component.css (0 bytes)
UPDATE  src/app/app.module.ts (604 bytes)
```



frontend\src\app\app.module.ts

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

import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { BrowserAnimationsModule } from '@angular/platform-browser/animations';
import { HeaderComponent } from './components/template/header/header.component';

@NgModule({
  declarations: [
    AppComponent,
    HeaderComponent
  ],
  imports: [
    BrowserModule,
    AppRoutingModule,
    BrowserAnimationsModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule {}
```

frontend\src\app\app.component.html

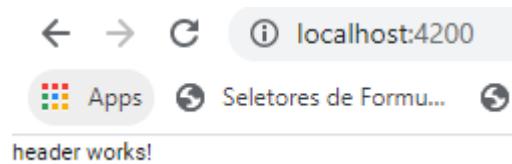
```
<app-header></app-header>
```

frontend\src\app\components\template\header\header.component.html

```
<p>header works!</p>
```

- No browser:

localhost:4200

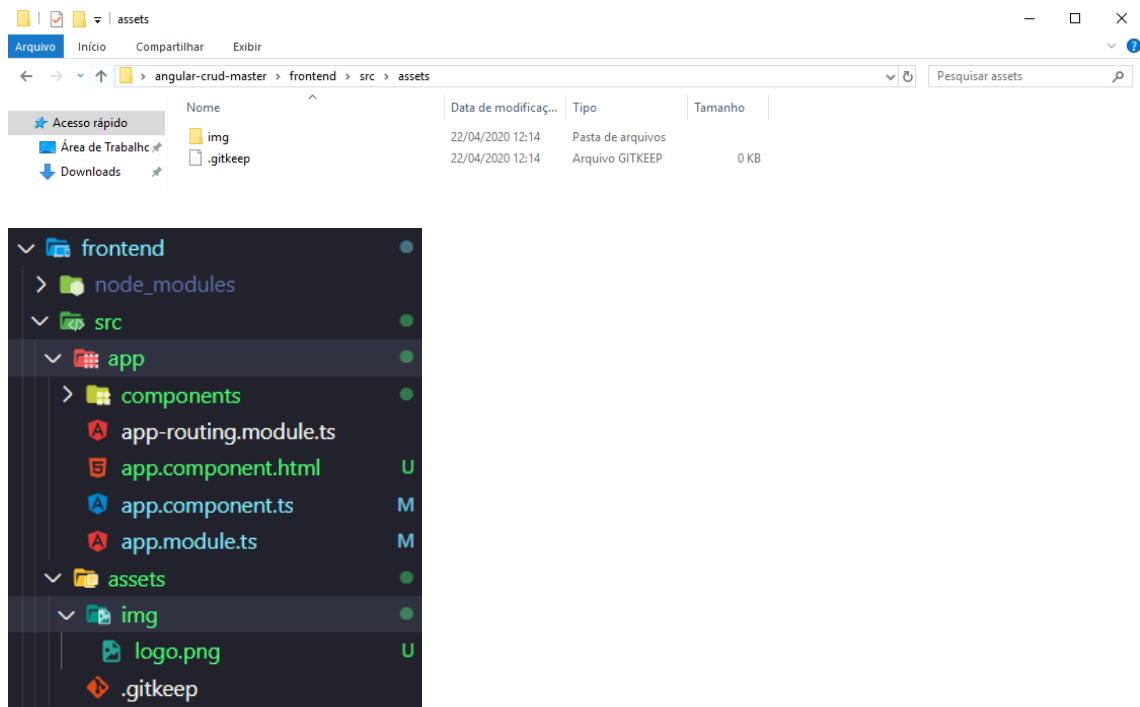


frontend\src\app\app.module.ts

```
import { NgModule } from '@angular/core';
import { MatToolbarModule } from '@angular/material/toolbar';
import { BrowserModule } from '@angular/platform-browser';
import { BrowserAnimationsModule } from '@angular/platform-browser/animations';
import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { HeaderComponent } from './components/template/header/header.component';

@NgModule({
  declarations: [AppComponent, HeaderComponent],
  imports: [
    BrowserModule,
    AppRoutingModule,
    BrowserAnimationsModule,
    MatToolbarModule
  ],
  providers: [],
  bootstrap: [AppComponent],
})
export class AppModule {}
```

- Do arquivo baixado, copie a pasta "C:\Users\beto1\Desktop\angular-crud-master\frontend\src\assets\img" para dentro da pasta "frontend\src\assets"



frontend\src\app\app.component.html

```
<app-header></app-header>
```

Observação: No Visual Studio Code pressione **<alt><shift><f>** para formatar o código.

frontend\src\app\components\template\header\header.component.css

```
.header {
  display: flex;
  align-items: center;
}

.header a {
  display: flex;
  align-items: center;
  text-decoration: none;
}

.header .logo {
  max-height: 32px;
}

.header .title-group {
  padding-left: 250px;
}

.header .title-group i {
  padding-right: 5px;
}
```

- No browser:

localhost:4200

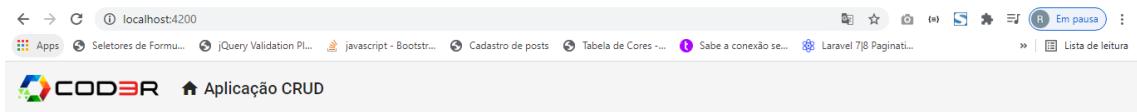


src\app\components\template\header\header.component.html

```
<mat-toolbar class="header mat-elevation-z4">
  <span>
    <a>
      
    </a>
  </span>
  <span class="title-group">
    <a>
      <!-- <i class="material-icons"> dashboard </i> -->
      <i class="material-icons"> home </i>
      Aplicação CRUD
    </a>
  </span>
</mat-toolbar>
```

- No browser:

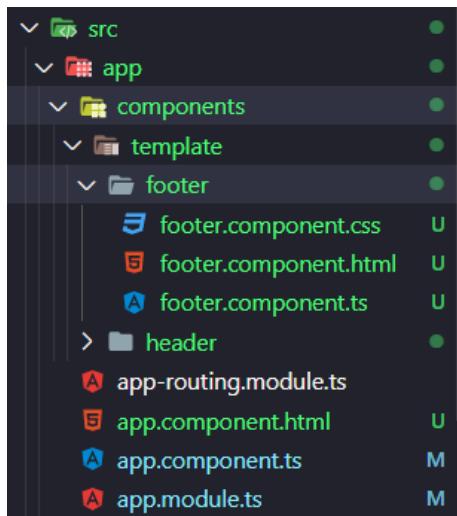
localhost:4200



Aula 09 - Componente Rodapé

ng generate component components/template/footer

```
C:\coder\angular\crud\frontend>ng generate component components/template/footer
CREATE  src/app/components/template/footer/footer.component.html (21 bytes)
CREATE  src/app/components/template/footer/footer.component.ts (275 bytes)
CREATE  src/app/components/template/footer/footer.component.css (0 bytes)
UPDATE  src/app/app.module.ts (774 bytes)
```



frontend\src\app\app.component.html

```
<app-header></app-header>
<app-footer></app-footer>
```

src\app\components\template\footer\footer.component.html

```
<mat-toolbar class="footer">
  <span>
    Desenvolvido com
    <i class="material-icons red v-middle"> favorite </i>
    por <strong>Cod<span class="red">3</span>r</strong>
  </span>
</mat-toolbar>
```

src\styles.css

```
/* You can add global styles to this file, and also import other style files */

html,
body {
  height: 100%;
}
body {
  margin: 0;
  font-family: Roboto, 'Helvetica Neue', sans-serif;
}

.red {
  color: red;
}

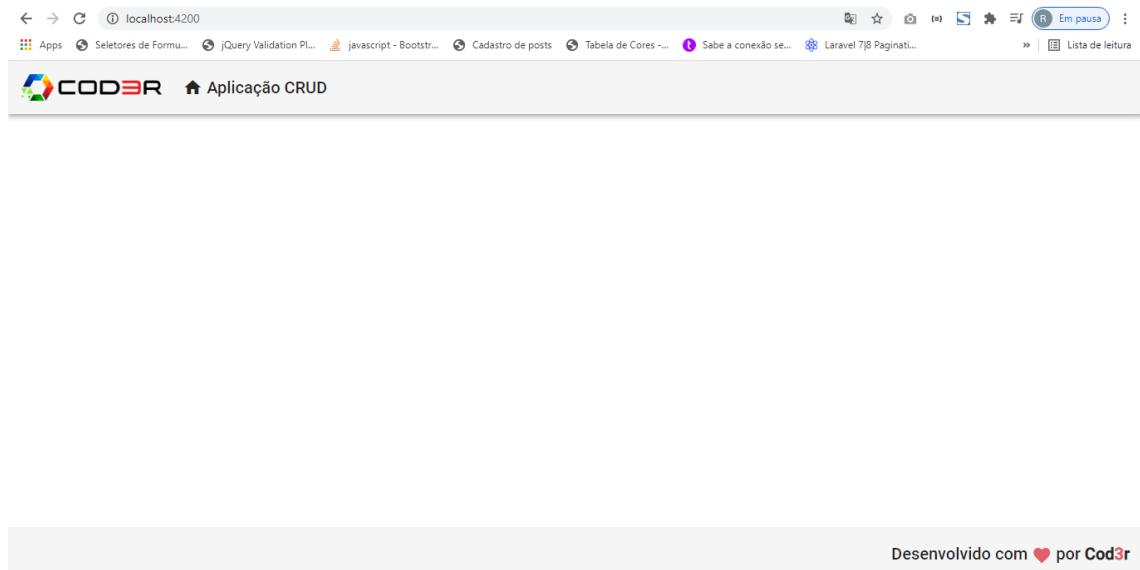
.v-middle {
  vertical-align: middle;
}
```

src\app\components\template\footer\footer.component.css

```
.footer {
  position: fixed;
  bottom: 0;
  display: flex;
  justify-content: flex-end;
}
.footer > span {
  font-size: 1.1rem;
  font-weight: 300;
}
```

- No browser:

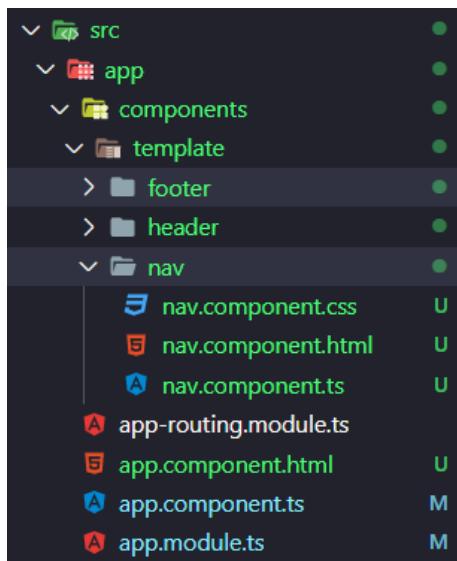
localhost:4200



Aula 10 - Componente Navegação

```
ng generate component components/template/nav
```

```
c:\coder\angular\crud\frontend>ng generate component components/template/nav
CREATE  src/app/components/template/nav/nav.component.html (18 bytes)
CREATE  src/app/components/template/nav/nav.component.ts (263 bytes)
CREATE  src/app/components/template/nav/nav.component.css (0 bytes)
UPDATE  src/app/app.module.ts (860 bytes)
```



src\app\app.module.ts

```
import { NgModule } from '@angular/core';
import { MatListModule } from '@angular/material/list';
import { MatSidenavModule } from '@angular/material/sidenav';
import { MatToolbarModule } from '@angular/material/toolbar';
import { BrowserModule } from '@angular/platform-browser';
import { BrowserAnimationsModule } from '@angular/platform-browser/animations';
import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { FooterComponent } from './components/template/footer/footer.component';
import { HeaderComponent } from './components/template/header/header.component';
import { NavComponent } from './components/template/nav/nav.component';

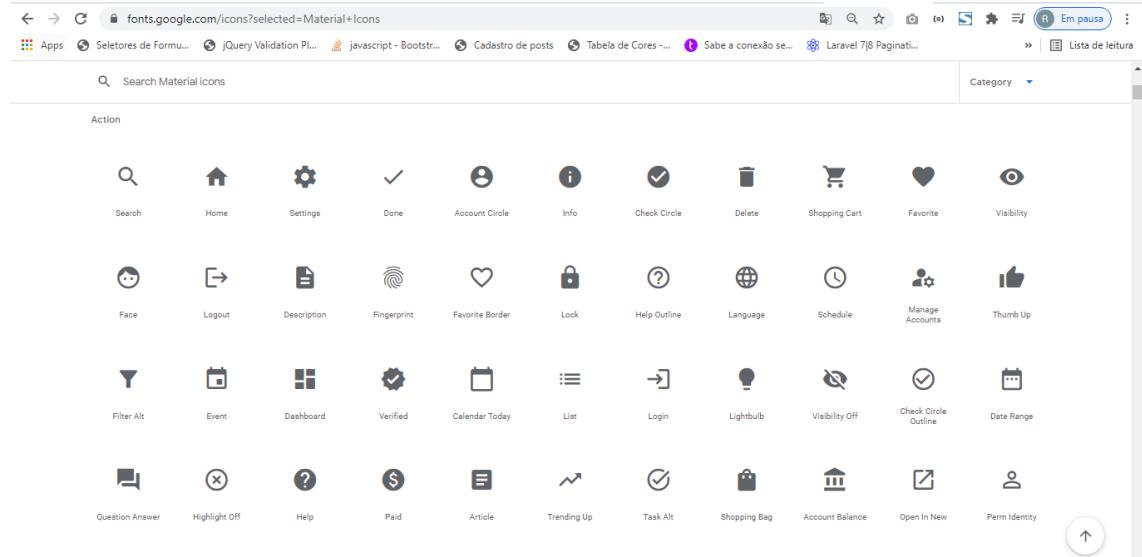
@NgModule({
  declarations: [AppComponent, HeaderComponent, FooterComponent, NavComponent],
  imports: [
    BrowserModule,
    AppRoutingModule,
    BrowserAnimationsModule,
    MatToolbarModule,
    MatSidenavModule,
    MatListModule,
  ],
  providers: [],
  bootstrap: [AppComponent],
})
export class AppModule {}
```

src\app\app.component.html

```
<app-header></app-header>
<app-nav></app-nav>
<app-footer></app-footer>
```

Ícones do Material

<https://fonts.google.com/icons?selected=Material+Icons>



src\app\components\template\nav\nav.component.html

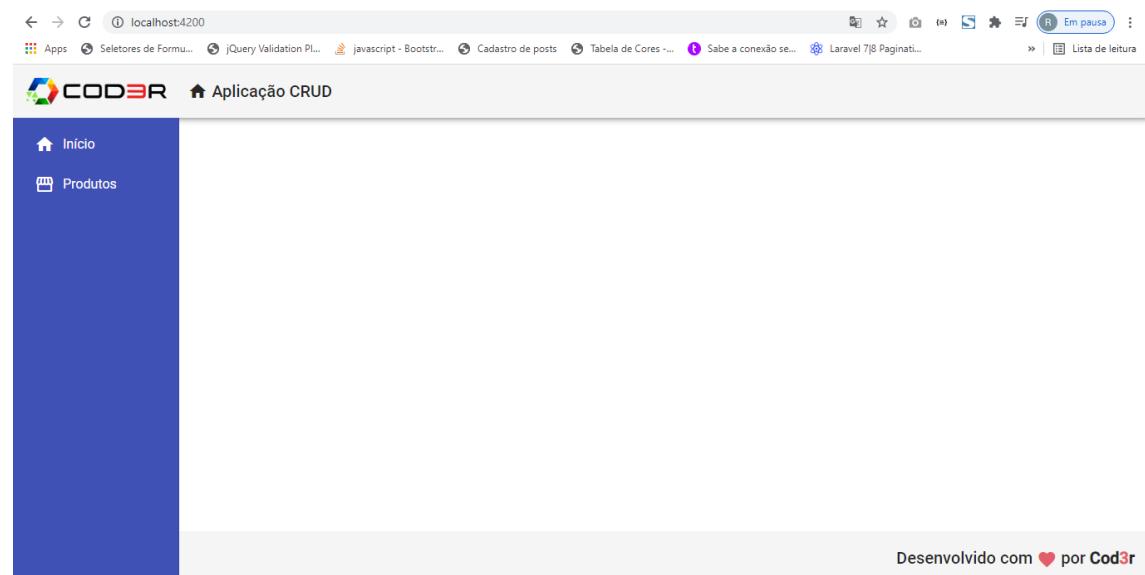
```
<mat-sidenav-container class="container">
  <mat-sidenav
    class="sidenav"
    mode="side"
    opened
    fixedInViewport="true"
    fixedTopGap="64"
  >
    <mat-nav-list class="nav-list">
      <a mat-list-item>
        <i class="material-icons"> home </i>
        Início
      </a>
      <a mat-list-item>
        <i class="material-icons"> storefront </i>
        Produtos
      </a>
    </mat-nav-list>
  </mat-sidenav>
</mat-sidenav-container>
```

src\app\components\template\nav\nav.component.css

```
.sidenav {  
  background-color: #3f51b5;  
  width: 200px;  
}  
.sidenav a {  
  color: #fff;  
}  
.sidenav i {  
  padding: 10px;  
}
```

- No browser:

localhost:4200



Aula 11 - Elementos do Angular - Parte 1

Assuntos importantes:

- Componentes
- Diretivas
- Rotas
- Pipes

Diretivas

1) Diretiva de atributo

Altera a aparência e o comportamento de um elemento, componente ou outra diretiva.



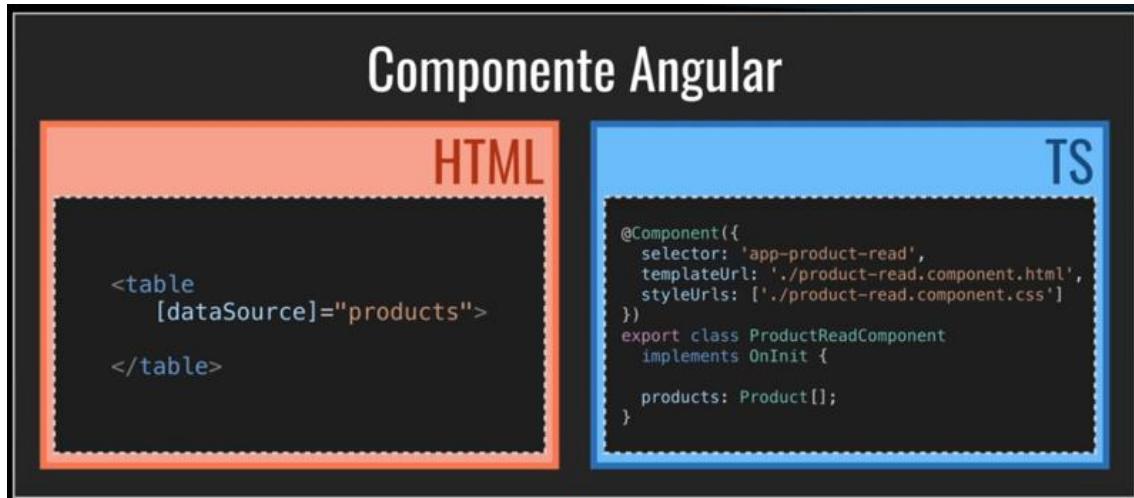
2) Diretiva estrutural

Altera o layout adicionando e removendo elementos da DOM.

