

## ANGULAR 19 - INDICE

[Extensões do vscode usadas no curso](#)

[Adicionadas bibliotecas flex e uuid](#)

[Diferença entre LocalStorage e SessionStorage](#)

[Navegando entre páginas e passando parametros](#)

[Adicionando mascaras](#)

[Mostrando mensagens aos usuarios](#)

[APIs para diversas aplicacoes - brasilapi](#)

[Consumindo a api de listagem de uf](#)

[Instalando Tailwindcss](#)

[Standalone no Angular](#)

[Hierarquia do XAMA dos módulos](#)

[Navegando entre paginas e passando parametros](#)

[Formularios reativos do angular](#)

[Json server instalando](#)

[Renderizacao das estrelas na galeria](#)

[Pesquisa de lugares - versao correta do json server \(0.17.4\)](#)

[Aspas simples no option do select = opcao selecionada por padrao](#)

[Capturando titulo e subtítulo das paginas pelas rotas](#)

[Autenticacao e login social](#)

[Protegendo as rotas com guards](#)

[Como trabalhar com ambientes](#)

[Comandos docker](#)

[Inserindo máscaras](#)

## Angular 19 - 22/05/2025

1) Extensoes: Github theme, Material icon theme, Angular language service.

2) Tipos de for:

```
<div class="container">
  <table>
    <thead>
      <tr>
        <th>ID</th>
        <th>Nome</th>
        <th>Ativo</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>1</td>
        <td>João</td>
        <td><input type="checkbox" /></td>
      </tr>
      <tr>
        <td>2</td>
        <td>Maria</td>
        <td><input type="checkbox" /></td>
      </tr>
      <tr>
        <td>3</td>
        <td>Pedro</td>
        <td><input type="checkbox" /></td>
      </tr>
      <tr>
        <td>4</td>
        <td>Ana</td>
        <td><input type="checkbox" /></td>
      </tr>
      <tr>
        <td>5</td>
        <td>Carlos</td>
        <td><input type="checkbox" /></td>
      </tr>
    </tbody>
  </table>
```

3) Rodando testes com cobertura: ng test --code-coverage

4) Para criar um projeto em angular diferente da versão do angular cli que já está instalado, PRIMEIRO, tem que instalar o angular cli globalmente na versão desejada, para depois CRIAR o projeto na versao do angular que se quer, senão dá pau.

5) Quando adicionar **routerLink** em app.component.html, tem que adicionar **RouterModule.forRoot([])** nos imports do teste!

6) adicionou a biblioteca flex-layout: ng add @angular/flex-layout

7) adicionou a biblioteca uuid: ng add uuid@11.0.2

(uuid = universal unic id)

8) Ao acrescentar mat-form-field, tem que adicionar o “animations” no teste:

```
<div fxLayout="row" fxLayoutAlign="center">
  <div fxLayout="column" fxFlex="1000px">
    <form class="mt-60" #clientesFrm="ngForm">
      <mat-card>
        </mat-card-header>
        <mat-card-content class="mt-20">
          <div fxLayout="row" fxLayoutAlign="start" fxLayout-gap="16px">
            <mat-form-field>
              <mat-label>Nome: *</mat-label>
              <input type="text" placeholder="Ex: Frederico Ramos" matInput/>
            </mat-form-field>
          </div>
        </mat-card-content>
      </mat-card>
    </form>
  </div>
</div>
```

```
import { provideAnimations } from '@angular/platform-browser/animations';
```

```
providers: [
  provideAnimations()
]
```

9) Diferença entre LocalStorage e SessionStorage:

SessionStorage apaga tudo quando voce fecha o navegador.

LocalStorage mantem os dados salvos mesmo que voce feche o navegador.