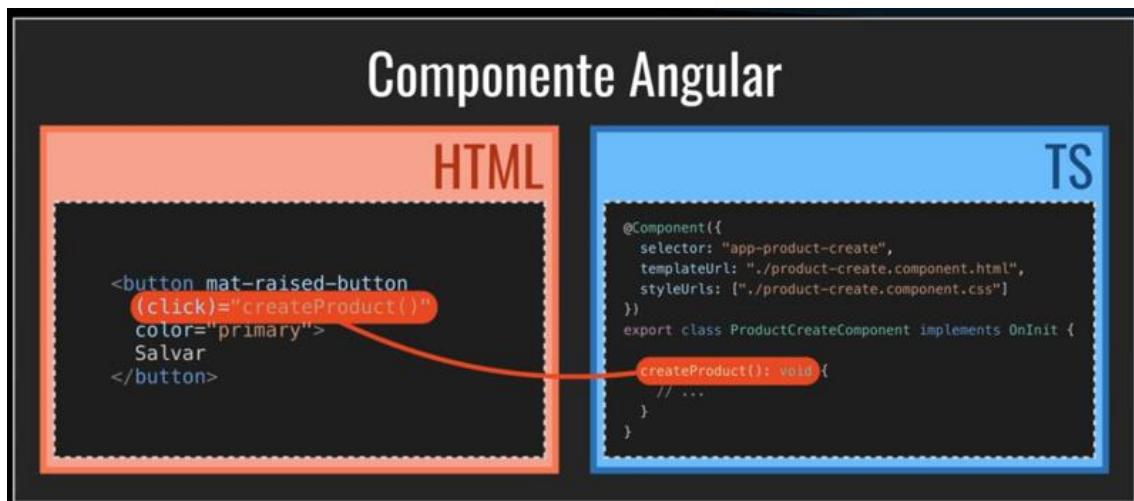


Binding

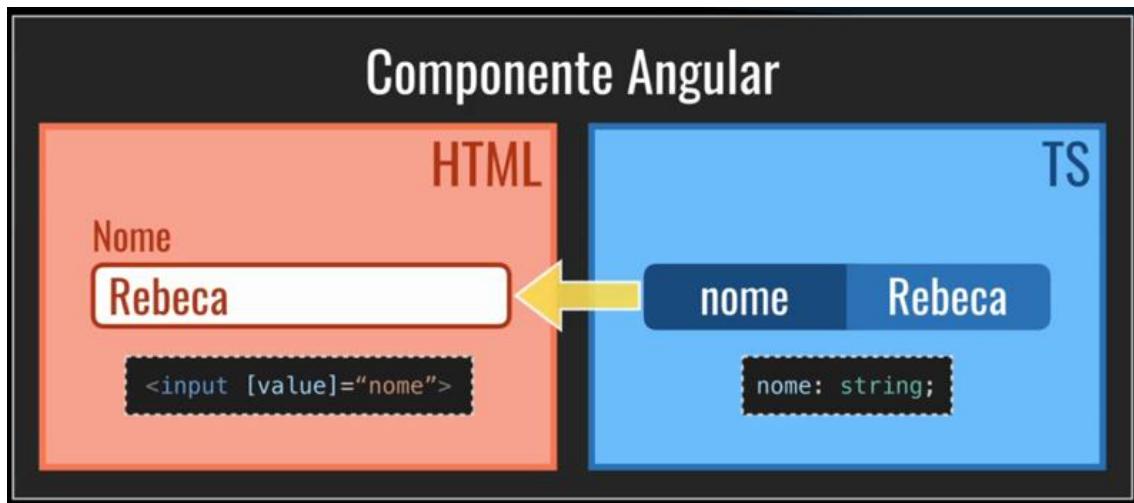
Binding de propriedade (Property Binding)



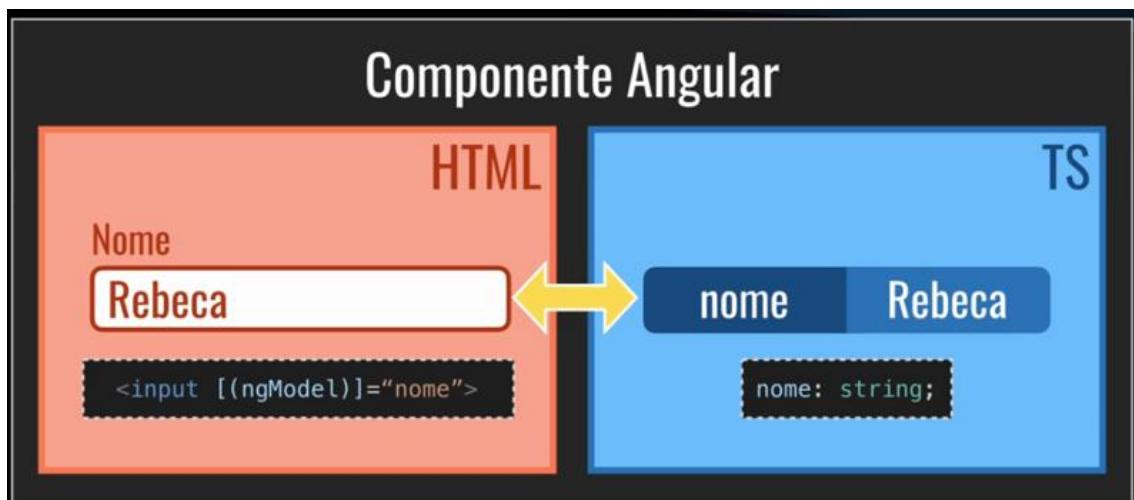
Binding de evento (Event Binding)



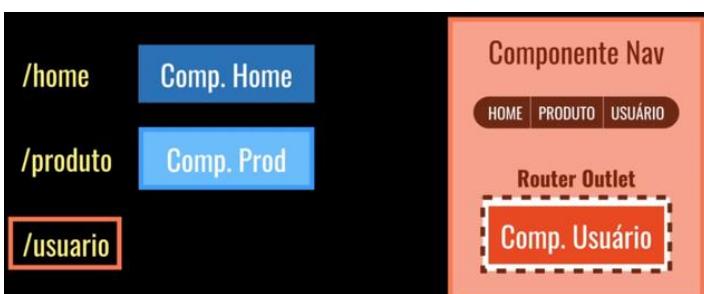
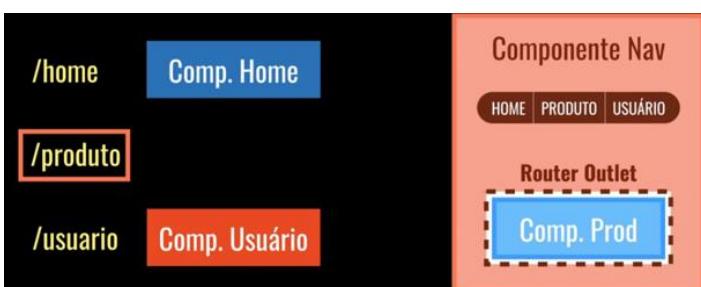
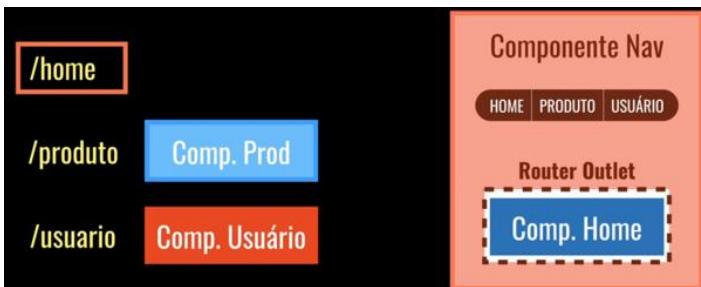
One Way Data Binding



Two Way Data Binding



Rotas (Angular Router)



A diagram illustrating the code structure for the routerLink and routes definitions. On the left, there is an HTML snippet with a link to '/products'. On the right, there is a code block defining routes:

```
<a routerLink="/products">
  Produtos
</a>

const routes: Routes = [
  {
    path: "products",
    component: ProductCrudComponent
  },
  {
    path: "products/create",
    component: ProductCreateComponent
  }];

```

Yellow arrows point from the 'routerLink' attribute in the HTML to the 'path' field in the first route definition, and from the 'products/create' path in the routes array to the 'path' field in the second route definition.

Pipes (Angular Pipes)

```
<p>
  O vencimento é
    {{ produto.vencimento | date }}
</p>
```



```
<td mat-cell *matCellDef="let product">
  {{ product.price | currency: 'BRL' }}
</td>
```



COM PARAMETROS

Encadeamento de pipes:

```
<p>
  O vencimento é
    {{ produto.vencimento | date: 'fullDate' | uppercase }}
</p>
```



CHAINING

Aula 12 - Elementos do Angular - Parte 2

Observable

Programação reativa

- O Angular usa o Reactive X.

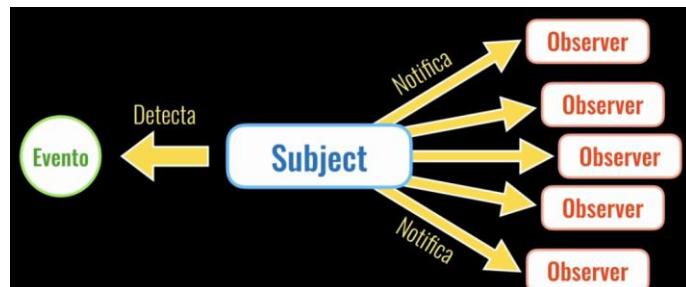
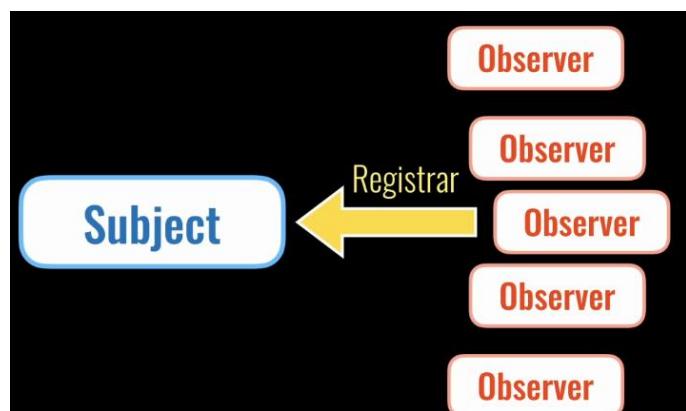
Exemplo:

```
import { Observable } from "rxjs";
```

Padrão Observer

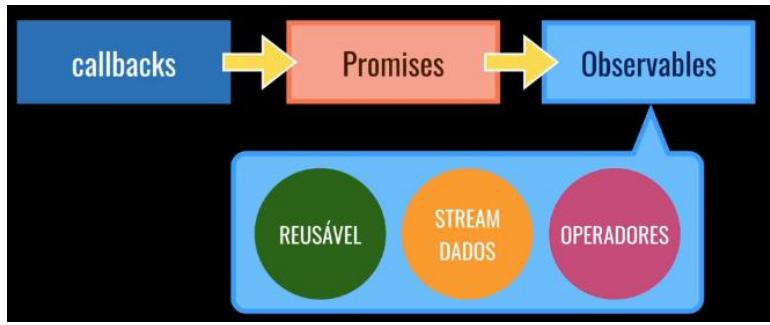
- O padrão Observer é o padrão de projeto mais utilizado da Web.

É um padrão orientado a evento;



Entendendo Observables

Código assíncrono: não há uma chamada imediata do código.



- callbacks: Com uma callback dentro de outra e assim por diante, fica um código de difícil leitura;
- Promises: Permite encadear várias chamadas. Só podem ser executadas uma única vez. Não é possível reutilizar;
- Observables: É reusável. Com um único observable é possível receber uma string de dados em sequência. Operadores como map.

Exemplo:

```

criarNoBackend(produto: Produto): Observable<Produto> {
  return this.http.post<Produto>(this.url, produto);
}

criarProduto(): void {
  this.criarNoBackend(this.produto).subscribe(() => {
    this.exibirMensagem("Salvo com sucesso!");
  });
}

```

Aula 13 - Elementos do Angular - Parte 3

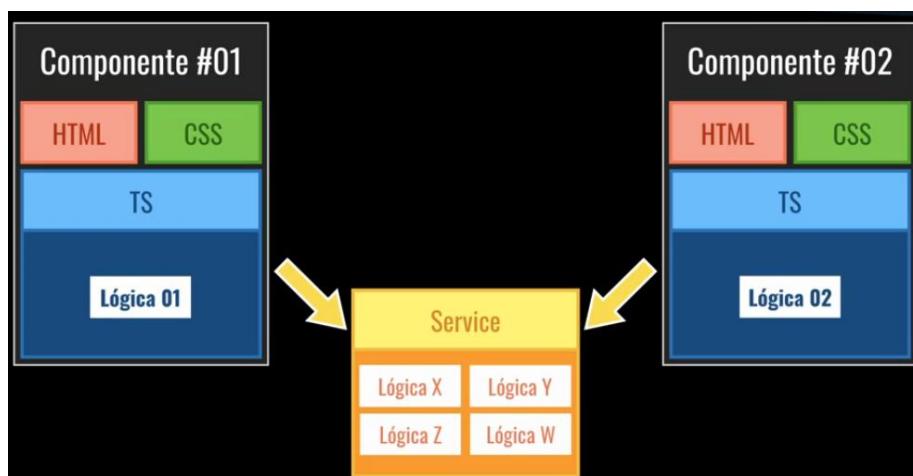
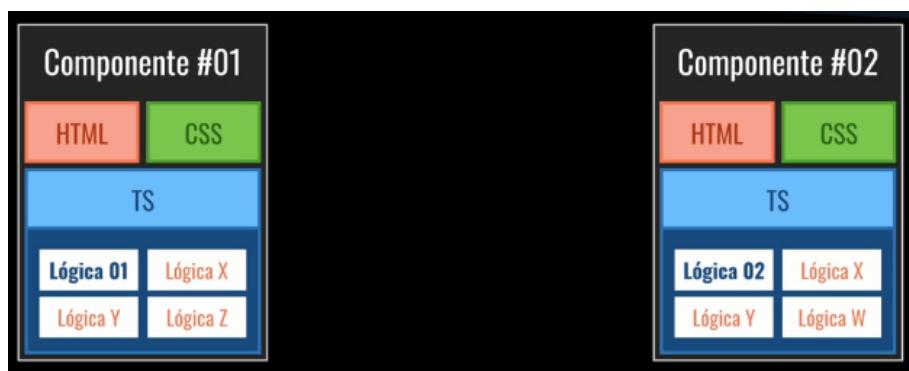
Services

Services são classes que tem como principal objetivo organizar e compartilhar métodos e dados entre componentes.

É possível usar services dentro de diretivas.

Motivação para o service

- Os componentes são responsáveis pela parte visual.



Criando um service

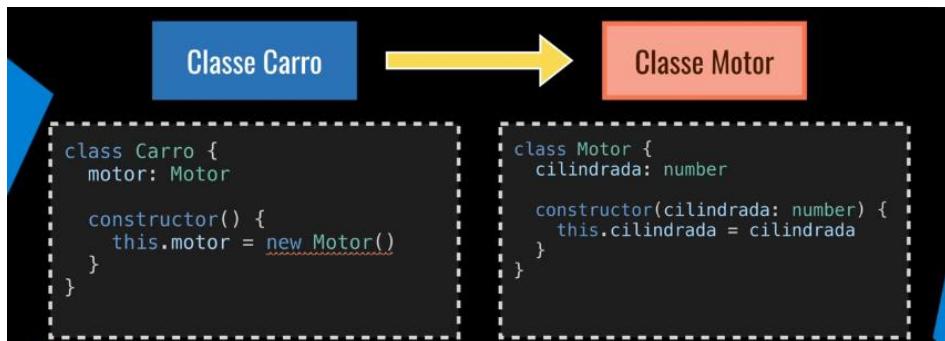
ng g s services/product

```
@Injectable({
  providedIn: "root",
})
export class ProductService {
  // ...
}
```

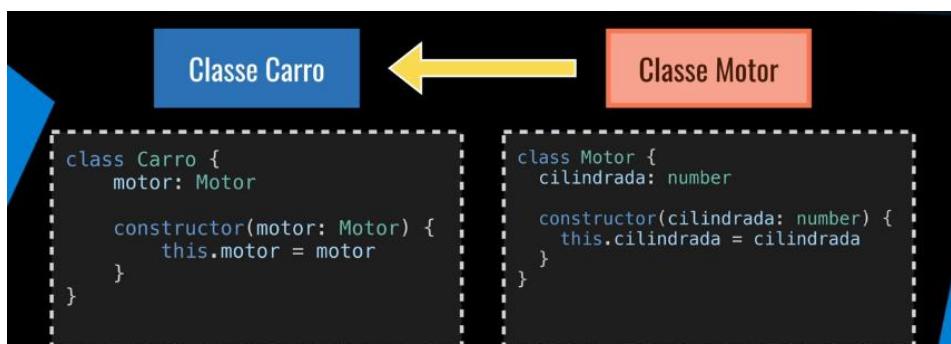
Injeção de dependência

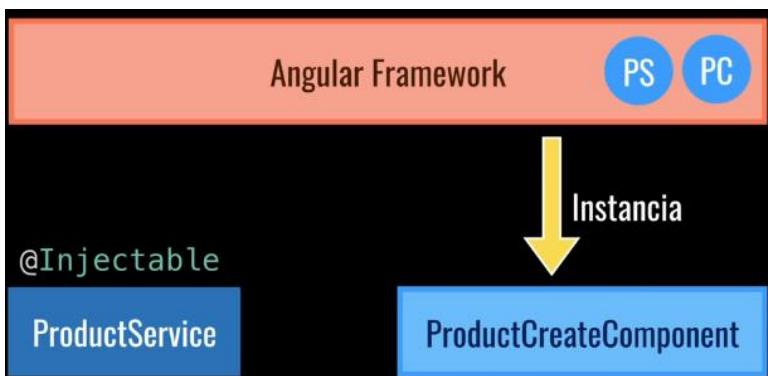
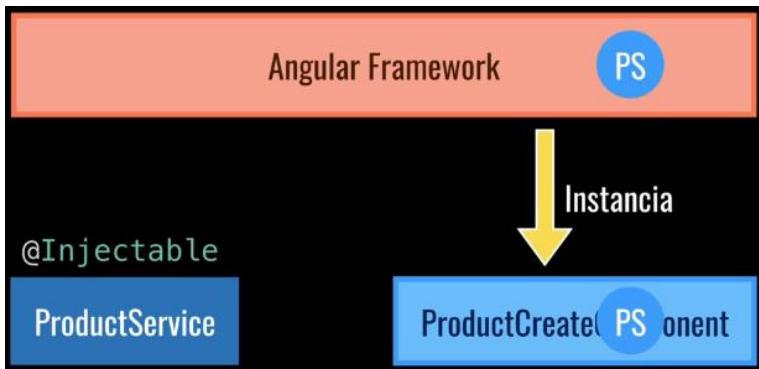
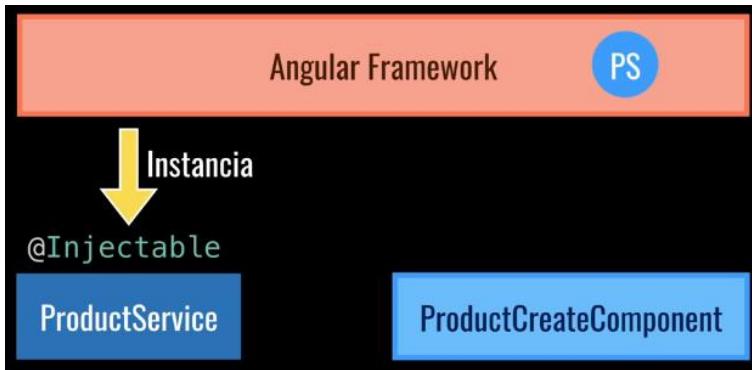
É um padrão no qual a classe recebe as dependências de uma fonte externa ao invés de criar por conta própria.

Exemplo sem injeção de dependência:



Exemplo com injeção de dependência:





- Services são singletons dentro do escopo de um injector.



- Injetor de módulo: uma única instância em toda a aplicação;
- Injetor de elemento;

The diagram illustrates the Angular dependency injection system. At the top, there are two rounded rectangular boxes: a red one labeled "AppModule" and a brown one labeled "Injector". Below them, a dashed rectangular box encloses a portion of the code. A yellow arrow points from the "Injector" box down towards the "providedIn: 'root'" annotation in the code.

```
@Injectable({
  providedIn: "root",
})
export class ProductService {
  // ...
}
```

- No lugar do root é possível ter qualquer outro módulo (sem aspas) da aplicação.

Aula 14 - Componente Início (Home)

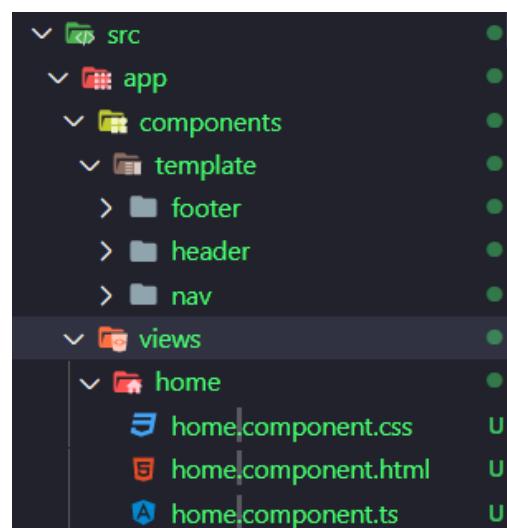
src\app\components\template\header\header.component.html

```
<mat-toolbar class="header mat-elevation-z4">
  <span>
    <a>
      
    </a>
  </span>
  <span class="title-group">
    <a>
      <!-- <i class="material-icons"> dashboard </i> -->
      <i class="material-icons"> home </i>
      Início
    </a>
  </span>
</mat-toolbar>
```

Criando o componente home

ng generate component views/home

```
C:\coder\angular\crud\frontend>ng generate component views/home
CREATE src/app/views/home/home.component.html (19 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 (1095 bytes)
```



src\app\components\template\nav\nav.component.html

```
<mat-sidenav-container class="container">
  <mat-sidenav
    class="sidenav"
    mode="side"
    opened
    fixedInViewport="true"
    fixedTopGap="64"
  >
    <mat-nav-list class="nav-list">
      <a mat-list-item>
        <i class="material-icons"> home </i>
        Início
      </a>
      <a mat-list-item>
        <i class="material-icons"> storefront </i>
        Produtos
      </a>
    </mat-nav-list>
  </mat-sidenav>
  <mat-sidenav-content class="content">
    <app-home></app-home>
  </mat-sidenav-content>
</mat-sidenav-container>
```

src\app\app.module.ts

```
import { NgModule } from '@angular/core';
import { MatCardModule } from '@angular/material/card';
import { MatListModule } from '@angular/material/list';
import { MatSidenavModule } from '@angular/material/sidenav';
import { MatToolbarModule } from '@angular/material/toolbar';
import { BrowserModule } from '@angular/platform-browser';
import { BrowserAnimationsModule } from '@angular/platform-browser/animations';
import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { FooterComponent } from './components/template/footer/footer.component';
import { HeaderComponent } from './components/template/header/header.component';
import { NavComponent } from './components/template/nav/nav.component';
import { HomeComponent } from './views/home/home.component';

@NgModule({
  declarations: [
    AppComponent,
    HeaderComponent,
    FooterComponent,
    NavComponent,
    HomeComponent,
  ],
  imports: [
    BrowserModule,
    AppRoutingModule,
    BrowserAnimationsModule,
    MatToolbarModule,
    MatSidenavModule,
    MatListModule,
    MatCardModule,
  ],
  providers: [],
  bootstrap: [AppComponent],
})
export class AppModule {}
```

src\app\views\home\home.component.html

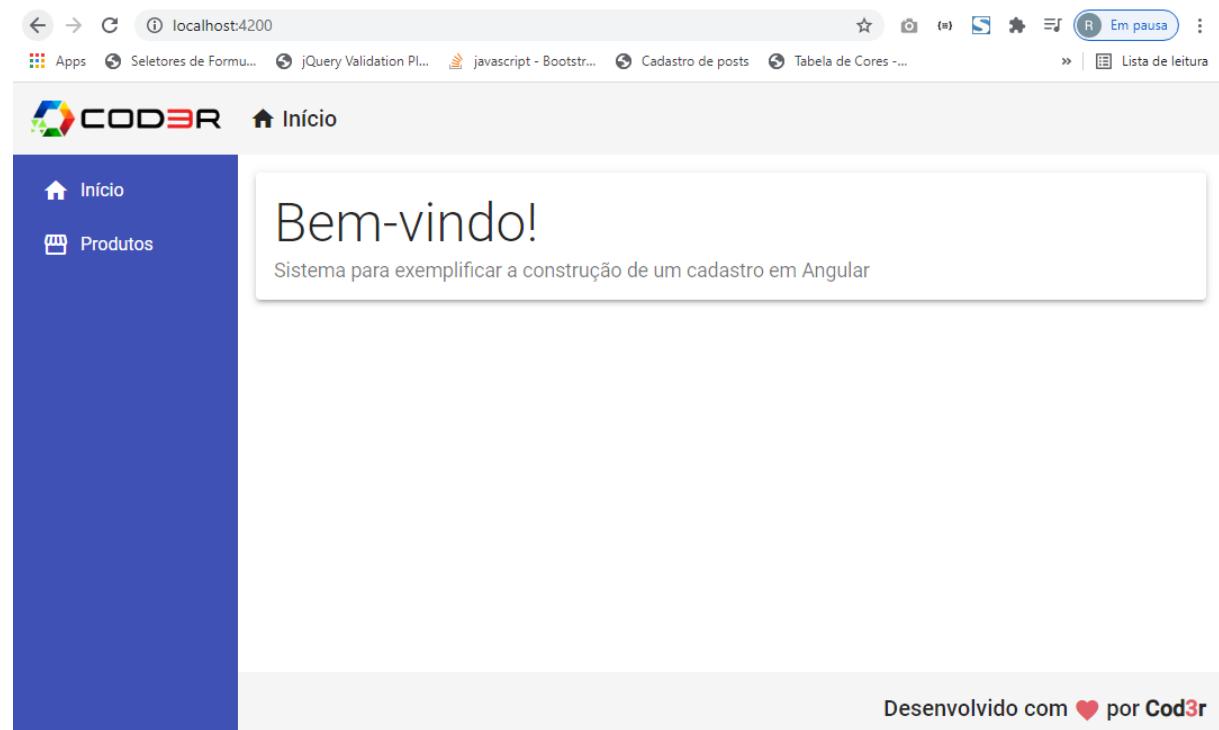
```
<mat-card class="home mat-elevation-z3">
  <mat-card-title class="title"> Bem-vindo! </mat-card-title>
  <mat-card-subtitle class="subtitle">
    Sistema para exemplificar a construção de um cadastro em Angular
  </mat-card-subtitle>
</mat-card>
```

src\app\views\home\home.component.css

```
.home .title {  
    font-size: 3rem;  
    font-weight: 300;  
    line-height: 1.2;  
}  
.home .subtitle {  
    font-size: 1.1rem;  
}
```

- No browser:

localhost:4200

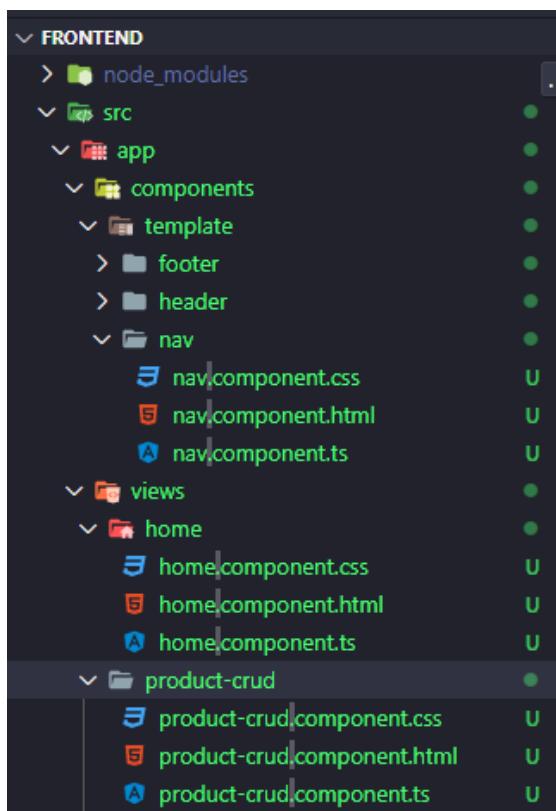


Aula 15 - Navegando entre componentes

Criando o componente de product

```
ng generate component views/product-crud
```

```
c:\coder\angular\crud\frontend>ng generate component views/product-crud
CREATE src/app/views/product-crud/product-crud.component.html (27 bytes)
CREATE src/app/views/product-crud/product-crud.component.ts (298 bytes)
CREATE src/app/views/product-crud/product-crud.component.css (0 bytes)
UPDATE src/app/app.module.ts (1305 bytes)
```



src\app\components\template\nav\nav.component.html

```
<mat-sidenav-container class="container">
<mat-sidenav
  class="sidenav"
  mode="side"
  opened
  fixedInViewport="true"
  fixedTopGap="64"
>
  <mat-nav-list class="nav-list">
    <a mat-list-item routerLink="/">
      <i class="material-icons"> home </i>
      Início
    </a>
    <a mat-list-item routerLink="/products">
      <i class="material-icons"> storefront </i>
      Produtos
    </a>
  </mat-nav-list>
</mat-sidenav>
<mat-sidenav-content class="content">
  <router-outlet></router-outlet>
</mat-sidenav-content>
</mat-sidenav-container>
```

Trabalhando com rotas

src\app\app-routing.module.ts

```
import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';
import { HomeComponent } from './views/home/home.component';
import { ProductCrudComponent } from './views/product-crud/product-crud.component';

const routes: Routes = [
  {
    path: '',
    component: HomeComponent,
  },
  {
    path: 'products',
    component: ProductCrudComponent,
  },
];

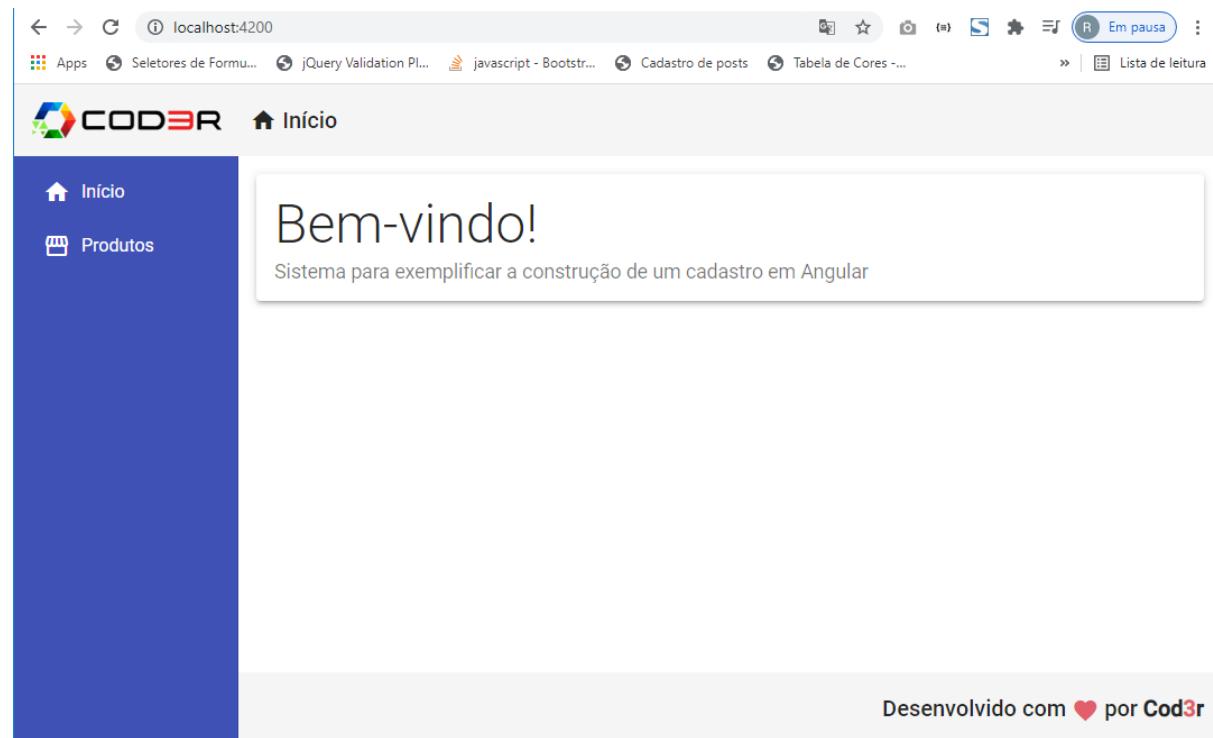
@NgModule({
  imports: [RouterModule.forRoot(routes)],
  exports: [RouterModule],
})
export class AppRoutingModule {}
```

src\app\components\template\header\header.component.html

```
<mat-toolbar class="header mat-elevation-z4">
  <span>
    <a routerLink="/">
      
    </a>
  </span>
  <span class="title-group">
    <a>
      <i class="material-icons"> home </i>
      Início
    </a>
  </span>
</mat-toolbar>
```

- No browser:

localhost:4200



- Clicando no link "**Produtos**" do menu vertical:

A screenshot of a web browser window. The address bar shows the URL `localhost:4200/products`. The page content includes a sidebar with icons for 'Início' and 'Produtos'. The main area displays the text 'product-crud works!'. In the bottom right corner, there is a footer with the text 'Desenvolvido com ❤️ por Cod3r'.

- Clicando no link "Início" do menu vertical, ou no logo:

A screenshot of a web browser window. The address bar shows the URL `localhost:4200`. The page content includes a sidebar with icons for 'Início' and 'Produtos'. The main area features a large box containing the text 'Bem-vindo!' and 'Sistema para exemplificar a construção de um cadastro em Angular'. In the bottom right corner, there is a footer with the text 'Desenvolvido com ❤️ por Cod3r'.

Aula 16 - Diretivas na prática - Parte 1

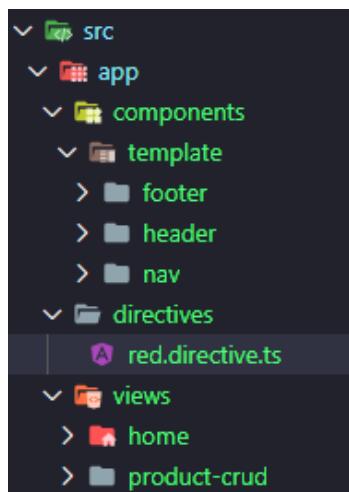
- Há dois tipos de diretivas:

1. **Diretivas de atributo**: são capazes de trabalhar tanto na parte do estilo como no comportamento
2. **Diretivas estruturais**: são capazes de alterar a estrutura da página, ou seja são capazes de adicionar ou remover elemento na DOM. As estruturas estruturais utilizam asterisco na frente.

Criando uma diretiva de atributo

ng g d directives/red

```
C:\coder\angular\crud\frontend>ng g d directives/red
CREATE src/app/directives/red.directive.ts (135 bytes)
UPDATE src/app/app.module.ts (1382 bytes)
```



src\app\app.module.ts

```
import { NgModule } from '@angular/core';
import { MatCardModule } from '@angular/material/card';
import { MatListModule } from '@angular/material/list';
import { MatSidenavModule } from '@angular/material/sidenav';
import { MatToolbarModule } from '@angular/material/toolbar';
import { BrowserModule } from '@angular/platform-browser';
import { BrowserAnimationsModule } from '@angular/platform-browser/animations';
import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { FooterComponent } from './components/template/footer/footer.component';
import { HeaderComponent } from './components/template/header/header.component';
import { NavComponent } from './components/template/nav/nav.component';
import { HomeComponent } from './views/home/home.component';
import { ProductCrudComponent } from './views/product-crud/product-crud.component';
import { RedDirective } from './directives/red.directive';

@NgModule({
  declarations: [
    AppComponent,
    HeaderComponent,
    FooterComponent,
    NavComponent,
    HomeComponent,
    ProductCrudComponent,
    RedDirective,
  ],
  imports: [
    BrowserModule,
    AppRoutingModule,
    BrowserAnimationsModule,
    MatToolbarModule,
    MatSidenavModule,
    MatListModule,
    MatCardModule,
  ],
  providers: [],
  bootstrap: [AppComponent],
})
export class AppModule {}
```

src\app\directives\red.directive.ts

```
import { Directive, ElementRef } from '@angular/core';

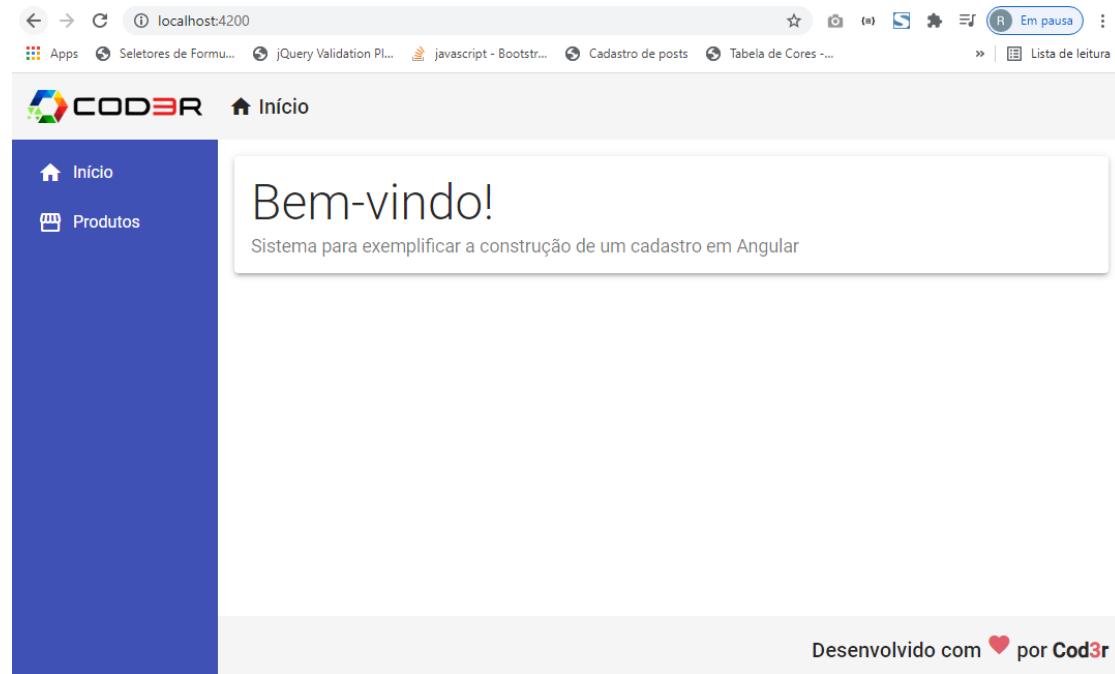
@Directive({
  selector: '[appRed]',
})
export class RedDirective {
  constructor(private el: ElementRef) {
    el.nativeElement.style.color = '#e35e6b';
  }
}
```

src\app\components\template\footer\footer.component.html

```
<mat-toolbar class="footer">
<span>
  Desenvolvido com
  <i class="material-icons middle" appRed> favorite </i>
  por <strong>Cod<span class="red">3</span>r</strong>
</span>
</mat-toolbar>
```

- No browser:

localhost:4200

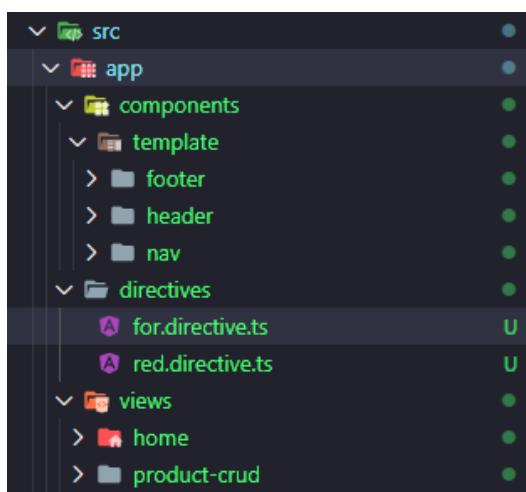


Aula 17 - Diretivas na prática - Parte 2

Criando uma diretiva estrutural

ng g d directives/for

```
C:\coder\angular\crud\frontend>ng g d directives/for
CREATE src/app/directives/for.directive.ts (135 bytes)
UPDATE src/app/app.module.ts (1459 bytes)
```



- No Visual Studio Code instale o plugin "Auto Import":

The screenshot shows the 'Auto Import' extension page on the Visual Studio Code Marketplace. The extension is developed by 'steoates' and has a version of 1.55.0. It has a rating of 4 stars and is compatible with TypeScript and TSX. The description states: 'Automatically finds, parses and provides code actions and code completion for all available imports. Works with Typescript and TSX.' A note below says: 'Big thanks to everyone who has used this over the years. If you have found this helpful at all, feel free to buy me a coffee! (<https://www.buymeacoffee.com/steoates>)'. The configuration section includes settings for 'filesToScan' (glob for files to scan, defaulting to '**/*.{ts, tsx}') and 'showNotifications' (controls if notifications are shown, defaulting to false). There is also a 'Details' tab and a 'Feature Contributions' tab.

src\app\directives\for.directive.ts

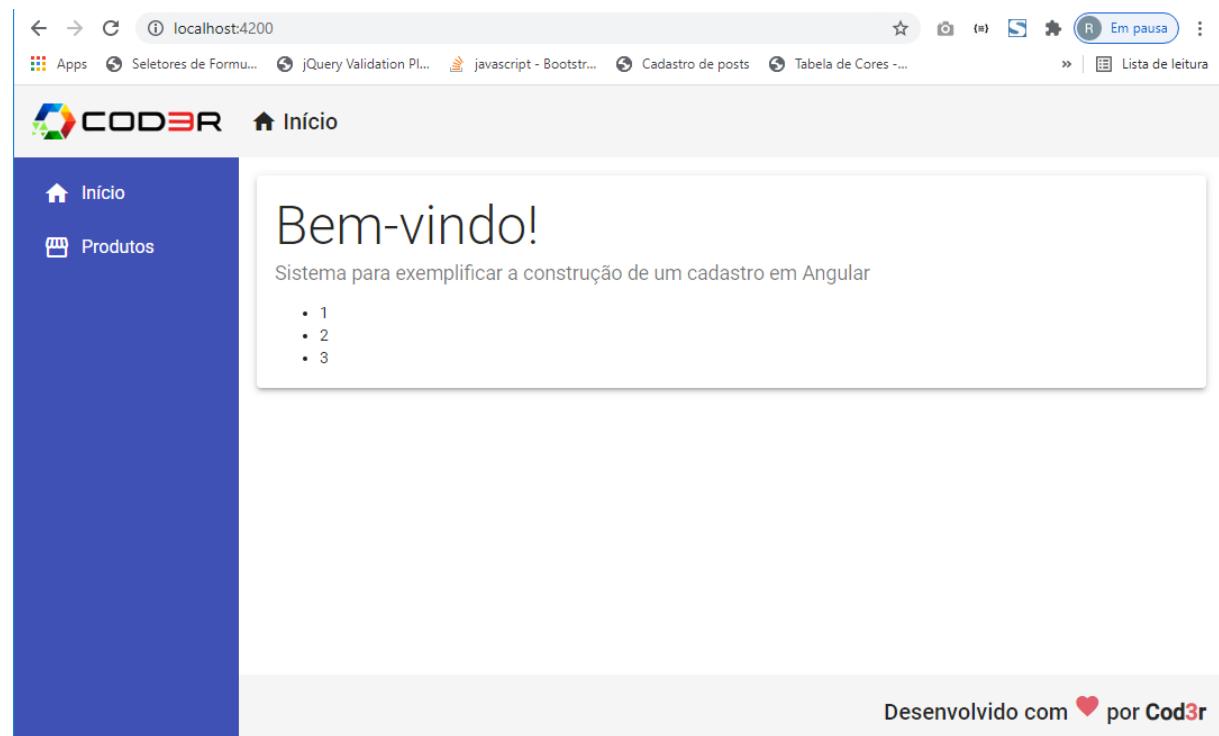
```
import {  
  Directive,  
  Input,  
  OnInit,  
  TemplateRef,  
  ViewContainerRef,  
} from '@angular/core';  
  
@Directive({  
  selector: '[myFor]',  
})  
export class ForDirective implements OnInit {  
  @Input('myForEm') numbers: number[];  
  
  constructor(  
    private container: ViewContainerRef,  
    private template: TemplateRef<any>  
  ) {}  
  
  ngOnInit(): void {  
    for (let number of this.numbers) {  
      this.container.createEmbeddedView(this.template, { $implicit: number });  
    }  
  }  
}
```

src\app\views\home\home.component.html

```
<mat-card class="home mat-elevation-z3">
  <mat-card-title class="title"> Bem-vindo! </mat-card-title>
  <mat-card-subtitle class="subtitle">
    Sistema para exemplificar a construção de um cadastro em Angular
  </mat-card-subtitle>
  <ul>
    <li *myFor="let n; em: [1, 2, 3]">{{ n }}</li>
  </ul>
</mat-card>
```