## Aula 39 - Navegando entre páginas e passando parâmetros

Passo 1) Passar o id desejado com o router:

```
preparaEditar(id:string){  
  this.router.navigate(['/cadastro'], {queryParams: {"id": id}});  
}
```

Passo 2) Receber o dado passado com ActivatedRoute:

```
//  
export class CadastroComponent {  
  
  constructor(private clienteService: ClienteService,  
              private route: ActivatedRoute  
  ){}  
  
  ngOnInit(){  
    this.route.queryParamMap.subscribe((query:any) =>{  
      const parametros = query['params'];  
      console.log('parametros recebidos ==>', parametros)  
    })  
  }  
}
```

a

## Aula 44) adicionando mascaras

ng add ngx-mask

```
import { NgxMaskDirective, provideNgxMask } from 'ngx-mask';

@Component({
  selector: 'app-cadastro',
  imports: [FlexLayoutModule,
    MatCardModule,
    FormsModule,
    MatFormFieldModule,
    MatInputModule,
    MatIconModule,
    MatButtonModule,
    NgxMaskDirective
  ],
  providers: [
    provideNgxMask()
  ],
  templateUrl: './cadastro.component.html',
  styleUrls: ['./cadastro.component.scss']
})
export class CadastroComponent {
```

```
<div fxLayout="row" fxLayoutAlign="start" fxLayout-gap="16px">
  <div fxLayout = "column" flex="400px">
    <mat-form-field class="full-width">
      <mat-label>Data Nascimento: *</mat-label>
      <input
        type="text"
        matInput
        [(ngModel)]="cliente.dataNascimento"
        name="dataNascimento"
        mask="00/00/0000"/>
    </mat-form-field>
  </div>
</div>
```

a

## Aula 45 - Mostrando mensagens aos usuários

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

```
export class CadastroComponent {  
  
  atualizando: Boolean = false;  
  snack: MatSnackBar = inject(MatSnackBar)
```

```
salvar() {  
  if (!this.atualizando) {  
    this.clienteService.salvar(this.cliente);  
    this.mostrarMensagemAoUsuario("Usuario salvo com sucesso!");  
    this.cliente = Cliente.newCliente();  
  } else {  
    this.clienteService.atualizar(this.cliente);  
    this.mostrarMensagemAoUsuario("Usuario atualizado com sucesso!");  
  }  
}
```

## Aula 46 - APIs diversas para aplicações

<https://brasilapi.com.br/docs>

1) extensão no chrome para ver json: json viewer

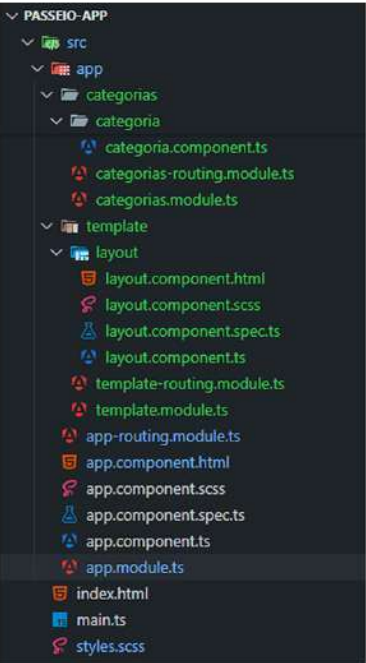
## Aula 48 - Consumindo a API de listagem de UF

1) em [app.config.ts](#):

```
import { provideHttpClient, withFetch } from '@angular/common/http';
```

```
providers: [  
  provideZoneChangeDetection({ eventCoalescing: true }),  
  provideRouter(routes),  
  provideAnimationsAsync(),  
  provideHttpClient(withFetch())  
]
```

2) ng g s brasilapi



### AppModule

```
src > app > app.modules > AppModule
1 @NgModule({
2   declarations: [
3     AppComponent
4   ],
5   imports: [
6     BrowserModule,
7     AppRoutingModule
8   ],
9   providers: [],
10  bootstrap: [AppComponent]
11 })
12 export class AppModule { }
```