- No browser:

localhost: 4200

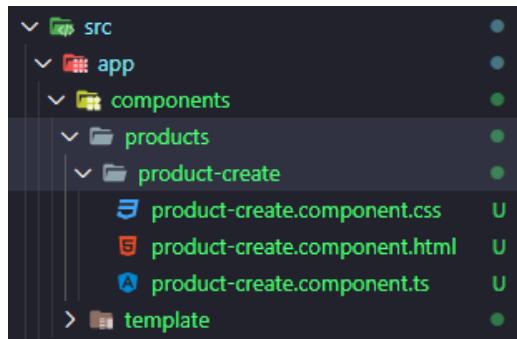


Aula 18 - Componente Criar Produto

Criando o componente

```
ng g c components/products/product-create
```

```
C:\coder\angular\crud\frontend>ng g c components/products/product-create
CREATE src/app/components/products/product-create/product-create.component.html (29 bytes)
CREATE src/app/components/products/product-create/product-create.component.ts (306 bytes)
CREATE src/app/components/products/product-create/product-create.component.css (0 bytes)
UPDATE src/app/app.module.ts (1591 bytes)
```



src\app\app-routing.module.ts

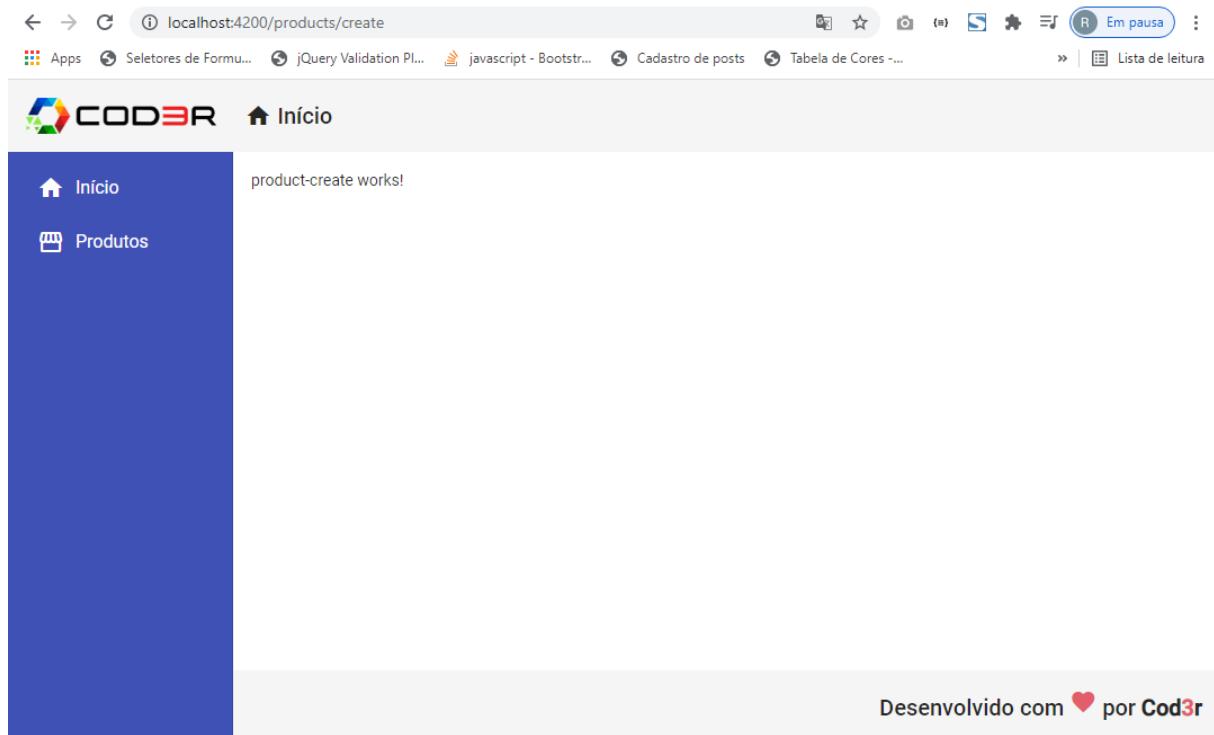
```
import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';
import { ProductCreateComponent } from './components/products/product-create/product-create.component';
import { HomeComponent } from './views/home/home.component';
import { ProductCrudComponent } from './views/product-crud/product-crud.component';

const routes: Routes = [
{
  path: '',
  component: HomeComponent,
},
{
  path: 'products',
  component: ProductCrudComponent,
},
{
  path: 'products/create',
  component: ProductCreateComponent,
},
];

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

- No browser:

localhost:4200/products/create



Aula 19 - Bindings

Binding:

- de evento
- de atributo

Binding de evento

src\app\components\products\product-create\product-create.component.ts

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

@Component({
  selector: 'app-product-create',
  templateUrl: './product-create.component.html',
  styleUrls: ['./product-create.component.css'],
})
export class ProductCreateComponent implements OnInit {

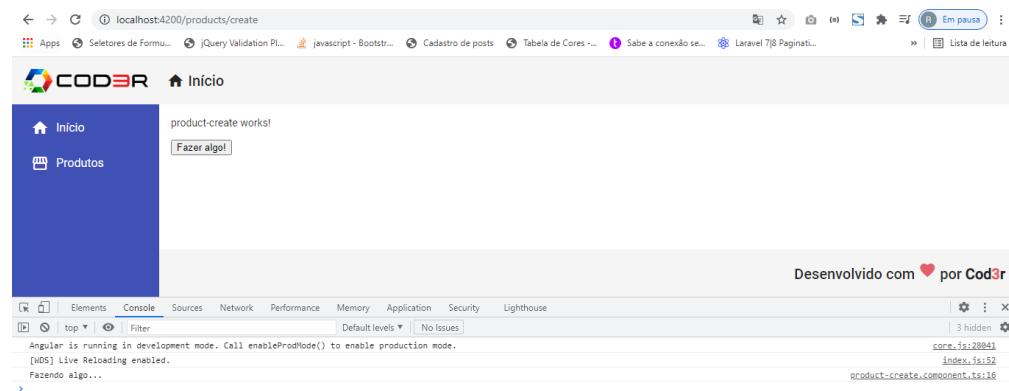
  constructor() {}
  ngOnInit(): void {}

  fazerAlgo(): void {
    console.log('Fazendo algo...');
  }
}
```

src\app\components\products\product-create\product-create.component.html

```
<p>product-create works!</p>
<button (click)="fazerAlgo()">Fazer algo!</button>
```

Clicando no botão:



Binding de atributo

src\app\components\products\product-create\product-create.component.ts

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

@Component({
  selector: 'app-product-create',
  templateUrl: './product-create.component.html',
  styleUrls: ['./product-create.component.css'],
})
export class ProductCreateComponent implements OnInit {
  propLegal = 'qualquer';

  constructor() {}

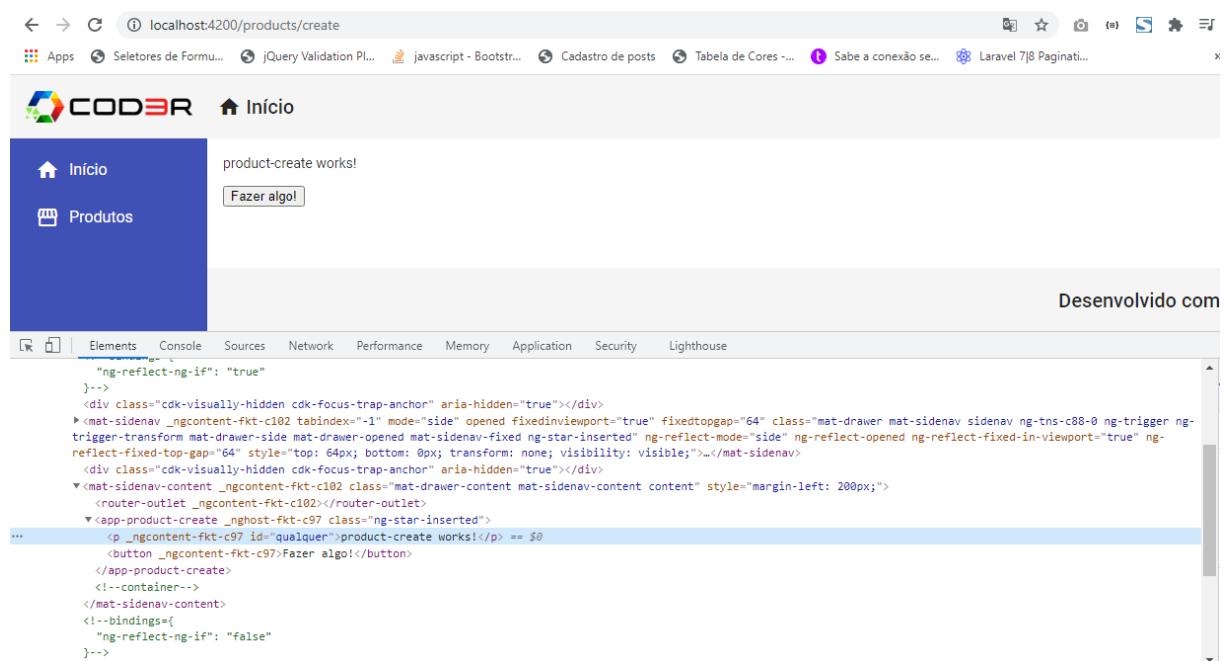
  ngOnInit(): void {}

  fazerAlgo(): void {
    console.log('Fazendo algo...');
  }
}
```

src\app\components\products\product-create\product-create.component.html

```
<p [id]="propLegal">product-create works!</p>
<button (click)="fazerAlgo()">Fazer algo!</button>
```

- No browser:



Aula 20 - Navegar para a tela de criar produtos

src\app\app.module.ts

```
import { NgModule } from '@angular/core';
import { MatButtonModule } from '@angular/material/button';
import { MatCardModule } from '@angular/material/card';
import { MatListModule } from '@angular/material/list';
import { MatSidenavModule } from '@angular/material/sidenav';
import { MatToolbarModule } from '@angular/material/toolbar';
import { BrowserModule } from '@angular/platform-browser';
import { BrowserAnimationsModule } from '@angular/platform-browser/animations';
import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { ProductCreateComponent } from './components/products/product-create/product-create.component';
import { FooterComponent } from './components/template/footer/footer.component';
import { HeaderComponent } from './components/template/header/header.component';
import { NavComponent } from './components/template/nav/nav.component';
import { ForDirective } from './directives/for.directive';
import { RedDirective } from './directives/red.directive';
import { HomeComponent } from './views/home/home.component';
import { ProductCrudComponent } from './views/product-crud/product-crud.component';

@NgModule({
  declarations: [
    AppComponent,
    HeaderComponent,
    FooterComponent,
    NavComponent,
    HomeComponent,
    ProductCrudComponent,
    RedDirective,
    ForDirective,
    ProductCreateComponent,
  ],
  imports: [
    BrowserModule,
    AppRoutingModule,
    BrowserAnimationsModule,
    MatToolbarModule,
    MatSidenavModule,
    MatListModule,
    MatCardModule,
    MatButtonModule,
  ],
  providers: [],
  bootstrap: [AppComponent],
})
export class AppModule {}
```

src\app\views\product-crud\product-crud.component.html

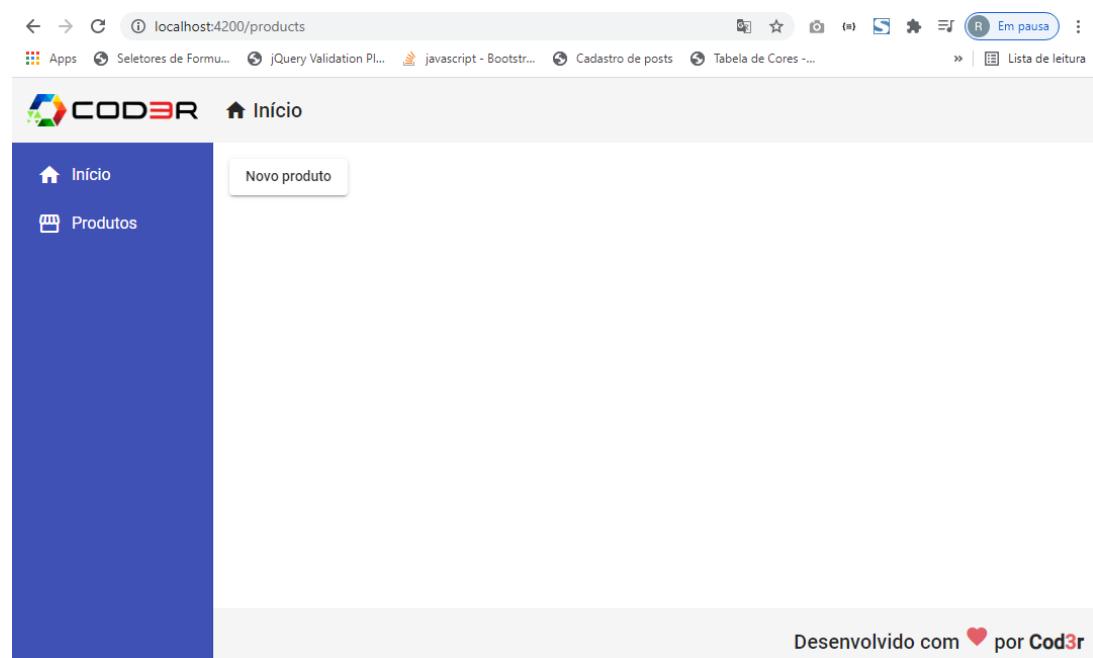
```
<button mat-raised-button (click)="navigateToProductCreate()">  
  Novo produto  
</button>
```

src\app\views\product-crud\product-crud.component.ts

```
import { Component, OnInit } from '@angular/core';  
import { Router } from '@angular/router';  
  
@Component({  
  selector: 'app-product-crud',  
  templateUrl: './product-crud.component.html',  
  styleUrls: ['./product-crud.component.css'],  
})  
export class ProductCrudComponent implements OnInit {  
  constructor(private router: Router) {}  
  
  ngOnInit(): void {}  
  
  navigateToProductCreate(): void {  
    this.router.navigate(['/products/create']);  
  }  
}
```

- No browser:

<http://localhost:4200/products>



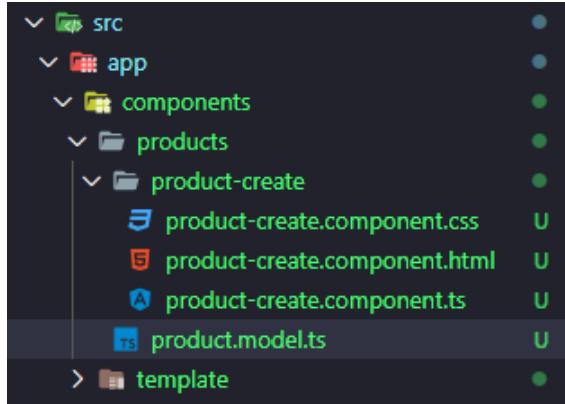
- Clicando no botão "Novo produto":

The screenshot shows a web browser window with the URL `localhost:4200/products/create` in the address bar. The browser interface includes standard navigation buttons (back, forward, search) and a toolbar with various icons. Below the address bar is a navigation bar with links: "Apps", "Seletores de Formu...", "jQuery Validation Pl...", "javascript - Bootstr...", "Cadastro de posts", "Tabela de Cores -...", and "Lista de leitura". A "Em pausa" (Paused) button is also visible in the toolbar.

The main content area has a header with the Cod3R logo and the text "Início". On the left, there is a sidebar with two items: "Início" and "Produtos". The main content area displays the message "product-create works!" and a button labeled "Fazer algo!". At the bottom right of the page, there is a footer with the text "Desenvolvido com ❤️ por Cod3R".

Aula 21 - Produto Modelo

- Dentro da pasta "src\app\components\products" adicione um arquivo chamado "product.model.ts":



src\app\components\products\product.model.ts

```
export interface Product {  
  id?: number;  
  name: string;  
  price: number | null;  
}
```

frontend\tsconfig.json

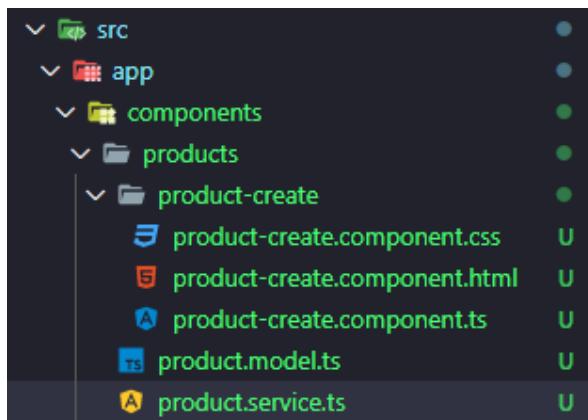
```
/* To learn more about this file see: https://angular.io/config/tsconfig. */
{
  "compileOnSave": false,
  "compilerOptions": {
    "baseUrl": "./",
    "outDir": "./dist/out-tsc",
    "forceConsistentCasingInFileNames": true,
    "strict": true,
    "noImplicitReturns": true,
    "noFallthroughCasesInSwitch": true,
    "sourceMap": true,
    "declaration": false,
    "downlevelIteration": true,
    "experimentalDecorators": true,
    "moduleResolution": "node",
    "importHelpers": true,
    "target": "es2017",
    "module": "es2020",
    "lib": [
      "es2018",
      "dom"
    ]
  },
  "angularCompilerOptions": {
    "enableI18nLegacyMessageIdFormat": false,
    "strictInjectionParameters": true,
    "strictInputAccessModifiers": true,
    "strictTemplates": true,
    "strictPropertyInitialization": false,
    "strictNullChecks": false
  }
}
```

Aula 22 - Produto Service

Criando o service para products

ng g s components/products/product

```
C:\coder\angular\crud\frontend>ng g s components/products/product
CREATE src/app/components/products/product.service.ts (136 bytes)
```



src\app\components\products\product.service.ts

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

```
@Injectable({
  providedIn: 'root',
})
export class ProductService {
  constructor() {}

  showOnConsole(msg: string): void {
    console.log(msg);
  }
}
```

src\app\components\products\product-create\product-create.component.ts

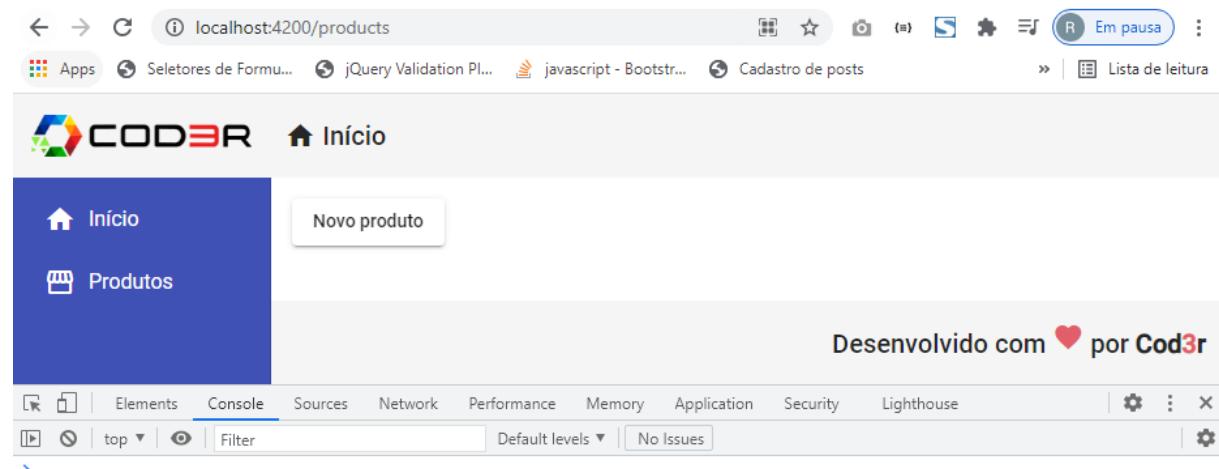
```
import { Component, OnInit } from '@angular/core';
import { ProductService } from './product.service';

@Component({
  selector: 'app-product-create',
  templateUrl: './product-create.component.html',
  styleUrls: ['./product-create.component.css'],
})
export class ProductCreateComponent implements OnInit {
  constructor(private productService: ProductService) {}

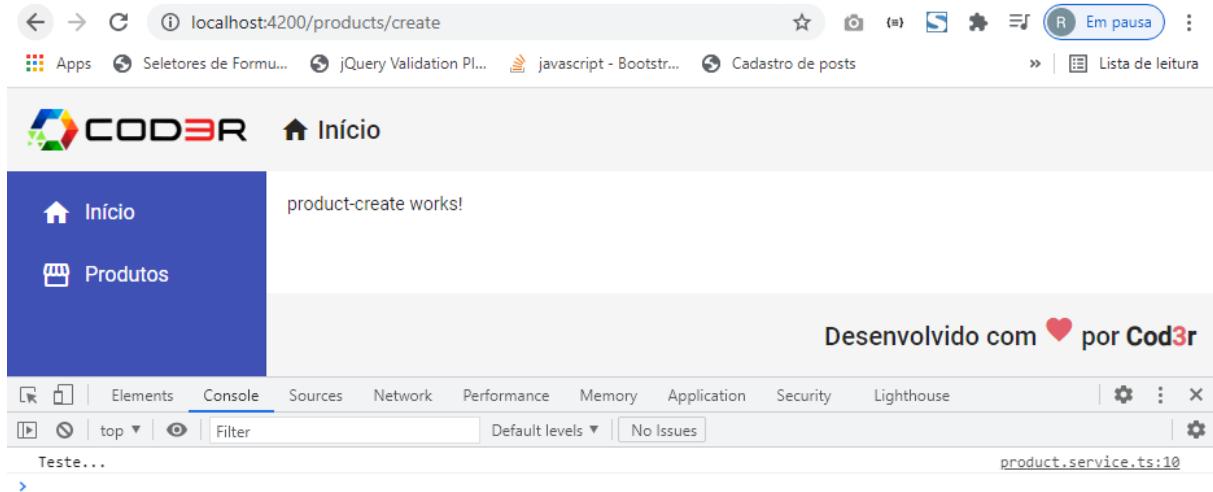
  ngOnInit(): void {
    this.productService.showOnConsole('Teste...');
  }
}
```

- No browser:

<http://localhost:4200/products>



- Clicando no botão "Novo produto":



src\app\components\products\product.service.ts

```
import { Injectable } from '@angular/core';
import { MatSnackBar } from '@angular/material/snack-bar';

@Injectable({
  providedIn: 'root',
})
export class ProductService {
  constructor(private snackBar: MatSnackBar) {}

  showMessage(msg: string): void {
    this.snackBar.open(msg, 'X', {
      duration: 3000,
      horizontalPosition: 'right',
      verticalPosition: 'top',
    });
  }
}
```

src\app\components\products\product-create\product-create.component.ts

```
import { Component, OnInit } from '@angular/core';
import { Router } from '@angular/router';
import { ProductService } from './../../../product.service';

@Component({
  selector: 'app-product-create',
  templateUrl: './product-create.component.html',
  styleUrls: ['./product-create.component.css'],
})
export class ProductCreateComponent implements OnInit {
  constructor(private productService: ProductService, private router: Router) {}

  ngOnInit(): void {}

  createProduct(): void {
    this.productService.showMessage('Produto cadastrado com sucesso!');
  }

  cancel(): void {
    this.router.navigate(['/products']);
  }
}
```

src\app\components\products\product-create\product-create.component.html

```
<p>product-create works!</p>

<button mat-raised-button (click)="createProduct()" color="primary">
  Salvar
</button>