import

AppRoutingModule

```
src > app > app-routing.modules > ...
1 import { NgModule } from '@angular/core';
2 import { RouterModule, Routes } from '@angular/router';
3
4 const routes: Routes = [
5   {path: 'paginas', loadChildren: () => import('./template/template.module')
6     .then(m => m.TemplateModule)}
7 ];
8
9 @NgModule({
10  imports: [RouterModule.forRoot(routes)],
11  exports: [RouterModule]
12 })
13 export class AppRoutingModule { }
```

```
src > app > template > template.modules > TemplateModule
1 @NgModule({
2   declarations: [
3     LayoutComponent
4   ],
5   imports: [
6     CommonModule,
7     TemplateRoutingModule,
8     CategoriasModule
9   ],
10 })
11 export class TemplateModule { }
```

```
src > app > template > layout > layout.components > LayoutComponent
1 @Component({
2   selector: 'app-layout',
3   standalone: false,
4   templateUrl: './layout.component.html',
5   styleUrls: ['./layout.component.scss']
6 })
7 export class LayoutComponent {
8 }
9
10
11 }
```

```
src > app > categorias > categorias.modules > ...
1 @NgModule({
2   declarations: [
3     CategoriaComponent
4   ],
5   imports: [
6     CommonModule,
7     CategoriasRoutingModule
8   ],
9 })
10 export class CategoriasModule { }
```

```
src > app > template > template-routing.modules > TemplateRoutingModule
1 const routes: Routes = [
2   {path: '', component: LayoutComponent,
3     children: [
4       {path: 'categorias', loadChildren: () => import('../categorias/categorias.module')
5         .then(m => m.CategoriasModule)}
6     ]
7   },
8 ];
9
10 @NgModule({
11  imports: [RouterModule.forChild(routes)],
12  exports: [RouterModule]
13 })
14 export class TemplateRoutingModule { }
```

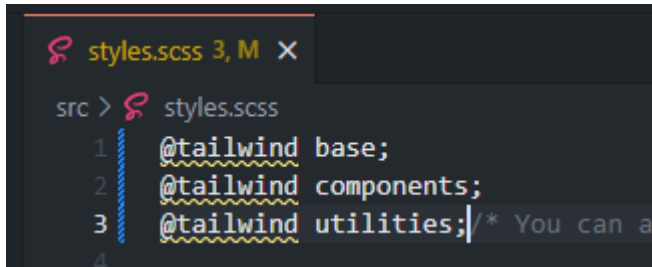


## Seção 4 - Angular avançado com Tailwindcss, módulos e APIs

1) npm install --save tailwindcss @3.4.17 postcss autoprefixer

2) npx tailwindcss init

3) em styles.css




```
1 @tailwind base;
2 @tailwind components;
3 @tailwind utilities; /* You can a
4
```

4) ng generate module template --routing

Esse template será o “esqueleto” da aplicação, ou seja, a página base.

5) ng generate component template/layout

6) Sobre o standalone



```
1 import { Component } from '@angular/core';
2
3 @Component({
4   selector: 'app-layout',
5   standalone: false,
6   templateUrl: './layout.component.html',
7   styleUrls: ['./layout.component.scss']
8 })
9 export class LayoutComponent {
10
11 }
12
```

standalone = não tem módulos.

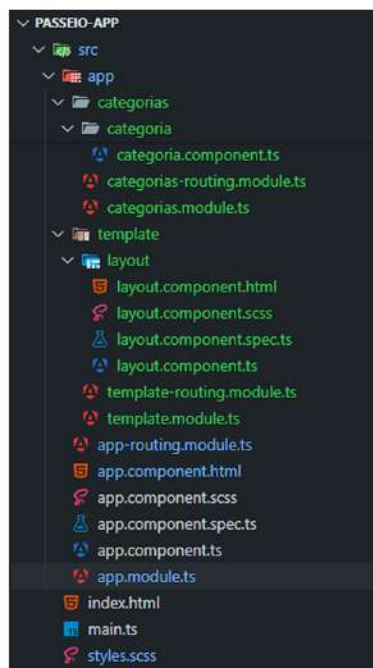
Do Angular 18 para trás, standalone será false por padrão.

Se vc quiser criar um projeto standalone nas versoes do angular de 18 pra trás, terá que colocar standalone = true.