<button mat-raised-button (click)="cancel()>Cancelar</button>
```

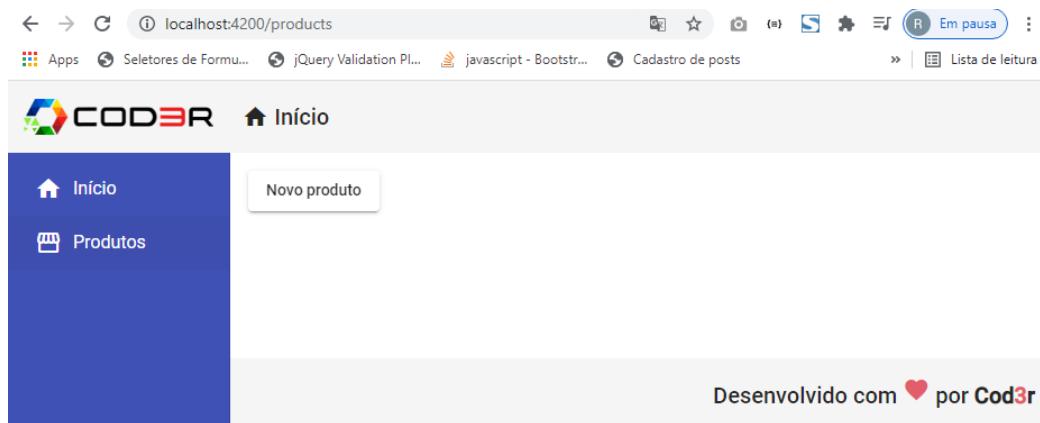
src\app\app.module.ts

```
import { NgModule } from '@angular/core';
import { MatButtonModule } from '@angular/material/button';
import { MatCardModule } from '@angular/material/card';
import { MatListModule } from '@angular/material/list';
import { MatSidenavModule } from '@angular/material/sidenav';
import { MatSnackBarModule } from '@angular/material/snack-bar';
import { MatToolbarModule } from '@angular/material/toolbar';
import { BrowserModule } from '@angular/platform-browser';
import { BrowserAnimationsModule } from '@angular/platform-browser/animations';
import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { ProductCreateComponent } from './components/products/product-create/product-create.component';
import { FooterComponent } from './components/template/footer/footer.component';
import { HeaderComponent } from './components/template/header/header.component';
import { NavComponent } from './components/template/nav/nav.component';
import { ForDirective } from './directives/for.directive';
import { RedDirective } from './directives/red.directive';
import { HomeComponent } from './views/home/home.component';
import { ProductCrudComponent } from './views/product-crud/product-crud.component';

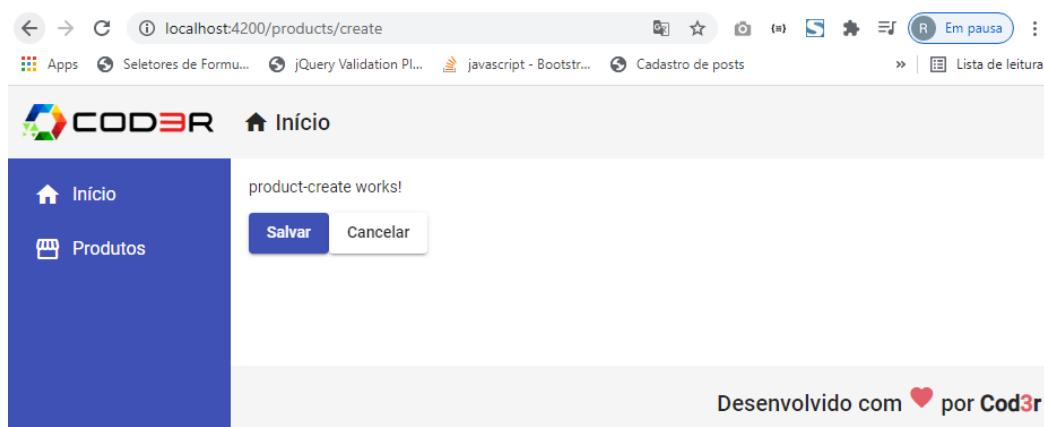
@NgModule({
  declarations: [
    AppComponent,
    HeaderComponent,
    FooterComponent,
    NavComponent,
    HomeComponent,
    ProductCrudComponent,
    RedDirective,
    ForDirective,
    ProductCreateComponent,
  ],
  imports: [
    BrowserModule,
    AppRoutingModule,
    BrowserAnimationsModule,
    MatToolbarModule,
    MatSidenavModule,
    MatListModule,
    MatCardModule,
    MatButtonModule,
    MatSnackBarModule,
  ],
  providers: [],
  bootstrap: [AppComponent],
})
export class AppModule {}
```

- No browser:

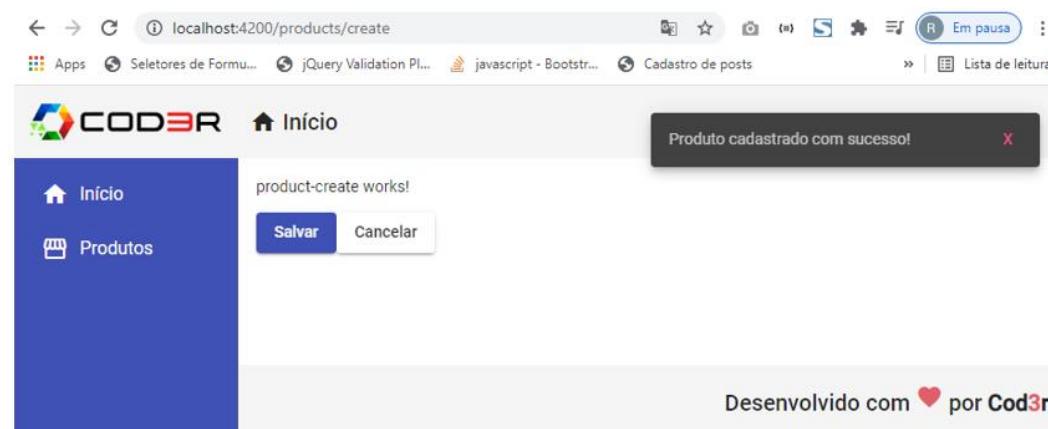
localhost:4200/products



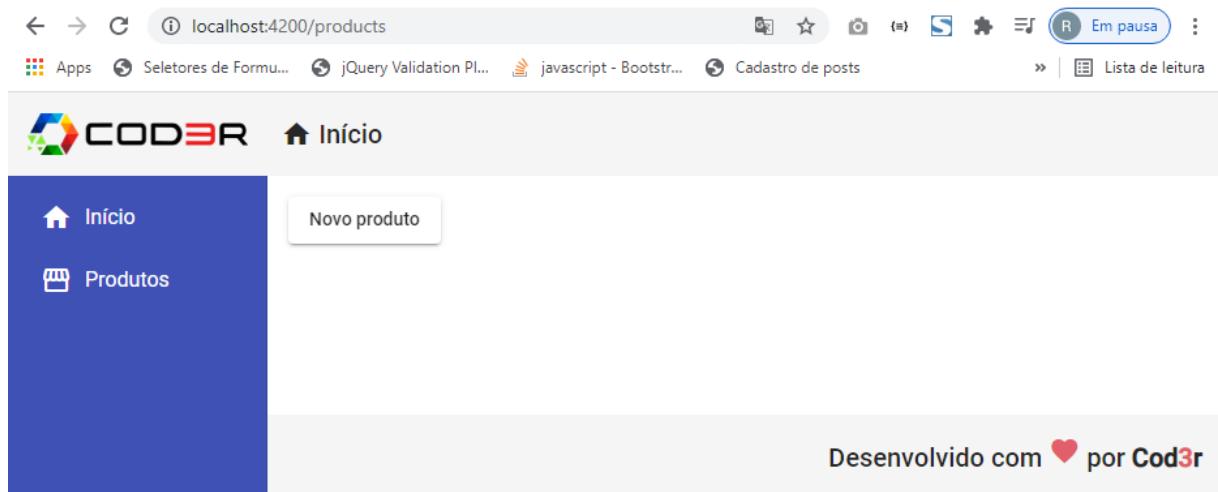
- Clique no botão "Novo produto":



- Clicando no botão "Salvar", é apresentado um toast com uma mensagem, durante 3 segundos, no canto superior direito. Essa mensagem pode ser fechada antes desse tempo clicando em "X" na parte direita do toast:



- Clicando no botão "Cancelar", volta para a tela anterior:



Aula 23 - Método Criar Produto

src\app\app.module.ts

```
import { HttpClientModule } from '@angular/common/http';
import { NgModule } from '@angular/core';
import { MatButtonModule } from '@angular/material/button';
import { MatCardModule } from '@angular/material/card';
import { MatListModule } from '@angular/material/list';
import { MatSidenavModule } from '@angular/material/sidenav';
import { MatSnackBarModule } from '@angular/material/snack-bar';
import { MatToolbarModule } from '@angular/material/toolbar';
import { BrowserModule } from '@angular/platform-browser';
import { BrowserAnimationsModule } from '@angular/platform-browser/animations';
import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { ProductCreateComponent } from './components/products/product-create/product-create.component';
import { FooterComponent } from './components/template/footer/footer.component';
import { HeaderComponent } from './components/template/header/header.component';
import { NavComponent } from './components/template/nav/nav.component';
import { ForDirective } from './directives/for.directive';
import { RedDirective } from './directives/red.directive';
import { HomeComponent } from './views/home/home.component';
import { ProductCrudComponent } from './views/product-crud/product-crud.component';

@NgModule({
  declarations: [
    AppComponent,
    HeaderComponent,
    FooterComponent,
    NavComponent,
    HomeComponent,
    ProductCrudComponent,
    RedDirective,
    ForDirective,
    ProductCreateComponent,
  ],
  imports: [
    BrowserModule,
    AppRoutingModule,
    BrowserAnimationsModule,
    MatToolbarModule,
    MatSidenavModule,
    MatListModule,
    MatCardModule,
    MatButtonModule,
    MatSnackBarModule,
    HttpClientModule,
  ],
  providers: [],
  bootstrap: [AppComponent],
})
export class AppModule {}
```

frontend\src\app\components\products\product.service.ts

```
import { HttpClient } from '@angular/common/http';
import { Injectable } from '@angular/core';
import { MatSnackBar } from '@angular/material/snack-bar';
import { Observable } from 'rxjs';
import { Product } from './product.model';

@Injectable({
  providedIn: 'root',
})
export class ProductService {
  baseUrl = 'http://localhost:3001/products';

  constructor(private snackBar: MatSnackBar, private http: HttpClient) {}

  showMessage(msg: string): void {
    this.snackBar.open(msg, 'X', {
      duration: 3000,
      horizontalPosition: 'right',
      verticalPosition: 'top',
    });
  }

  create(product: Product): Observable<Product> {
    return this.http.post<Product>(this.baseUrl, product);
  }
}
```

frontend\src\app\components\products\product-create\product-create.component.ts

```
import { Component, OnInit } from '@angular/core';
import { Router } from '@angular/router';
import { Product } from './product.model';
import { ProductService } from './product.service';

@Component({
  selector: 'app-product-create',
  templateUrl: './product-create.component.html',
  styleUrls: ['./product-create.component.css'],
})
export class ProductCreateComponent implements OnInit {
  product: Product = {
    name: 'Produto de teste',
    price: 125.98,
  };

  constructor(private productService: ProductService, private router: Router) {}

  ngOnInit(): void {}

  createProduct(): void {
    this.productService.create(this.product).subscribe(() => {
      this.productService.showMessage('Produto cadastrado com sucesso!');
      this.router.navigate(['/products']);
    });
  }

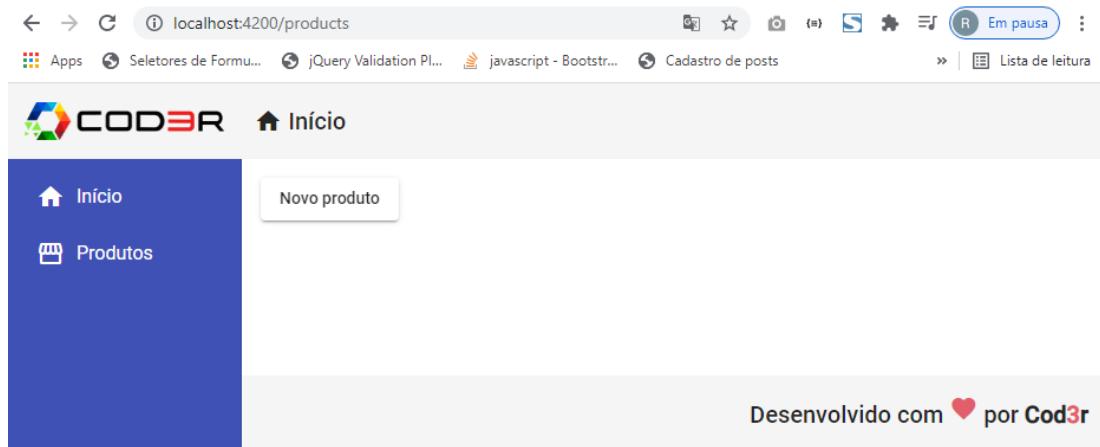
  cancel(): void {
    this.router.navigate(['/products']);
  }
}
```

backend\db.json

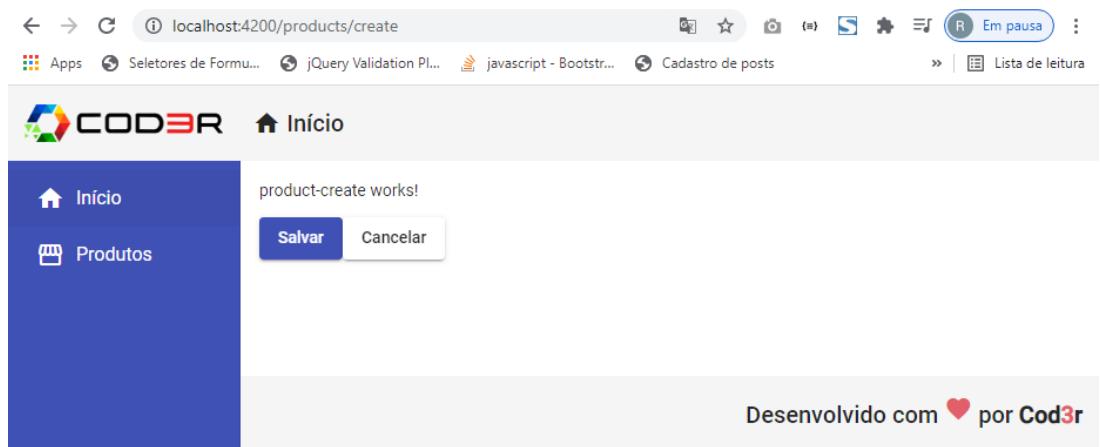
```
{  
  "products": [  
    {  
      "id": 1,  
      "name": "Caneta BIC Preta",  
      "price": 5.89  
    },  
    {  
      "id": 2,  
      "name": "Notebook Mac Pro",  
      "price": 12000.89  
    },  
    {  
      "id": 3,  
      "name": "Samsung S20+",  
      "price": 5000.89  
    }  
  ]  
}
```

- No browser:

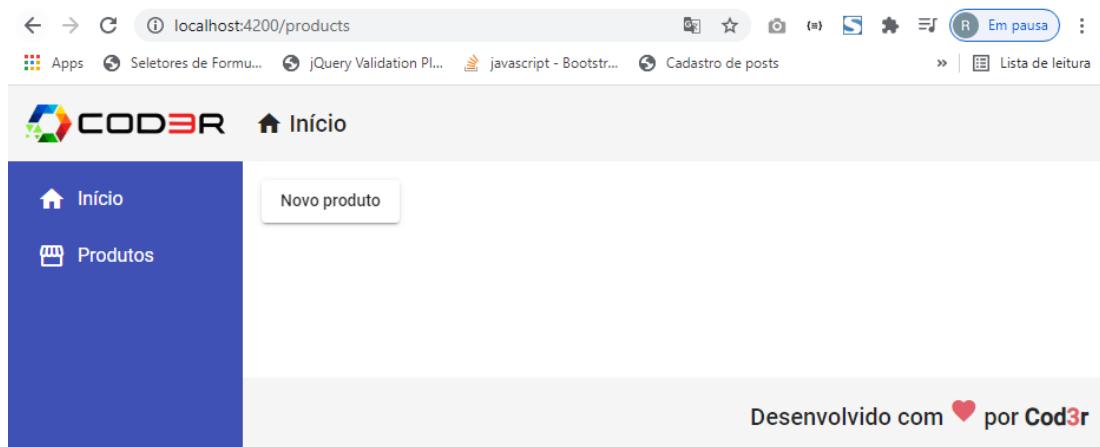
localhost:4200/products



- Clique no botão "Novo produto"



- Clique no botão "Salvar".



backend\db.json

```
{  
  "products": [  
    {  
      "id": 1,  
      "name": "Caneta BIC Preta",  
      "price": 5.89  
    },  
    {  
      "id": 2,  
      "name": "Notebook Mac Pro",  
      "price": 12000.89  
    },  
    {  
      "id": 3,  
      "name": "Samsung S20+",  
      "price": 5000.89  
    },  
    {  
      "name": "Produto de teste",  
      "price": 125.98,  
      "id": 4  
    }  
  ]  
}
```

Aula 24 - Formulário Criar Produto

frontend\src\app\app.module.ts

```
import { HttpClientModule } from '@angular/common/http';
import { NgModule } from '@angular/core';
import { FormsModule } from '@angular/forms';
import { MatButtonModule } from '@angular/material/button';
import { MatCardModule } from '@angular/material/card';
import { MatFormFieldModule } from '@angular/material/form-field';
import { MatInputModule } from '@angular/material/input';
import { MatListModule } from '@angular/material/list';
import { MatSidenavModule } from '@angular/material/sidenav';
import { MatSnackBarModule } from '@angular/material/snack-bar';
import { MatToolbarModule } from '@angular/material/toolbar';
import { BrowserModule } from '@angular/platform-browser';
import { BrowserAnimationsModule } from '@angular/platform-browser/animations';
import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { ProductCreateComponent } from './components/products/product-create/product-create.component';
import { FooterComponent } from './components/template/footer/footer.component';
import { HeaderComponent } from './components/template/header/header.component';
import { NavComponent } from './components/template/nav/nav.component';
import { ForDirective } from './directives/for.directive';
import { RedDirective } from './directives/red.directive';
import { HomeComponent } from './views/home/home.component';
import { ProductCrudComponent } from './views/product-crud/product-crud.component';

@NgModule({
  declarations: [
    AppComponent,
    HeaderComponent,
    FooterComponent,
    NavComponent,
    HomeComponent,
    ProductCrudComponent,
    RedDirective,
    ForDirective,
    ProductCreateComponent,
  ],
  imports: [
    BrowserModule,
    AppRoutingModule,
    BrowserAnimationsModule,
    MatToolbarModule,
    MatSidenavModule,
    MatListModule,
    MatCardModule,
    MatButtonModule,
    MatSnackBarModule,
    HttpClientModule,
    FormsModule,
    MatFormFieldModule,
    MatInputModule,
  ],
})
```

```
providers: [],
bootstrap: [AppComponent],
})
export class AppModule {}
```

Criando o formulário

frontend\src\app\components\products\product-create\product-create.component.html

```
<mat-card>
  <mat-card-title> Novo produto </mat-card-title>
  <form action="" method="post">
    <mat-form-field>
      <input
        matInput
        placeholder="Nome"
        [(ngModel)]="product.name"
        name="name"
      />
    </mat-form-field>
    <mat-form-field>
      <input
        matInput
        placeholder="Preço (R$)"
        [(ngModel)]="product.price"
        name="price"
      />
    </mat-form-field>
  </form>

  <button mat-raised-button (click)="createProduct()" color="primary">
    Salvar
  </button>

  <button mat-raised-button (click)="cancel()">Cancelar</button>
</mat-card>
```

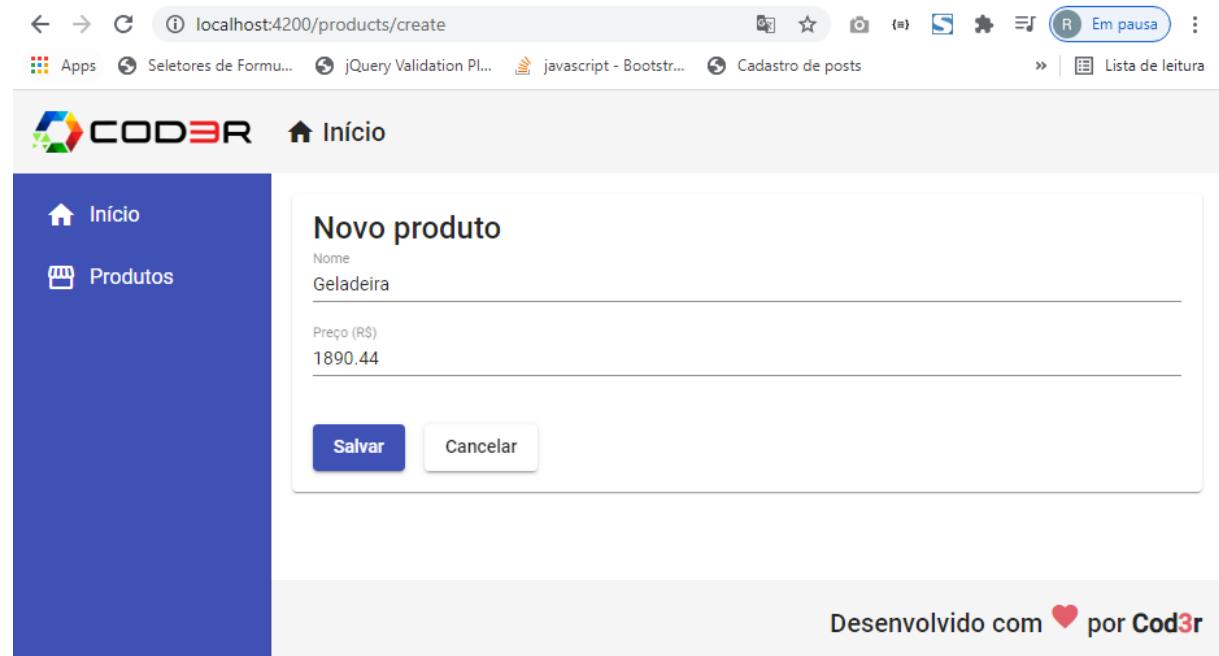
frontend\src\app\components\products\product-create\product-create.component.css

```
form {
  display: flex;
  flex-direction: column;
}
button {
  margin: 20px 15px 0px 0px;
}
```

Cadastrando produtos

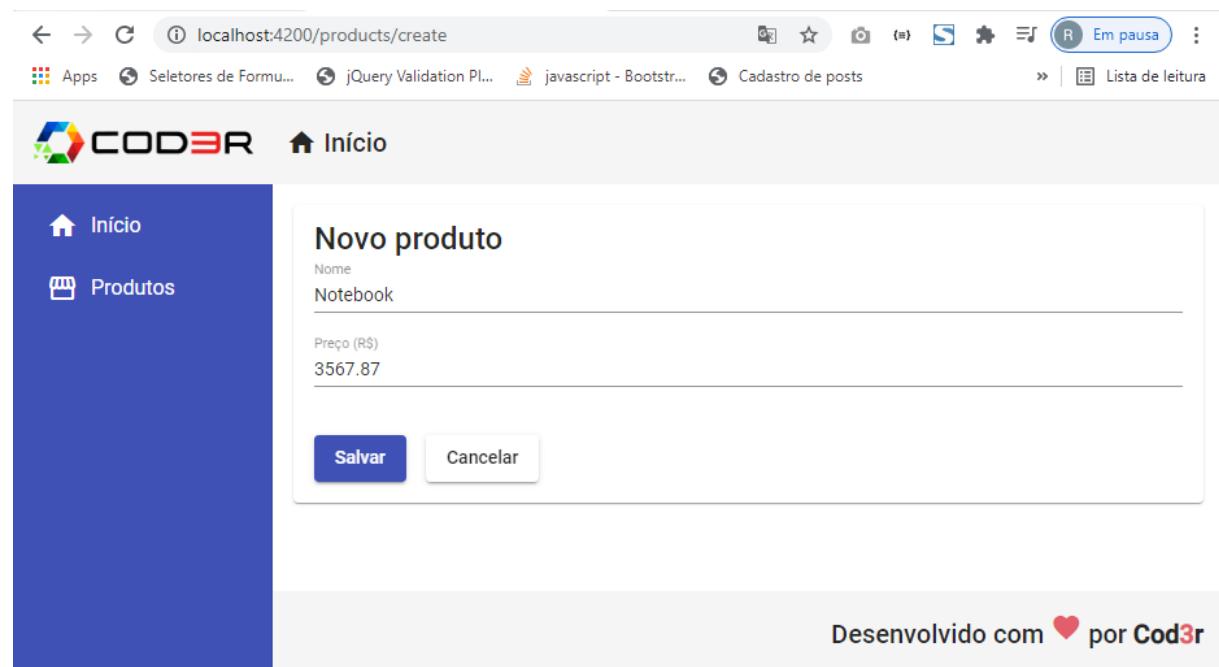
- No browser:

localhost:4200/products/create



A screenshot of a web browser window titled "localhost:4200/products/create". The page has a header with the Cod3r logo and a breadcrumb trail: "Início > Novo produto". On the left, there's a sidebar with "Início" and "Produtos" buttons. The main content area contains a "Novo produto" form with two fields: "Nome" (filled with "Geladeira") and "Preço (R\$)" (filled with "1890.44"). At the bottom are "Salvar" and "Cancelar" buttons. A footer banner at the bottom right says "Desenvolvido com ❤ por Cod3r".

localhost:4200/products/create



A screenshot of a web browser window titled "localhost:4200/products/create". The page has a header with the Cod3r logo and a breadcrumb trail: "Início > Novo produto". On the left, there's a sidebar with "Início" and "Produtos" buttons. The main content area contains a "Novo produto" form with two fields: "Nome" (filled with "Notebook") and "Preço (R\$)" (filled with "3567.87"). At the bottom are "Salvar" and "Cancelar" buttons. A footer banner at the bottom right says "Desenvolvido com ❤ por Cod3r".

backend\db.json

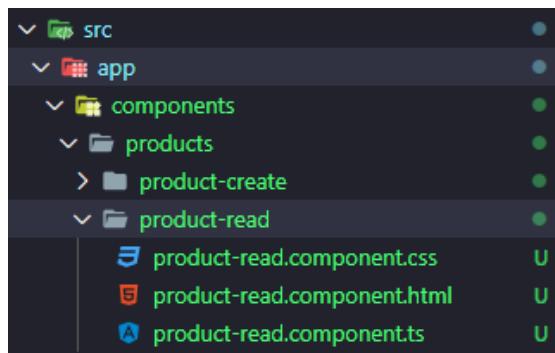
```
{  
  "products": [  
    {  
      "id": 1,  
      "name": "Caneta BIC Preta",  
      "price": 5.89  
    },  
    {  
      "id": 2,  
      "name": "Notebook Mac Pro",  
      "price": 12000.89  
    },  
    {  
      "id": 3,  
      "name": "Samsung S20+",  
      "price": 5000.89  
    },  
    {  
      "name": "Geladeira",  
      "price": "1890.44",  
      "id": 4  
    },  
    {  
      "name": "Notebook",  
      "price": "3567.87",  
      "id": 5  
    }  
  ]  
}
```

Aula 25 - Componente Obter Produto

Criando o componente product-read

```
ng g c components/products/product-read
```

```
C:\coder\angular\crud\frontend>ng g c components/products/product-read
CREATE src/app/components/products/product-read/product-read.component.html (27 bytes)
CREATE src/app/components/products/product-read/product-read.component.ts (298 bytes)
CREATE src/app/components/products/product-read/product-read.component.css (0 bytes)
UPDATE src/app/app.module.ts (2195 bytes)
```



frontend\src\app\views\product-crud\product-crud.component.html

```
<button mat-raised-button (click)="navigateToProductCreate()" color="primary">
  Novo produto
</button>

<app-product-read></app-product-read>
```

frontend\src\app\views\product-crud\product-crud.component.css

```
button {
  margin-bottom: 30px;
}
```

frontend\src\app\components\products\product.service.ts

```
import { HttpClient } from '@angular/common/http';
import { Injectable } from '@angular/core';
import { MatSnackBar } from '@angular/material/snack-bar';
import { Observable } from 'rxjs';
import { Product } from './product.model';

@Injectable({
  providedIn: 'root',
})
export class ProductService {
  baseUrl = 'http://localhost:3001/products';

  constructor(private snackBar: MatSnackBar, private http: HttpClient) {}

  showMessage(msg: string): void {
    this.snackBar.open(msg, 'X', {
      duration: 3000,
      horizontalPosition: 'right',
      verticalPosition: 'top',
    });
  }

  create(product: Product): Observable<Product> {
    return this.http.post<Product>(this.baseUrl, product);
  }

  read(): Observable<Product[]> {
    return this.http.get<Product[]>(this.baseUrl);
  }
}
```

frontend\src\app\components\products\product-read\product-read.component.ts

```
import { Component, OnInit } from '@angular/core';
import { ProductService } from './product.service';
import { Product } from '../../product.model';

@Component({
  selector: 'app-product-read',
  templateUrl: './product-read.component.html',
  styleUrls: ['./product-read.component.css'],
})
export class ProductReadComponent implements OnInit {
  products!: Product[];

  constructor(private productService: ProductService) {}

  ngOnInit(): void {
    this.productService.read().subscribe((products) => {
      this.products = products;
      console.log(products);
    });
  }
}
```

- No browser:

localhost:4200/products

```
1: {id: 2, name: "Notebook Mac Pro", price: 12000.89}
2: {id: 3, name: "Samsung S20+", price: 5000.89}
3: {name: "Geladeira", price: "1890.44", id: 4}
4: {name: "Notebook", price: "3567.87", id: 5}
length: 5
proto_: Array(0)
```

Aula 26 - Exibindo produtos com *ngFor

frontend\src\app\components\products\product-read\product-read.component.html

```
<ul>
<li *ngFor="let product of products">
  {{ product.id }} => {{ product.name }}
</li>
</ul>
```

- No browser:

<http://localhost:4200/products>

The screenshot shows a browser window with the URL localhost:4200/products. The page content is as follows:

```
• 1 => Caneta BIC Preta
• 2 => Notebook Mac Pro
• 3 => Sansung S20+
• 4 => Geladeira
• 5 => Notebook
```

The Angular developer tools console at the bottom shows the following logs:

```
Angular is running in development mode. Call enableProdMode() to enable production mode.
[WDS] Live Reloading enabled.
▶ (5) [{}], [{}], [{}], [{}], [{}]
```

Aula 27 - Gerando componente Tabela

<https://material.angular.io/guide/schematics>

Component schematics

In addition to the installation schematic, Angular Material comes with multiple schematics that can be used to easily generate Material Design components:

Name	Description
address-form	Component with a form group that uses Material Design form controls to prompt for a shipping address
navigation	Creates a component with a responsive Material Design sidenav and a toolbar for showing the app name
dashboard	Component with multiple Material Design cards and menus which are aligned in a grid layout
table	Generates a component with a Material Design data table that supports sorting and pagination
tree	Component that interactively visualizes a nested folder structure by using the <code><mat-tree></code> component

Additionally, the Angular CDK also comes with a collection of component schematics:

Name	Description
drag-drop	Component that uses the <code>@angular/cdk/drag-drop</code> directives for creating an interactive to-do list

Address form schematic

Running the `address-form` schematic generates a new Angular component that can be used to get started with a Material Design form group consisting of:

- Material Design form fields
- Material Design radio controls
- Material Design buttons

```
ng generate @angular/material:address-form <component-name>
```

Navigation schematic

The `navigation` schematic will create a new component that includes a toolbar with the app name, and a responsive side nav based on Material breakpoints.

```
ng generate @angular/material:navigation <component-name>
```

Guide Content

Address form schematic
Navigation schematic
Table schematic
Dashboard schematic
Tree schematic
Drag and Drop schematic

Table schematic

The `table` schematic will create a component that renders an Angular Material `<table>` which has been pre-configured with a datasource for sorting and pagination.

```
ng generate @angular/material:table <component-name>
```

Guide Content

[Address form schematic](#)
[Navigation schematic](#)
Table schematic
[Dashboard schematic](#)
[Tree schematic](#)
[Drag and Drop schematic](#)

Dashboard schematic

The `dashboard` schematic will create a new component that contains a dynamic grid list of Material Design cards.

```
ng generate @angular/material:dashboard <component-name>
```

Tree schematic

The `tree` schematic can be used to quickly generate an Angular component that uses the Angular Material `<mat-tree>` component to visualize a nested folder structure.

```
ng generate @angular/material:tree <component-name>
```

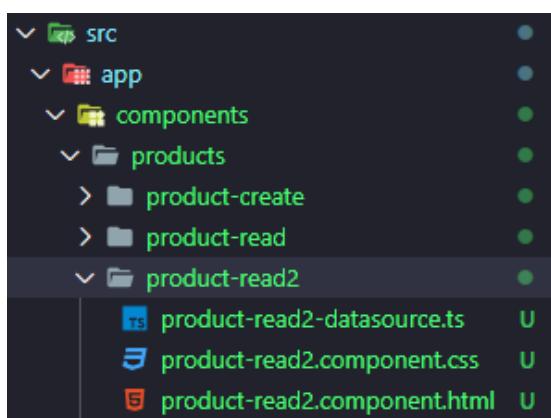
Drag and Drop schematic

The `drag-drop` schematic is provided by the `@angular/cdk` and can be used to generate a component that uses the CDK drag and drop directives.

```
ng generate @angular/cdk:drag-drop <component-name>
```

ng generate @angular/material:table components/products/product-read2

```
C:\coder\angular\crud\frontend>ng generate @angular/material:table components/products/product-read2
CREATE src/app/components/products/product-read2/product-read2.datasource.ts (3676 bytes)
CREATE src/app/components/products/product-read2/product-read2.component.html (853 bytes)
CREATE src/app/components/products/product-read2/product-read2.component.ts (1083 bytes)
CREATE src/app/components/products/product-read2/product-read2.component.css (37 bytes)
UPDATE src/app/app.module.ts (2566 bytes)
```



frontend\src\app\app.module.ts

```
import { HttpClientModule } from '@angular/common/http';
import { NgModule } from '@angular/core';
import { FormsModule } from '@angular/forms';
import { MatButtonModule } from '@angular/material/button';
import { MatCardModule } from '@angular/material/card';
import { MatFormFieldModule } from '@angular/material/form-field';
import { MatInputModule } from '@angular/material/input';
import { MatListModule } from '@angular/material/list';
import { MatSidenavModule } from '@angular/material/sidenav';
import { MatSnackBarModule } from '@angular/material/snack-bar';
import { MatToolbarModule } from '@angular/material/toolbar';
import { BrowserModule } from '@angular/platform-browser';
import { BrowserAnimationsModule } from '@angular/platform-browser/animations';
import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { ProductCreateComponent } from './components/products/product-create/product-create.component';
import { FooterComponent } from './components/template/footer/footer.component';
import { HeaderComponent } from './components/template/header/header.component';
import { NavComponent } from './components/template/nav/nav.component';
import { ForDirective } from './directives/for.directive';
import { RedDirective } from './directives/red.directive';
import { HomeComponent } from './views/home/home.component';
import { ProductCrudComponent } from './views/product-crud/product-crud.component';
import { ProductReadComponent } from './components/products/product-read/product-read.component';
import { ProductRead2Component } from './components/products/product-read2/product-read2.component';
import { MatTableModule } from '@angular/material/table';
import { MatPaginatorModule } from '@angular/material/paginator';
import { MatSortModule } from '@angular/material/sort';

@NgModule({
  declarations: [
    AppComponent,
    HeaderComponent,
    FooterComponent,
    NavComponent,
    HomeComponent,
    ProductCrudComponent,
    RedDirective,
    ForDirective,
    ProductCreateComponent,
    ProductReadComponent,
    ProductRead2Component,
  ],
  imports: [
    BrowserModule,
    AppRoutingModule,
    BrowserAnimationsModule,
    MatToolbarModule,
    MatSidenavModule,
    MatListModule,
    MatCardModule,
    MatButtonModule,
    MatSnackBarModule,
    HttpClientModule,
```

```

FormsModule,
MatFormFieldModule,
MatInputModule,
MatTableModule,
MatPaginatorModule,
MatSortModule,
],
providers: [],
bootstrap: [AppComponent],
})
export class AppModule {}

```

frontend\src\app\views\product-crud\product-crud.component.html

```

<button mat-raised-button (click)="navigateToProductCreate()" color="primary">
  Novo produto
</button>

<app-product-read></app-product-read>

<app-product-read2></app-product-read2>

```

frontend\src\app\components\products\product-read2\product-read2.component.html

```

<div class="mat-elevation-z8">
  <table mat-table class="full-width-table" matSort aria-label="Elements">
    <!-- Id Column -->
    <ng-container matColumnDef="id">
      <th mat-header-cell *matHeaderCellDef mat-sort-header>Id</th>
      <td mat-cell *matCellDef="let row">{{row.id}}</td>
    </ng-container>

    <!-- Name Column -->
    <ng-container matColumnDef="name">
      <th mat-header-cell *matHeaderCellDef mat-sort-header>Name</th>
      <td mat-cell *matCellDef="let row">{{row.name}}</td>
    </ng-container>

    <tr mat-header-row *matHeaderRowDef="displayedColumns"></tr>
    <tr mat-row *matRowDef="let row; columns: displayedColumns;"></tr>
  </table>

  <mat-paginator #paginator
    [length]="dataSource?.data?.length"
    [pageIndex]="0"
    [pageSize]="10"
    [pageSizeOptions]="[5, 10, 20]">
  </mat-paginator>
</div>

```

- No browser:

<http://localhost:4200/products>

The screenshot shows a web browser window with the URL localhost:4200/products in the address bar. The page title is "Produtos". On the left, there's a sidebar with "Início" and "Produtos" buttons. A blue button labeled "Novo produto" is at the top right. Below it is a list of items:

- 1 => Caneta BIC Preta
- 2 => Notebook Mac Pro
- 3 => Samsung S20+
- 4 => Geladeira
- 5 => Notebook

Below this is a table with columns "Id" and "Name". The data is:

Id	Name
1	Hydrogen
2	Helium
3	Lithium
4	Beryllium
5	Boron

At the bottom right, there's a footer note: "Desenvolvido com ❤ por Cod3r".

Aula 28 - Entendendo componente gerado

frontend\src\app\components\products\product-read2\product-read2-datasource.ts

```
import { DataSource } from '@angular/cdk/collections';
import { MatPaginator } from '@angular/material/paginator';
import { MatSort } from '@angular/material/sort';
import { merge, Observable, of as observableOf } from 'rxjs';
import { map } from 'rxjs/operators';

// TODO: Replace this with your own data model type
export interface Product {
  name: string;
  id: number;
  price: number;
}

// TODO: replace this with real data from your application
const EXAMPLE_DATA: Product[] = [
  { id: 1, name: 'Hydrogen', price: 9.99 },
  { id: 2, name: 'Helium', price: 9.99 },
  { id: 3, name: 'Lithium', price: 9.99 },
  { id: 4, name: 'Beryllium', price: 9.99 },
  { id: 5, name: 'Boron', price: 9.99 },
  { id: 6, name: 'Carbon', price: 9.99 },
  { id: 7, name: 'Nitrogen', price: 9.99 },
  { id: 8, name: 'Oxygen', price: 9.99 },
  { id: 9, name: 'Fluorine', price: 9.99 },
  { id: 10, name: 'Neon', price: 9.99 },
  { id: 11, name: 'Sodium', price: 9.99 },
  { id: 12, name: 'Magnesium', price: 9.99 },
  { id: 13, name: 'Aluminum', price: 9.99 },
  { id: 14, name: 'Silicon', price: 9.99 },
  { id: 15, name: 'Phosphorus', price: 9.99 },
  { id: 16, name: 'Sulfur', price: 9.99 },
  { id: 17, name: 'Chlorine', price: 9.99 },
  { id: 18, name: 'Argon', price: 9.99 },
  { id: 19, name: 'Potassium', price: 9.99 },
  { id: 20, name: 'Calcium', price: 9.99 },
];

/**
 * Data source for the ProductRead2 view. This class should
 * encapsulate all logic for fetching and manipulating the displayed data
 * (including sorting, pagination, and filtering).
 */
export class ProductRead2DataSource extends DataSource<Product> {
  data: Product[] = EXAMPLE_DATA;
  paginator: MatPaginator | undefined;
  sort: MatSort | undefined;

  constructor() {
    super();
  }
}
```

```

    /**
     * Connect this data source to the table. The table will only update when
     * the returned stream emits new items.
     * @returns A stream of the items to be rendered.
     */
    connect(): Observable<Product[]> {
        if (this.paginator && this.sort) {
            // Combine everything that affects the rendered data into one update
            // stream for the data-table to consume.
            return merge(
                observableOf(this.data),
                this.paginator.page,
                this.sort.sortChange
            ).pipe(
                map(() => {
                    return this.getPagedData(this.getSortedData([...this.data]));
                })
            );
        } else {
            throw Error(
                'Please set the paginator and sort on the data source before connecting.'
            );
        }
    }

    /**
     * Called when the table is being destroyed. Use this function, to clean up
     * any open connections or free any held resources that were set up during connect.
     */
    disconnect(): void {}

    /**
     * Paginate the data (client-side). If you're using server-side pagination,
     * this would be replaced by requesting the appropriate data from the server.
     */
    private getPagedData(data: Product[]): Product[] {
        if (this.paginator) {
            const startIndex = this.paginator.pageIndex * this.paginator.pageSize;
            return data.splice(startIndex, this.paginator.pageSize);
        } else {
            return data;
        }
    }

    /**
     * Sort the data (client-side). If you're using server-side sorting,
     * this would be replaced by requesting the appropriate data from the server.
     */
    private getSortedData(data: Product[]): Product[] {
        if (!this.sort || !this.sort.active || this.sort.direction === '') {
            return data;
        }

        return data.sort((a, b) => {
            const isAsc = this.sort?.direction === 'asc';

```

```
switch (this.sort?.active) {
  case 'price':
    return compare(a.price, b.price, isAsc);
  case 'name':
    return compare(a.name, b.name, isAsc);
  case 'id':
    return compare(+a.id, +b.id, isAsc);
  default:
    return 0;
}
});
}

/** Simple sort comparator for example ID/Name columns (for client-side sorting). */
function compare(
  a: string | number,
  b: string | number,
  isAsc: boolean
): number {
  return (a < b ? -1 : 1) * (isAsc ? 1 : -1);
}
```

frontend\src\app\components\products\product-read2\product-read2.component.ts

```
import { AfterViewInit, Component, ViewChild } from '@angular/core';
import { MatPaginator } from '@angular/material/paginator';
import { MatSort } from '@angular/material/sort';
import { MatTable } from '@angular/material/table';
import { Product } from './product.model';
import { ProductRead2DataSource } from './product-read2-datasource';

@Component({
  selector: 'app-product-read2',
  templateUrl: './product-read2.component.html',
  styleUrls: ['./product-read2.component.css'],
})
export class ProductRead2Component implements AfterViewInit {
  @ViewChild(MatPaginator) paginator!: MatPaginator;
  @ViewChild(MatSort) sort!: MatSort;
  @ViewChild(MatTable) table!: MatTable<Product>;
  dataSource: ProductRead2DataSource;

  /** Columns displayed in the table. Columns IDs can be added, removed, or reordered. */
  displayedColumns = ['id', 'name', 'price'];

  constructor() {
    this.dataSource = new ProductRead2DataSource();
  }

  ngAfterViewInit(): void {
    this.dataSource.sort = this.sort;
    this.dataSource.paginator = this.paginator;
    this.table.dataSource = this.dataSource;
  }
}
```

frontend\src\app\components\products\product-read2\product-read2.component.html

```
<div class="mat-elevation-z8">
  <table mat-table class="full-width-table" matSort aria-label="Elements">
    <!-- Id Column -->
    <ng-container matColumnDef="id">
      <th mat-header-cell *matHeaderCellDef mat-sort-header>Id</th>
      <td mat-cell *matCellDef="let row">{{ row.id }}</td>
    </ng-container>

    <!-- Name Column -->
    <ng-container matColumnDef="name">
      <th mat-header-cell *matHeaderCellDef mat-sort-header>Nome</th>
      <td mat-cell *matCellDef="let row">{{ row.name }}</td>
    </ng-container>

    <!-- Price Column -->
    <ng-container matColumnDef="price">
      <th mat-header-cell *matHeaderCellDef mat-sort-header>Preço</th>
      <td mat-cell *matCellDef="let row">{{ row.price }}</td>
    </ng-container>

    <tr mat-header-row *matHeaderRowDef="displayedColumns"></tr>
    <tr mat-row *matRowDef="let row; columns: displayedColumns"></tr>
  </table>