7) ng g m categorias --routing

8) ng g c categorias/categoria

# HIERARQUIA DO "XAMA" DOS MÓDULOS



AppModule

```
app.module.ts M X
src > app > app.module.ts > AppModule
1 @NgModule({
2   declarations: [
3     AppComponent
4   ],
5   imports: [
6     BrowserModule,
7     AppRoutingModule
8   ],
9   providers: [],
10  bootstrap: [AppComponent]
11 })
12 export class AppModule { }
```

import

AppRoutingModule

```
app > app-routing.module.ts > ...
1 import { NgModule } from '@angular/core';
2 import { RouterModule, Routes } from '@angular/router';
3
4 const routes: Routes = [
5   {path: 'paginas', loadChildren: () => import('./template/template.module')
6     .then(m => m.TemplateModule)}
7 ];
8
9 @NgModule({
10  imports: [RouterModule.forRoot(routes)],
11  exports: [RouterModule]
12 })
13 export class AppRoutingModule { }
```

```
template.module.ts U X
src > app > template > template.module.ts > TemplateModule
1 @NgModule({
2   declarations: [
3     LayoutComponent
4   ],
5   imports: [
6     CommonModule,
7     TemplateRoutingModule,
8     CategoriasModule
9   ],
10 })
11 export class TemplateModule { }
```

```
layout.component.ts U X
src > app > template > layout > layout.component.ts > LayoutComponent
1 @Component({
2   selector: 'app-layout',
3   standalone: false,
4   templateUrl: './layout.component.html',
5   styleUrls: ['./layout.component.scss']
6 })
7 export class LayoutComponent {
8 }
9 }
```

template-routing.module

```
categorias.module.ts U X
src > app > categorias > categorias.module.ts > ...
1 @NgModule({
2   declarations: [
3     CategoriaComponent
4   ],
5   imports: [
6     CommonModule,
7     CategoriasRoutingModule
8   ],
9 })
10 export class CategoriasModule { }
```

```
template-routing.module.ts U X
src > app > template > template-routing.module.ts > TemplateRoutingModule
1 const routes: Routes = [
2   {path: '', component: LayoutComponent,
3     children: [
4       {path: 'categorias', loadChildren: () => import('../categorias/categorias.module')
5         .then(m => m.CategoriasModule)}
6     ]
7   },
8 ];
9
10 @NgModule({
11  imports: [RouterModule.forChild(routes)],
12  exports: [RouterModule]
13 })
14 export class TemplateRoutingModule { }
```

## Aula 39 - Navegando entre páginas e passando parâmetros

Passo 1) Passar o id desejado com o router:

```
preparaEditar(id:string){  
  this.router.navigate(['/cadastro'], {queryParams: {"id": id}});  
}
```

Passo 2) Receber o dado passado com ActivatedRoute:

```
export class CadastroComponent {  
  
  constructor(private clienteService: ClienteService,  
               private route: ActivatedRoute  
  ){}  
  
  ngOnInit(){  
    this.route.queryParamMap.subscribe((query:any) =>{  
      const parametros = query['params'];  
      console.log('parametros recebidos ==>', parametros)  
    })  
  }  
}
```

a

## Aula 44) adicionando mascaras

ng add ngx-mask

## Aula 58 - Formulários reativos do angular

1) Importe o REactiveFormsModule

```
categorias.module.ts M X
src > app > categorias > categorias.module.ts > ...
1 import { NgModule } from '@angular/core';
2
3 import { CategoriaComponent } from './categoria/categoria.component';
4 import { CategoriesRoutingModule } from './categorias-routing.module';
5 import { CategoriaComponent } from './categoria/categoria.component';
6 import { ReactiveFormsModule } from '@angular/forms';
7
8
9 @NgModule({
10   declarations: [
11     CategoriaComponent
12   ],
13   imports: [
14     CommonModule,
15     CategoriesRoutingModule,
16     ReactiveFormsModule
17   ]
18 })
19 export class CategoriesModule { }
20
```