  <mat-paginator
    #paginator
    [length]="dataSource?.data?.length"
    [pageIndex]="0"
    [pageSize]="10"
    [pageSizeOptions]="[5, 10, 20]"
  >
  </mat-paginator>
</div>
```

- No browser:

<http://localhost:4200/products>

The screenshot shows a web application interface for managing products. At the top, there is a navigation bar with the 'COD3R' logo and links for 'Início' and 'Produtos'. A blue button labeled 'Novo produto' is visible. Below the navigation, a sidebar on the left contains links for 'Início' and 'Produtos'. The main content area displays a table of products with columns for 'Id', 'Nome', and 'Preço'. The table has 10 rows, each representing a product. At the bottom right of the table, there are pagination controls and a message indicating 10 items per page.

ID	Nome	Preço
1	Hydrogen	9.99
2	Helium	9.99
3	Lithium	9.99
4	Beryllium	9.99
5	Boron	9.99
6	Carbon	9.99
7	Nitrogen	9.99
8	Oxygen	9.99
9	Fluorine	9.99
10	Neon	9.99

Items per page: 10 | 1 – 10 of 20 | < | >

Aula 29 - Tabela Produtos

frontend\src\app\views\product-crud\product-crud.component.html

```
<button mat-raised-button (click)="navigateToProductCreate()" color="primary">
  Novo produto
</button>

<app-product-read></app-product-read>
```

frontend\src\app\components\products\product-read\product-read.component.ts

```
import { Component, OnInit } from '@angular/core';
import { Product } from './product.model';
import { ProductService } from './product.service';

@Component({
  selector: 'app-product-read',
  templateUrl: './product-read.component.html',
  styleUrls: ['./product-read.component.css'],
})
export class ProductReadComponent implements OnInit {
  products!: Product[];
  displayedColumns = ['id', 'name', 'price'];

  constructor(private productService: ProductService) {}

  ngOnInit(): void {
    this.productService.read().subscribe((products) => {
      this.products = products;
    });
  }
}
```

frontend\src\app\components\products\product-read\product-read.component.html

```
<div class="mat-elevation-z4">
  <table mat-table [dataSource]="products">
    <!-- Id Column -->
    <ng-container matColumnDef="id">
      <th mat-header-cell *matHeaderCellDef>Id</th>
      <td mat-cell *matCellDef="let row">{{ row.id }}</td>
    </ng-container>

    <!-- Name Column -->
    <ng-container matColumnDef="name">
      <th mat-header-cell *matHeaderCellDef>Name</th>
      <td mat-cell *matCellDef="let row">{{ row.name }}</td>
    </ng-container>

    <!-- Price Column -->
    <ng-container matColumnDef="price">
      <th mat-header-cell *matHeaderCellDef>Preço</th>
      <td mat-cell *matCellDef="let row">{{ row.price | currency: 'BRL' }}</td>
    </ng-container>

    <tr mat-header-row *matHeaderRowDef="displayedColumns"></tr>
    <tr mat-row *matRowDef="let row; columns: displayedColumns"></tr>
  </table>
</div>
```

frontend\src\app\app.module.ts

```
import { registerLocaleData } from '@angular/common';
import { HttpClientModule } from '@angular/common/http';
import localePt from '@angular/common/locales/pt';
import { LOCALE_ID, NgModule } from '@angular/core';
import { FormsModule } from '@angular/forms';
import { MatButtonModule } from '@angular/material/button';
import { MatCardModule } from '@angular/material/card';
import { MatFormFieldModule } from '@angular/material/form-field';
import { MatInputModule } from '@angular/material/input';
import { MatListModule } from '@angular/material/list';
import { MatPaginatorModule } from '@angular/material/paginator';
import { MatSidenavModule } from '@angular/material/sidenav';
import { MatSnackBarModule } from '@angular/material/snack-bar';
import { MatSortModule } from '@angular/material/sort';
import { MatTableModule } from '@angular/material/table';
import { MatToolbarModule } from '@angular/material/toolbar';
import { BrowserModule } from '@angular/platform-browser';
import { BrowserAnimationsModule } from '@angular/platform-browser/animations';
import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { ProductCreateComponent } from './components/products/product-create/product-create.component';
import { ProductReadComponent } from './components/products/product-read/product-read.component';
import { ProductRead2Component } from './components/products/product-read2/product-read2.component';
```

```
import { FooterComponent } from './components/template/footer/footer.component';
import { HeaderComponent } from './components/template/header/header.component';
import { NavComponent } from './components/template/nav/nav.component';
import { ForDirective } from './directives/for.directive';
import { RedDirective } from './directives/red.directive';
import { HomeComponent } from './views/home/home.component';
import { ProductCrudComponent } from './views/product-crud/product-crud.component';

registerLocaleData(localePt);

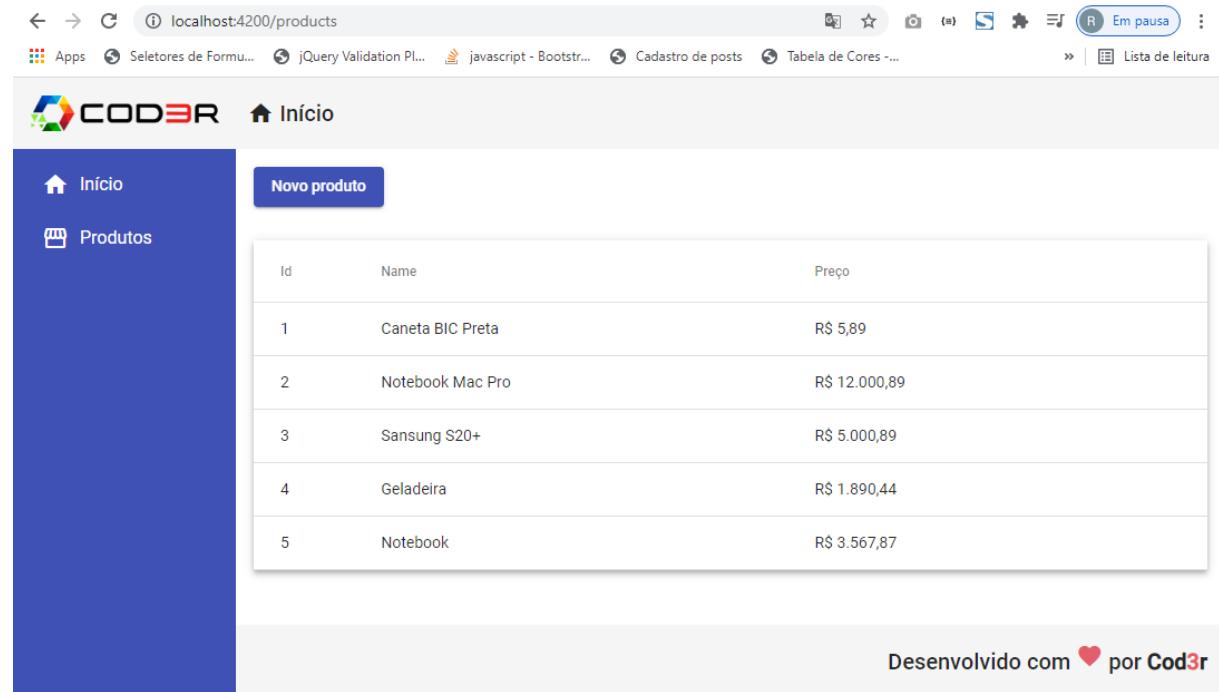
@NgModule({
  declarations: [
    AppComponent,
    HeaderComponent,
    FooterComponent,
    NavComponent,
    HomeComponent,
    ProductCrudComponent,
    RedDirective,
    ForDirective,
    ProductCreateComponent,
    ProductReadComponent,
    ProductRead2Component,
  ],
  imports: [
    BrowserModule,
    AppRoutingModule,
    BrowserAnimationsModule,
    MatToolbarModule,
    MatSidenavModule,
    MatListModule,
    MatCardModule,
    MatButtonModule,
    MatSnackBarModule,
    HttpClientModule,
    FormsModule,
    MatFormFieldModule,
    MatInputModule,
    MatTableModule,
    MatPaginatorModule,
    MatSortModule,
  ],
  providers: [
    {
      provide: LOCALE_ID,
      useValue: 'pt-BR',
    },
  ],
  bootstrap: [AppComponent],
})
export class AppModule {}
```

frontend\src\app\components\products\product-read\product-read.component.css

```
table {  
  width: 100%;  
}  
}
```

- No browser:

<http://localhost:4200/products>



The screenshot shows a web browser window with the URL `localhost:4200/products` in the address bar. The page title is "Início". On the left, there is a sidebar with icons for "Início" and "Produtos". The main content area displays a table titled "Novo produto" with the following data:

ID	Name	Preço
1	Caneta BIC Preta	R\$ 5,89
2	Notebook Mac Pro	R\$ 12.000,89
3	Samsung S20+	R\$ 5.000,89
4	Geladeira	R\$ 1.890,44
5	Notebook	R\$ 3.567,87

At the bottom right of the page, there is a footer message: "Desenvolvido com ❤️ por Cod3r".

Aula 30 - Adicionando coluna Ações

frontend\src\app\components\products\product-read\product-read.component.ts

```
import { Component, OnInit } from '@angular/core';
import { Product } from './product.model';
import { ProductService } from './product.service';

@Component({
  selector: 'app-product-read',
  templateUrl: './product-read.component.html',
  styleUrls: ['./product-read.component.css'],
})
export class ProductReadComponent implements OnInit {
  products!: Product[];
  displayedColumns = ['id', 'name', 'price', 'action'];

  constructor(private productService: ProductService) {}

  ngOnInit(): void {
    this.productService.read().subscribe((products) => {
      this.products = products;
    });
  }
}
```

frontend\src\app\components\products\product-read\product-read.component.html

```
<div class="mat-elevation-z4">
  <table mat-table [dataSource]="products">
    <!-- Id Column -->
    <ng-container matColumnDef="id">
      <th mat-header-cell *matHeaderCellDef>Id</th>
      <td mat-cell *matCellDef="let row">{{ row.id }}</td>
    </ng-container>

    <!-- Name Column -->
    <ng-container matColumnDef="name">
      <th mat-header-cell *matHeaderCellDef>Name</th>
      <td mat-cell *matCellDef="let row">{{ row.name }}</td>
    </ng-container>

    <!-- Price Column -->
    <ng-container matColumnDef="price">
      <th mat-header-cell *matHeaderCellDef>Preço</th>
      <td mat-cell *matCellDef="let row">{{ row.price | currency: 'BRL' }}</td>
    </ng-container>

    <!-- Action Column -->
    <ng-container matColumnDef="action">
      <th mat-header-cell *matHeaderCellDef>Ações</th>
      <td mat-cell *matCellDef="let row">
        <a routerLink="" class="edit">
          <i class="material-icons"> edit </i>
        </a>
        <a routerLink="" class="delete">
          <i class="material-icons"> delete </i>
        </a>
      </td>
    </ng-container>

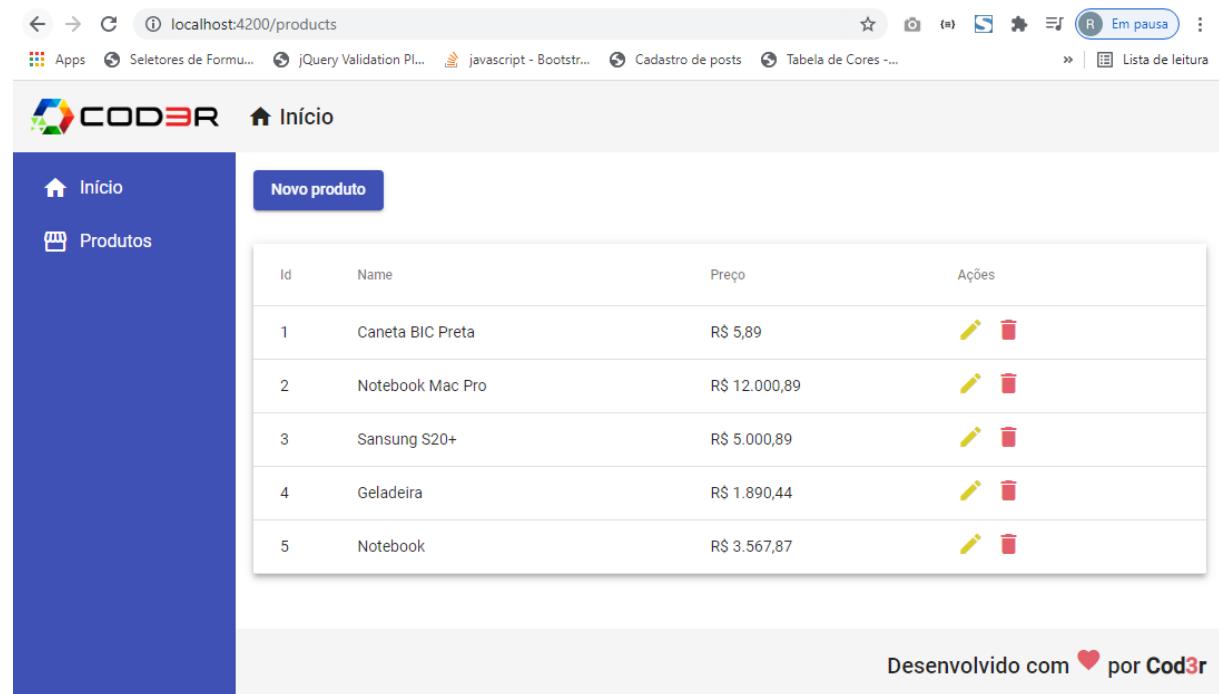
    <tr mat-header-row *matHeaderRowDef="displayedColumns"></tr>
    <tr mat-row *matRowDef="let row; columns: displayedColumns"></tr>
  </table>
</div>
```

frontend\src\app\components\products\product-read\product-read.component.css

```
table {  
  width: 100%;  
}  
  
.edit {  
  margin-right: 10px;  
}  
  
.edit > i {  
  color: #d9cd26;  
}  
  
.delete > i {  
  color: #e35e6b;  
}
```

- No browser:

localhost:4200/products



The screenshot shows a web browser window with the URL 'localhost:4200/products'. The page has a dark blue sidebar on the left with icons for 'Início' and 'Produtos'. The main content area has a title 'Novo produto' and a table with the following data:

ID	Name	Preço	Ações
1	Caneta BIC Preta	R\$ 5,89	
2	Notebook Mac Pro	R\$ 12.000,89	
3	Samsung S20+	R\$ 5.000,89	
4	Geladeira	R\$ 1.890,44	
5	Notebook	R\$ 3.567,87	

At the bottom right of the main content area, there is a footer message: 'Desenvolvido com ❤️ por Cod3r'.

Aula 31 - Novos métodos em Produto Service

frontend\src\app\components\products\product.service.ts

```
import { HttpClient } from '@angular/common/http';
import { Injectable } from '@angular/core';
import { MatSnackBar } from '@angular/material/snack-bar';
import { Observable } from 'rxjs';
import { Product } from './product.model';

@Injectable({
  providedIn: 'root',
})
export class ProductService {
  baseUrl = 'http://localhost:3001/products';

  constructor(private snackBar: MatSnackBar, private http: HttpClient) {}

  showMessage(msg: string): void {
    this.snackBar.open(msg, 'X', {
      duration: 3000,
      horizontalPosition: 'right',
      verticalPosition: 'top',
    });
  }

  create(product: Product): Observable<Product> {
    return this.http.post<Product>(this.baseUrl, product);
  }

  read(): Observable<Product[]> {
    return this.http.get<Product[]>(this.baseUrl);
  }

  readById(id: string): Observable<Product> {
    const url = `${this.baseUrl}/${id}`;
    return this.http.get<Product>(url);
  }

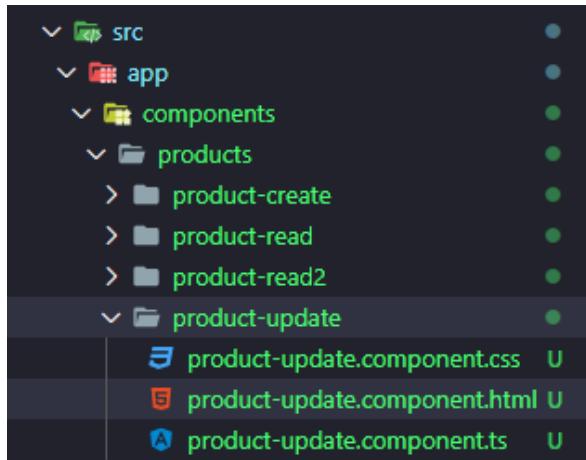
  update(product: Product): Observable<Product> {
    const url = `${this.baseUrl}/${product.id}`;
    return this.http.put<Product>(url, product);
  }
}
```

Aula 32 - Componente Atualizar Produto - Parte 1

Criando o componente

```
ng g c components/products/product-update
```

```
C:\coder\angular\crud\frontend>ng g c components/products/product-update
CREATE src/app/components/products/product-update/product-update.component.html (29 bytes)
CREATE src/app/components/products/product-update/product-update.component.ts (306 bytes)
CREATE src/app/components/products/product-update/product-update.component.css (0 bytes)
UPDATE src/app/app.module.ts (2911 bytes)
```



frontend\src\app\app-routing.module.ts

```
import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';
import { ProductCreateComponent } from './components/products/product-create/product-create.component';
import { ProductUpdateComponent } from './components/products/product-update/product-update.component';
import { HomeComponent } from './views/home/home.component';
import { ProductCrudComponent } from './views/product-crud/product-crud.component';

const routes: Routes = [
  {
    path: '',
    component: HomeComponent,
  },
  {
    path: 'products',
    component: ProductCrudComponent,
  },
  {
    path: 'products/create',
    component: ProductCreateComponent,
  },
  {
    path: 'products/update/:id',
    component: ProductUpdateComponent,
  },
];
;

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

frontend\src\app\components\products\product-read\product-read.component.html

```
<div class="mat-elevation-z4">
  <table mat-table [dataSource]="products">
    <!-- Id Column -->
    <ng-container matColumnDef="id">
      <th mat-header-cell *matHeaderCellDef>Id</th>
      <td mat-cell *matCellDef="let row">{{ row.id }}</td>
    </ng-container>

    <!-- Name Column -->
    <ng-container matColumnDef="name">
      <th mat-header-cell *matHeaderCellDef>Name</th>
      <td mat-cell *matCellDef="let row">{{ row.name }}</td>
    </ng-container>

    <!-- Price Column -->
    <ng-container matColumnDef="price">
      <th mat-header-cell *matHeaderCellDef>Preço</th>
      <td mat-cell *matCellDef="let row">{{ row.price | currency: 'BRL' }}</td>
    </ng-container>

    <!-- Action Column -->
    <ng-container matColumnDef="action">
      <th mat-header-cell *matHeaderCellDef>Ações</th>
      <td mat-cell *matCellDef="let row">
        <a routerLink="/products/update/{{ row.id }}" class="edit">
          <i class="material-icons">edit</i>
        </a>
        <a routerLink="" class="delete">
          <i class="material-icons">delete</i>
        </a>
      </td>
    </ng-container>

    <tr mat-header-row *matHeaderRowDef="displayedColumns"></tr>
    <tr mat-row *matRowDef="let row; columns: displayedColumns"></tr>
  </table>
</div>
```

Aula 33 - Componente Atualizar Produto - Parte 2

frontend\src\app\components\products\product.service.ts

```
import { HttpClient } from '@angular/common/http';
import { Injectable } from '@angular/core';
import { MatSnackBar } from '@angular/material/snack-bar';
import { Observable } from 'rxjs';
import { Product } from './product.model';

@Injectable({
  providedIn: 'root',
})
export class ProductService {
  baseUrl = 'http://localhost:3001/products';

  constructor(private snackBar: MatSnackBar, private http: HttpClient) {}

  showMessage(msg: string): void {
    this.snackBar.open(msg, 'X', {
      duration: 3000,
      horizontalPosition: 'right',
      verticalPosition: 'top',
    });
  }

  create(product: Product): Observable<Product> {
    return this.http.post<Product>(this.baseUrl, product);
  }

  read(): Observable<Product[]> {
    return this.http.get<Product[]>(this.baseUrl);
  }

  readById(id: string): Observable<Product> {
    const url = `${this.baseUrl}/${id}`;
    return this.http.get<Product>(url);
  }

  update(product: Product): Observable<Product> {
    const url = `${this.baseUrl}/${product.id}`;
    return this.http.put<Product>(url, product);
  }
}
```

frontend\src\app\components\products\product-update\product-update.component.ts

```
import { Component, OnInit } from '@angular/core';
import { ActivatedRoute, Router } from '@angular/router';
import { Product } from './product.model';
import { ProductService } from './product.service';

@Component({
  selector: 'app-product-update',
  templateUrl: './product-update.component.html',
  styleUrls: ['./product-update.component.css'],
})
export class ProductUpdateComponent implements OnInit {
  product!: Product;

  constructor(
    private productService: ProductService,
    private router: Router,
    private route: ActivatedRoute
  ) {}

  ngOnInit(): void {
    const id = this.route.snapshot.paramMap.get('id') || '';
    this.productService.readById(id).subscribe((product) => {
      this.product = product;
    });
  }

  updateProduct(): void {}

  cancel(): void {
    this.router.navigate(['/products']);
  }
}
```

- No browser:

localhost:4200/products/update/1

The screenshot shows a web browser window with the URL `localhost:4200/products/update/1`. The page title is "Alterar produto". On the left, there is a sidebar with icons for "Início" and "Produtos". The main content area contains fields for "Nome" (filled with "Caneta BIC Preta") and "Preço (R\$)" (filled with "5.89"). At the bottom are two buttons: "Salvar" (in blue) and "Cancelar". A footer banner at the bottom right says "Desenvolvido com ❤️ por Cod3r".

The screenshot shows a web browser window with the URL `localhost:4200/products/update/1`. The page title is "Alterar produto". The "Nome" field now contains "Caneta BIC Preta - Escrita fina" and the "Preço (R\$)" field contains "3.65". The rest of the interface is identical to the previous screenshot, including the sidebar, buttons, and footer banner.

- Clique no botão "Salvar":

localhost:4200/products

The screenshot shows a web application interface for managing products. At the top, there is a navigation bar with links to 'localhost:4200/products', 'Seletores de Formu...', 'jQuery Validation Pl...', 'javascript - Bootstr...', 'Cadastro de posts', 'Tabela de Cores -...', '»', and 'Lista de leitura'. A 'Em pausa' button is also present. Below the navigation is a header with the Cod3r logo and a 'Início' button. On the left, a sidebar has 'Início' and 'Produtos' buttons. The main content area has a 'Novo produto' button and a table listing five products:

ID	Name	Preço	Ações
1	Caneta BIC Preta - Escrita fina	R\$ 3,65	
2	Notebook Mac Pro	R\$ 12.000,89	
3	Samsung S20+	R\$ 5.000,89	
4	Geladeira	R\$ 1.890,44	
5	Notebook	R\$ 3.567,87	

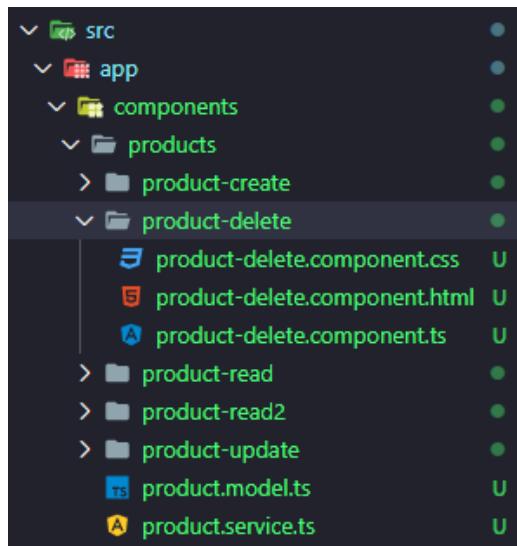
At the bottom right of the main content area, there is a footer message: 'Desenvolvido com ❤️ por Cod3r'.

Aulas 34, 35, 36 e 37 - Método Excluir Produto

Criando o componente

```
ng g c components/products/product-delete
```

```
C:\coder\angular\crud\frontend>ng g c components/products/product-delete
CREATE src/app/components/products/product-delete/product-delete.component.html (29 bytes)
CREATE src/app/components/products/product-delete/product-delete.component.ts (306 bytes)
CREATE src/app/components/products/product-delete/product-delete.component.css (0 bytes)
UPDATE src/app/app.module.ts (3043 bytes)
```



frontend\src\app\components\products\product.service.ts

```
import { HttpClient } from '@angular/common/http';
import { Injectable } from '@angular/core';
import { MatSnackBar } from '@angular/material/snack-bar';
import { Observable } from 'rxjs';
import { Product } from './product.model';

@Injectable({
  providedIn: 'root',
})
export class ProductService {
  baseUrl = 'http://localhost:3001/products';

  constructor(private snackBar: MatSnackBar, private http: HttpClient) {}

  showMessage(msg: string): void {
    this.snackBar.open(msg, 'X', {
      duration: 3000,
      horizontalPosition: 'right',
      verticalPosition: 'top',
    });
  }

  create(product: Product): Observable<Product> {
    return this.http.post<Product>(this.baseUrl, product);
  }

  read(): Observable<Product[]> {
    return this.http.get<Product[]>(this.baseUrl);
  }

  readById(id: number): Observable<Product> {
    const url = `${this.baseUrl}/${id}`;
    return this.http.get<Product>(url);
  }

  update(product: Product): Observable<Product> {
    const url = `${this.baseUrl}/${product.id}`;
    return this.http.put<Product>(url, product);
  }

  delete(id: number): Observable<Product> {
    const url = `${this.baseUrl}/${id}`;
    return this.http.delete<Product>(url);
  }
}
```

frontend\src\app\components\products\product-delete\product-delete.component.ts

```
import { Component, OnInit } from '@angular/core';
import { ActivatedRoute, Router } from '@angular/router';
import { Product } from './product.model';
import { ProductService } from './product.service';

@Component({
  selector: 'app-product-delete',
  templateUrl: './product-delete.component.html',
  styleUrls: ['./product-delete.component.css'],
})
export class ProductDeleteComponent implements OnInit {
  product!: Product;

  constructor(
    private productService: ProductService,
    private router: Router,
    private route: ActivatedRoute
  ) {}

  ngOnInit(): void {
    const id = +this.route.snapshot.paramMap.get('id');
    this.productService.readById(id).subscribe((product) => {
      this.product = product;
    });
  }

  deleteProduct(): void {
    this.productService.delete(this.product.id).subscribe(() => {
      this.productService.showMessage('Produto excluido com sucesso!');
      this.router.navigate(['/products']);
    });
  }

  cancel(): void {
    this.router.navigate(['/products']);
  }
}
```

frontend\src\app\app-routing.module.ts

```
import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';
import { ProductCreateComponent } from './components/products/product-create/product-create.component';
import { ProductDeleteComponent } from './components/products/product-delete/product-delete.component';
import { ProductUpdateComponent } from './components/products/product-update/product-update.component';
import { HomeComponent } from './views/home/home.component';
import { ProductCrudComponent } from './views/product-crud/product-crud.component';

const routes: Routes = [
  {
    path: '',
    component: HomeComponent,
  },
  {
    path: 'products',
    component: ProductCrudComponent,
  },
  {
    path: 'products/create',
    component: ProductCreateComponent,
  },
  {
    path: 'products/update/:id',
    component: ProductUpdateComponent,
  },
  {
    path: 'products/delete/:id',
    component: ProductDeleteComponent,
  },
];
@NgModule({
  imports: [RouterModule.forRoot(routes)],
  exports: [RouterModule],
})
export class AppRoutingModule {}
```

frontend\src\app\components\products\product-read\product-read.component.html

```
<div class="mat-elevation-z4">
  <table mat-table [dataSource]="products">
    <!-- Id Column -->
    <ng-container matColumnDef="id">
      <th mat-header-cell *matHeaderCellDef>Id</th>
      <td mat-cell *matCellDef="let row">{{ row.id }}</td>
    </ng-container>

    <!-- Name Column -->
    <ng-container matColumnDef="name">
      <th mat-header-cell *matHeaderCellDef>Name</th>
      <td mat-cell *matCellDef="let row">{{ row.name }}</td>
    </ng-container>

    <!-- Price Column -->
    <ng-container matColumnDef="price">
      <th mat-header-cell *matHeaderCellDef>Preço</th>
      <td mat-cell *matCellDef="let row">{{ row.price | currency: 'BRL' }}</td>
    </ng-container>

    <!-- Action Column -->
    <ng-container matColumnDef="action">
      <th mat-header-cell *matHeaderCellDef>Ações</th>
      <td mat-cell *matCellDef="let row">
        <a routerLink="/products/update/{{ row.id }}" class="edit">
          <i class="material-icons">edit</i>
        </a>
        <a routerLink="/products/delete/{{ row.id }}" class="delete">
          <i class="material-icons">delete</i>
        </a>
      </td>
    </ng-container>

    <tr mat-header-row *matHeaderRowDef="displayedColumns"></tr>
    <tr mat-row *matRowDef="let row; columns: displayedColumns"></tr>
  </table>
</div>
```

frontend\src\app\components\products\product-delete\product-delete.component.ts

```
form {
  display: flex;
  flex-direction: column;
  margin-top: 20px;
}

button {
  margin: 20px 15px 0px 0px;
}
```

Acertos

frontend\src\app\components\products\product-update\product-update.component.html

```
<mat-card>
  <mat-card-title> Alterar produto </mat-card-title>
  <form action="" method="post">
    <mat-form-field>
      <input
        matInput
        placeholder="Nome"
        [(ngModel)]="product.name"
        name="name"
      />
    </mat-form-field>
    <mat-form-field>
      <input
        matInput
        placeholder="Preço (R$)"
        [(ngModel)]="product.price"
        name="price"
      />
    </mat-form-field>
  </form>

  <button mat-raised-button (click)="updateProduct()" color="primary">
    Atualizar
  </button>

  <button mat-raised-button (click)="cancel()">Cancelar</button>
</mat-card>
```

frontend\src\app\components\products\product-update\product-update.component.ts

```
import { Component, OnInit } from '@angular/core';
import { ActivatedRoute, Router } from '@angular/router';
import { Product } from './product.model';
import { ProductService } from './product.service';

@Component({
  selector: 'app-product-update',
  templateUrl: './product-update.component.html',
  styleUrls: ['./product-update.component.css'],
})
export class ProductUpdateComponent implements OnInit {
  product!: Product;

  constructor(
    private productService: ProductService,
    private router: Router,
    private route: ActivatedRoute
  ) {}

  ngOnInit(): void {
    const id = +this.route.snapshot.paramMap.get('id');
    this.productService.readById(id).subscribe((product) => {
      this.product = product;
    });
  }

  updateProduct(): void {
    this.productService.update(this.product).subscribe(() => {
      this.productService.showMessage('Produto atualizado com sucesso!');
      this.router.navigate(['/products']);
    });
  }

  cancel(): void {
    this.router.navigate(['/products']);
  }
}
```

- No browser:

localhost:4200/products

The screenshot shows a web browser window with the URL `localhost:4200/products`. The page has a header with the Cod3r logo and a navigation bar with links for 'Início' and 'Produtos'. A blue button labeled 'Novo produto' is visible. The main content is a table listing five products:

ID	Name	Preço	Ações
1	Caneta BIC Preta - Escrita fina	R\$ 3,65	
2	Notebook Mac Pro	R\$ 12.000,89	
3	Sansung S20+	R\$ 5.000,89	
4	Geladeira	R\$ 1.890,44	
5	Notebook	R\$ 3.567,87	

At the bottom right, there is a footer message: 'Desenvolvido com ❤️ por Cod3r'.

`http://localhost:4200/products/delete/5`

The screenshot shows a web browser window with the URL `http://localhost:4200/products/delete/5`. The page has a header with the Cod3r logo and a navigation bar with links for 'Início' and 'Produtos'. The main content is a confirmation dialog titled 'Excluir produto' with the following details:

Nome: Notebook
Preço (R\$): 3567,87

At the bottom are two buttons: 'Excluir' and 'Cancelar'.

At the bottom right, there is a footer message: 'Desenvolvido com ❤️ por Cod3r'.

localhost:4200/products

The screenshot shows a web application interface. At the top, there is a browser-like header with icons for back, forward, search, and other controls. The URL bar shows "localhost:4200/products". Below the header, there is a navigation bar with links for "Apps", "Seletores de Formu...", "jQuery Validation Pl...", "javascript - Bootstr...", "Cadastro de posts", "Tabela de Cores -...", "»", and "Lista de leitura". On the far right of the header, there is a button labeled "Em pausa" with a play/pause icon.

The main content area has a header with the "COD3R" logo and a "Início" button. To the left, there is a sidebar with two items: "Início" and "Produtos". The main content area contains a table titled "Novo produto" with the following data:

	Id	Name	Preço	Ações
	1	Caneta BIC Preta - Escrita fina	R\$ 3,65	
	2	Notebook Mac Pro	R\$ 12.000,89	
	3	Sansung S20+	R\$ 5.000,89	
	4	Geladeira	R\$ 1.890,44	

At the bottom right of the main content area, there is a footer message: "Desenvolvido com ❤️ por Cod3r".

Aula 38 - Exibindo as mensagens de erro

frontend\src\styles.css

```
/* You can add global styles to this file, and also import other style files */

html,
body {
  height: 100%;
}
body {
  margin: 0;
  font-family: Roboto, 'Helvetica Neue', sans-serif;
}

.red {
  color: rgb(227, 94, 107);
}

.v-middle {
  vertical-align: middle;
}

.msg-success {
  background-color: #28a745;
  color: #fff;
}

.msg-error {
  background-color: #e35e6b;
  color: #fff;
}
```

frontend\src\app\components\products\product.service.ts

```
import { HttpClient } from '@angular/common/http';
import { Injectable } from '@angular/core';
import { MatSnackBar } from '@angular/material/snack-bar';
import { EMPTY, Observable } from 'rxjs';
import { catchError, map } from 'rxjs/operators';
import { Product } from './product.model';

@Injectable({
  providedIn: 'root',
})
export class ProductService {
  baseUrl = 'http://localhost:3001/products';

  constructor(private snackBar: MatSnackBar, private http: HttpClient) {}

  showMessage(msg: string, isError: boolean = false): void {
    this.snackBar.open(msg, 'X', {
      duration: 3000,
      horizontalPosition: 'right',
      verticalPosition: 'top',
      panelClass: isError ? 'msg-error' : 'msg-success',
    });
  }

  create(product: Product): Observable<Product> {
    return this.http.post<Product>(this.baseUrl, product).pipe(
      map((obj) => obj),
      catchError((e) => this.errorHandler(e))
    );
  }

  read(): Observable<Product[]> {
    return this.http.get<Product[]>(this.baseUrl).pipe(
      map((obj) => obj),
      catchError((e) => this.errorHandler(e))
    );
  }

  readById(id: number): Observable<Product> {
    const url = `${this.baseUrl}/${id}`;
    return this.http.get<Product>(url).pipe(
      map((obj) => obj),
      catchError((e) => this.errorHandler(e))
    );
  }

  update(product: Product): Observable<Product> {
    const url = `${this.baseUrl}/${product.id}`;
    return this.http.put<Product>(url, product).pipe(
```

```

        map((obj) => obj),
        catchError((e) => this.errorHandler(e))
    );
}

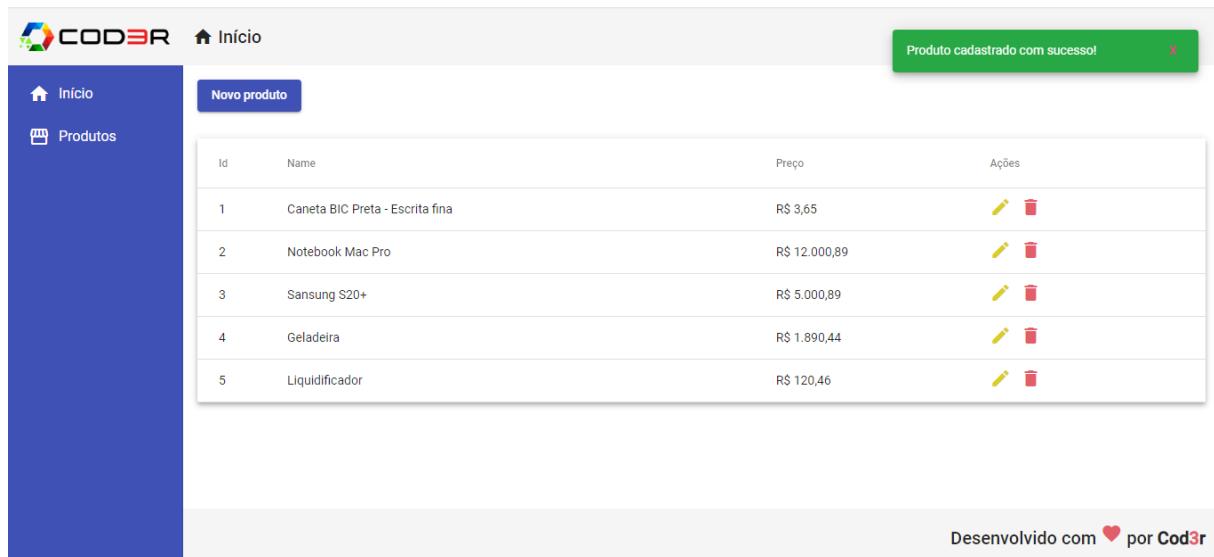
delete(id: number): Observable<Product> {
    const url = `${this.baseUrl}/${id}`;
    return this.http.delete<Product>(url).pipe(
        map((obj) => obj),
        catchError((e) => this.errorHandler(e))
    );
}

errorHandler(e: any): Observable<any> {
    this.showMessage('Ocorreu um erro!', true);
    return EMPTY;
}
}

```

Cadastrando um produto

- No browser:



The screenshot shows a web application interface for managing products. On the left, there's a sidebar with 'Início' and 'Produtos' buttons. The main area has a 'Novo produto' button. A green success message 'Produto cadastrado com sucesso!' is displayed at the top right. Below it is a table with columns: Id, Name, Preço, and Ações. The table contains five rows of product data.

Id	Name	Preço	Ações
1	Caneta BIC Preta - Escrita fina	R\$ 3,65	
2	Notebook Mac Pro	R\$ 12.000,89	
3	Samsung S20+	R\$ 5.000,89	
4	Geladeira	R\$ 1.890,44	
5	Liquidificador	R\$ 120,46	

At the bottom right, there's a footer message 'Desenvolvido com ❤️ por Cod3r'.

- Interrompa a conexão com o servidor de backend e tente cadastrar um novo produto:

localhost:4200/products/create

Início

Novo produto

Nome
Ventilador com pedestal

Preço (R\$)
145.30

Salvar Cancelar

Desenvolvido com ❤ por Cod3r

localhost:4200/products/create

Início

Novo produto

Nome
Ventilador com pedestal

Preço (R\$)
145.30

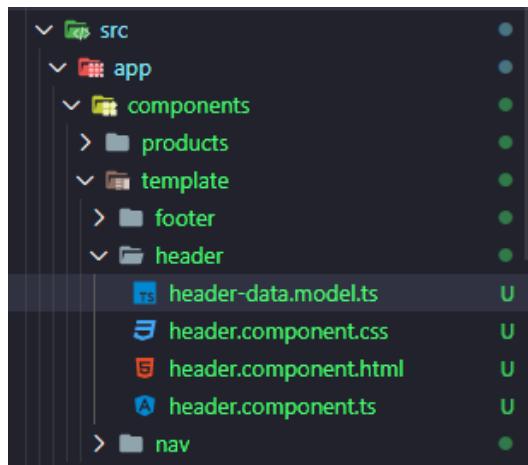
Salvar Cancelar

Ocorreu um erro!

Desenvolvido com ❤ por Cod3r

Aula 39 - Atualizando títulos

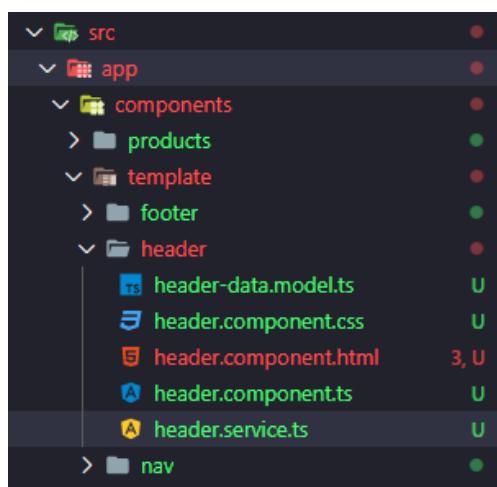
- Adicione à pasta "frontend\src\app\components\template\header" um arquivo chamado "header-data.model.ts":



Criando o service

```
ng g s components/template/header/header
```

```
C:\coder\angular\crud\frontend>ng g s components/template/header/header
CREATE src/app/components/template/header/header.service.ts (135 bytes)
```



frontend\src\app\components\template\header\header.service.ts

```
import { Injectable } from '@angular/core';
import { BehaviorSubject } from 'rxjs';
import { HeaderData } from './header-data.model';

@Injectable({
  providedIn: 'root',
})

export class HeaderService {
  private _headerData = new BehaviorSubject<HeaderData>({
    title: 'Início',
    icon: 'home',
    routeUrl: '',
  });

  constructor() {}

  get headerData(): HeaderData {
    return this._headerData.value;
  }

  set headerData(headerData: HeaderData) {
    this._headerData.next(headerData);
  }
}
```

frontend\src\app\components\template\header\header.component.ts

```
import { Component, OnInit } from '@angular/core';
import { HeaderService } from './header.service';

@Component({
  selector: 'app-header',
  templateUrl: './header.component.html',
  styleUrls: ['./header.component.css'],
})
export class HeaderComponent implements OnInit {
  constructor(private headerService: HeaderService) {}

  ngOnInit(): void {}

  get title(): string {
    return this.headerService.headerData.title;
  }

  get icon(): string {
    return this.headerService.headerData.icon;
  }

  get routeUrl(): string {
    return this.headerService.headerData.routeUrl;
  }
}
```

frontend\src\app\components\template\header\header.component.html

```
<mat-toolbar class="header mat-elevation-z4">
  <span>
    <a routerLink="/">
      
    </a>
  </span>
  <span class="title-group">
    <a routerLink="{{ routeUrl }}">
      <i class="material-icons">
        {{ icon }}
      </i>
      {{ title }}
    </a>
  </span>
</mat-toolbar>
```

frontend\src\app\components\template\header\header.component.css

```
.header {
  display: flex;
  align-items: center;
}
.header a {
  display: flex;
  align-items: center;
  text-decoration: none;
  color: #333;
}
.header .logo {
  max-height: 32px;
}
.header .title-group {
  padding-left: 25px;
}
.header .title-group i {
  padding-right: 5px;
}
```

frontend\src\app\views\product-crud\product-crud.component.ts

```
import { Component, OnInit } from '@angular/core';
import { Router } from '@angular/router';
import { HeaderService } from '../../components/template/header/header.service';

@Component({
  selector: 'app-product-crud',
  templateUrl: './product-crud.component.html',
  styleUrls: ['./product-crud.component.css'],
})
export class ProductCrudComponent implements OnInit {
  constructor(private router: Router, private headerService: HeaderService) {
    headerService.headerData = {
      title: 'Cadastro de Produtos',
      icon: 'storefront',
      routeUrl: '/products',
    };
  }

  ngOnInit(): void {}

  navigateToProductCreate(): void {
    this.router.navigate(['/products/create']);
  }
}
```

- No browser:

localhost:4200

The screenshot shows a web browser window with the URL `localhost:4200` in the address bar. The page has a header with the Cod3r logo and a navigation menu on the left containing 'Início' and 'Produtos'. The main content area displays a large 'Bem-vindo!' message and a subtitle 'Sistema para exemplificar a construção de um cadastro em Angular'. At the bottom right, there is a footer with the text 'Desenvolvido com ❤️ por Cod3r'.

`http://localhost:4200/products`

The screenshot shows a web browser window with the URL `localhost:4200/products` in the address bar. The page has a header with the Cod3r logo and a navigation menu on the left containing 'Início' and 'Produtos'. The main content area displays a table titled 'Cadastro de Produtos' with a 'Novo produto' button. The table lists five products with columns for Id, Name, Preço, and Ações (edit and delete icons). At the bottom right, there is a footer with the text 'Desenvolvido com ❤️ por Cod3r'.

Id	Name	Preço	Ações
1	Caneta BIC Preta - Escrita fina	R\$ 3,65	
2	Notebook Mac Pro	R\$ 12.000,89	
3	Sansung S20+	R\$ 5.000,89	
4	Geladeira	R\$ 1.890,44	
5	Liquidificador	R\$ 120,46	