2) Construa o formulario “camposForm” no arquivo ts:

```
import { Component } from '@angular/core';
import { FormGroup, FormControl, Validators } from '@angular/forms';

@Component({
  selector: 'app-categoria',
  standalone: false,
  templateUrl: './categoria.component.html',
  styleUrls: ['./categoria.component.scss']
})
export class CategoriaComponent {
  camposForm!: FormGroup;

  constructor() {
    this.camposForm = new FormGroup({
      nome: new FormControl('', Validators.required),
      descricao: new FormControl('', Validators.required)
    });
  }
}
```

a

3) aplique o formulario no html:

```
<form id="categoriaForm" [formGroup]="camposForm" (ngSubmit)="salvar()">
  <div class="mb-4">
    <label class="block text-gray-600 font-medium mb-2">Nome</label>
    <input type="text" id="nome" placeholder="Ex: Restaurantes, praias, shopping"
      class="w-full p-3 border border-gray-300 rounded shadow focus:ring focus:ring-bl
      required
      FormControlName="nome"/>
    <span class="text-red-500 text-sm" *ngIf="isInvalidField('nome')">Campo obrigatorio</span>
  </div>
```

4) instalando um servidor json para a API fake que criaremos  
npm install --save-dev json-server

\*\* --save-dev é usado apenas para ambiente de dev. Ele não salva como dependencias do projeto.

npm install --save-dev json-server

ANTES

DEPOIS

```
package.json X
package.json > ...
12   "dependencies": {
22     "rxjs": "~7.8.0",
23     "tailwindcss": "^3.4.17",
24     "tslib": "^2.3.0",
25     "zone.js": "~0.15.0"
26   },
27   "devDependencies": {
28     "@angular-devkit/build-angular": "^19.2.13",
29     "@angular/cli": "^19.2.13",
30     "@angular/compiler-cli": "^19.2.0",
31     "@types/jasmine": "~5.1.0",
32     "jasmine-core": "~5.6.0",
33     "karma": "~6.4.0",
34     "karma-chrome-launcher": "~3.2.0",
35     "karma-coverage": "~2.2.0",
36     "karma-jasmine": "~5.1.0",
37     "karma-jasmine-html-reporter": "~2.1.0",
38     "typescript": "~5.7.2"
39   }
40 }
```

```
package.json X
package.json > {} devDependencies
12   "dependencies": {
25     "zone.js": "~0.15.0"
26   },
27   "devDependencies": {
28     "@angular-devkit/build-angular": "^19.2.13",
29     "@angular/cli": "^19.2.13",
30     "@angular/compiler-cli": "^19.2.0",
31     "@types/jasmine": "~5.1.0",
32     "jasmine-core": "~5.6.0",
33     "json-server": "^1.0.0-beta.3",
34     "karma": "~6.4.0",
35     "karma-chrome-launcher": "~3.2.0",
36     "karma-coverage": "~2.2.0",
37     "karma-jasmine": "~5.1.0",
38     "karma-jasmine-html-reporter": "~2.1.0",
39     "typescript": "~5.7.2"
40   }
41 }
```


5) Como subir o json server:

- na raiz do projeto, crie uma pasta chamada “api”
- dentro da pasta api, crie um arquivo chamado “db.json”
- No arquivo package.json, na parte de scripts, adicione:

```
"server": "json-server ./src/api/db.json"
```

- suba o servidor com o comando: npm run server

```
Module '"@angular/common/http/testing"' has no exported member 'HttpClientTestingModule'.
```

 Isso acontece porque:

- `HttpClientTestingModule` e o `provideHttpClientTesting()` são recursos **novos** e ainda **experimentais**, introduzidos no **Angular 17+**, mas só **estáveis** a partir do **Angular 18** com o **"standalone test environment"** habilitado.

Se está dando esse erro, significa que:

- ✓ Ou sua versão do Angular ainda é **menor que 18**.
- ✓ Ou o **TypeScript config** e os **features experimentais** não estão habilitados no seu projeto (`environmentInjector`, etc).



## 76 - renderização das estrelas na galeria

1) Procure por entity character html

<https://www.toptal.com/designers/htmlarrows/symbols/>

## 77 - Pesquisa de lugares

**ATENÇÃO!**

o JSON SERVER só funcionou a busca corretamente na versão abaixo:

```
package.json > {} devDependencies
13   "dependencies": {
23     "rxjs": "~7.8.0",
24     "tailwindcss": "^3.4.17",
25     "tslib": "^2.3.0",
26     "zone.js": "~0.15.0"
27   },
28   "devDependencies": {
29     "@angular-devkit/build-angular": "^19.2.13",
30     "@angular/cli": "^19.2.13",
31     "@angular/compiler-cli": "^19.2.0",
32     "@types/jasmine": "~5.1.0",
33     "jasmine-core": "~5.6.0",
34     "json-server": "^0.17.4",
35     "karma": "~6.4.0",
36     "karma-chrome-launcher": "~3.2.0",
37     "karma-coverage": "~2.2.0",
38     "karma-jasmine": "~5.1.0",
39     "karma-jasmine-html-reporter": "~2.1.0",
40     "typescript": "~5.7.2"
41   }
42 }
43
```

No Angular, se voce não especificar nada no value do option, ele passa como parametro o que está dentro do option. No caso aqui, é o "Todas as categorias".

```
<select [(ngModel)]="categoriaFiltro"
  name="categoriaFiltro"
  class="w-full border border-gray-300 p-2 rounded
... <option [value]="">>Todas as Categorias</option>
  <option *ngFor="let cat of categoriasFiltro" [value]=
</select>
```

a

## 80 - Capturando título e subtítulo das páginas pelas rotas

```

aleria.component.html M  layout.component.html M  template-routing.module.ts M x
app > template > template-routing.module.ts > routes > children > data > subtítulo

const routes: Routes = [
  {path: '', component: LayoutComponent,
    children: [
      {path: 'categorias',
        loadChildren: () => import('../categorias/categorias.module')
          .then(m => m.CategoriasModule),
        pathMatch: 'full',
        data: {titulo: 'Categorias', subtítulo: 'Cadastro de novas categorias'}},
      {
        path: 'lugares',
        loadChildren: () => import('../lugares/lugares.module')
          .then(l => l.LugaresModule),
        pathMatch: 'full',
        data: {titulo: 'Eventos', subtítulo: 'Cadastro de novos eventos'}},
      {
        path: 'galeria',
        loadChildren: () => import('../galeria/galeria.module')
          .then(g => g.GaleriaModule),
        pathMatch: 'full',
        data: {titulo: 'Listagem dos eventos', subtítulo: 'Não que eu vá, mas hoje é onde?'}},
    ]
  },
]
```

```

export class LayoutComponent implements OnInit {

  props: LayoutProps = {titulo: '', subtítulo: ''};

  constructor(private router: Router,
    private activatedRoute: ActivatedRoute){}

  ngOnInit(): void {
    this.router.events
      .pipe(
        filter(() => this.activatedRoute.firstChild !== null),
        map(() => this.obterTituloESubtitulo())
      ).subscribe((props: LayoutProps) => this.props = props)
  }

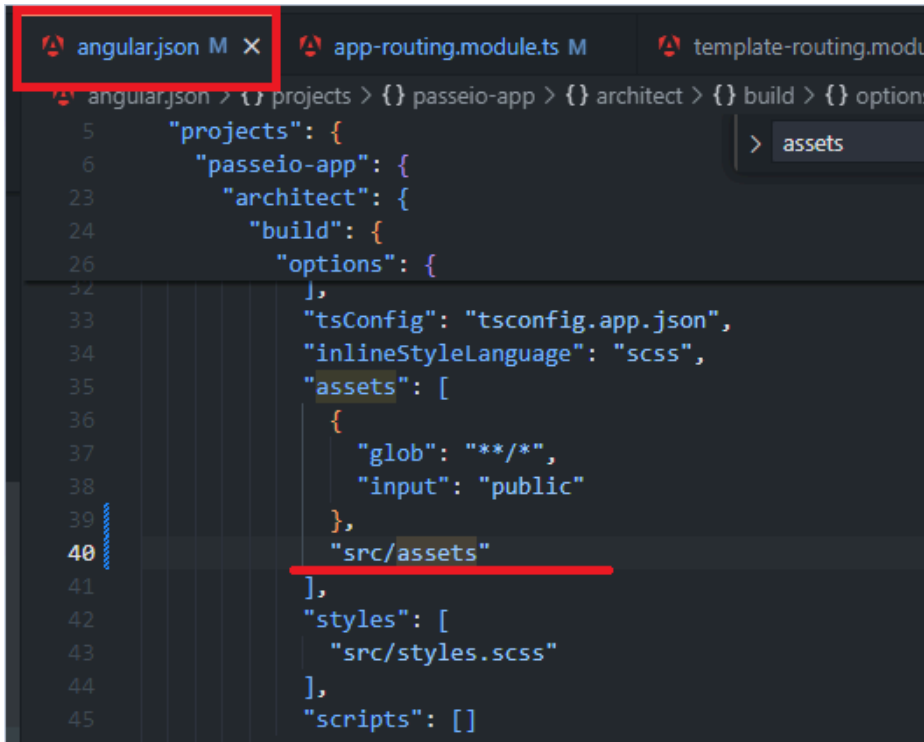
  obterTituloESubtitulo(): LayoutProps {
    let rotaFilha = this.activatedRoute.firstChild;
    while(rotaFilha?.firstChild){
      rotaFilha = rotaFilha.firstChild;
    }
    return rotaFilha?.snapshot.data as LayoutProps;
  }
}
```



## Seção 5 - Autenticação e login social

1) ng g c landingpage

2) Alterando arquivo angular.json para ler imagens a partir do diretório src:



```
angular.json M x app-routing.module.ts M template-routing.modu
angular.json > {} projects > {} passeio-app > {} architect > {} build > {} options
5   "projects": {
6     "passeio-app": {
23     "architect": {
24       "build": {
26         "options": {
32           },
33         "tsConfig": "tsconfig.app.json",
34         "inlineStyleLanguage": "scss",
35         "assets": [
36           {
37             "glob": "**/*",
38             "input": "public"
39           },
40           "src/assets"
41         ],
42         "styles": [
43           "src/styles.scss"
44         ],
45         "scripts": []
```

3) Configurar a aplicação no google cloud

4) instalar a dependencia de oauth2 na aplicação:

```
npm install --save angular-oauth2-oidc@19.0.0
```

## 86 - Protegendo as rotas com guards

1) ng generate guard auth

2) Desenvolveu a [auth.guard.ts](#)

3) Como definir quais rotas deverão ser guardadas pelo guardião de rotas acima? vá em [app.routing.module.ts](#) e configure o guarda na rota da pagina que voce quer:

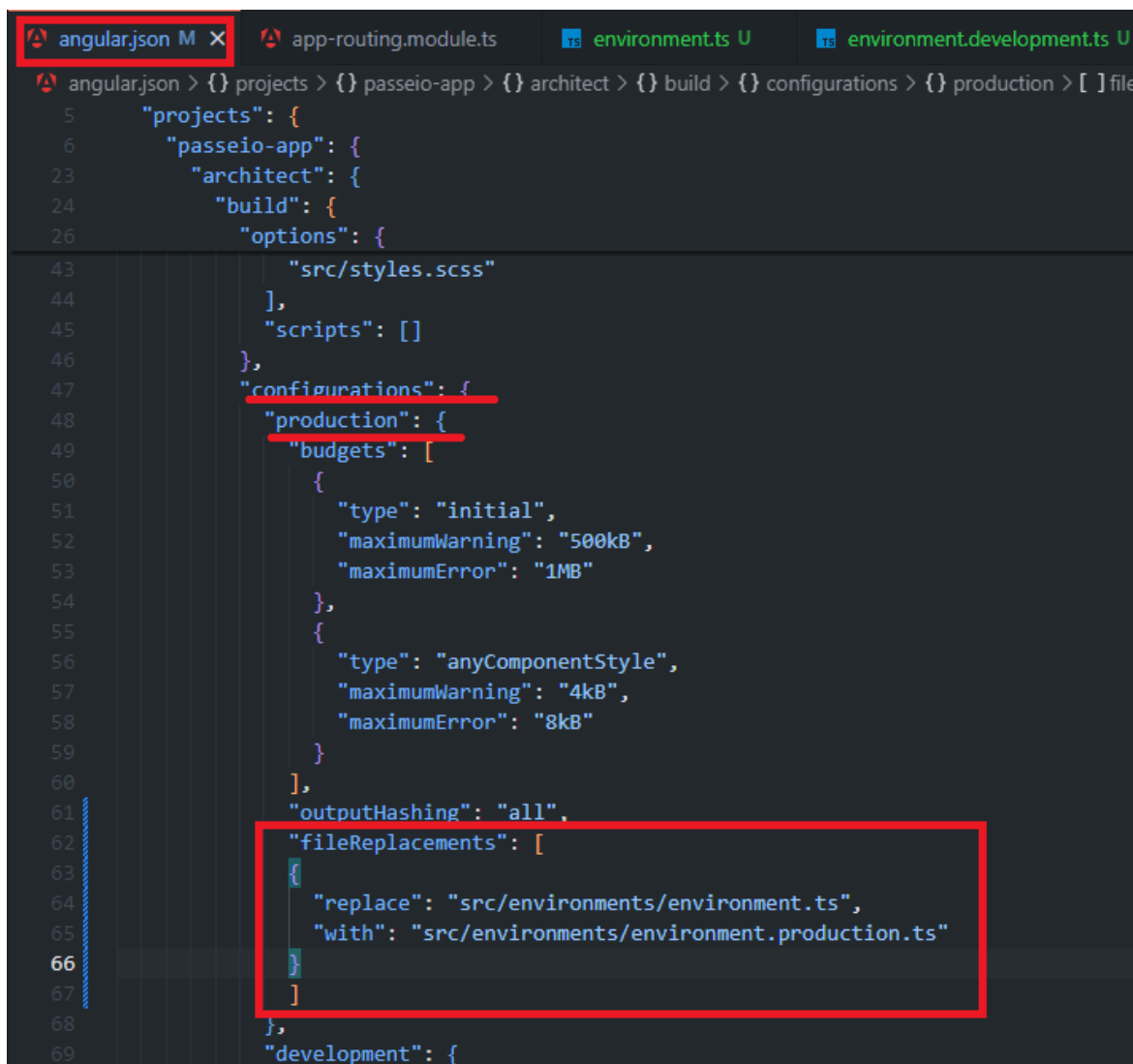
```
app-routing.module.ts M X  template-routing.module.ts  landingpage.component.ts  landingp
src > app > app-routing.module.ts > ...
1  import { NgModule } from '@angular/core';
2  import { RouterModule, Routes } from '@angular/router';
3  import { LandingpageComponent } from '../landingpage/landingpage.component';
4  import { authGuard } from '../auth.guard';
5
6  const routes: Routes = [
7    {
8      path:'', component: LandingpageComponent
9    },
10   {path:'paginas', loadChildren: () => import('../template/template.module')
11     .then(m => m.TemplateModule),
12     canActivate: [authGuard]}
13 ];
14
15 @NgModule({
16   imports: [RouterModule.forRoot(routes)],
17   exports: [RouterModule]
18 })
19 export class AppRoutingModule { }
20 |
```

## 87 - Como trabalhar com ambientes

- 1) ng generate environments
- 2) Criou uma variável apiURL em environment.
- 3) Criou outra api do json server: db.production.json
- 4) Criou o arquivo [environment.production.ts](#)
- 5) Alterou o arquivo package.json adicionando em scripts:

```
"server:production": "json-server ./src/api/db.production.json --port:4000"
```

- 6) alterou angular.json:



```
angular.json > {} projects > {} passeio-app > {} architect > {} build > {} configurations > {} production > [ ] file
5  "projects": {
6    "passeio-app": {
23     "architect": {
24       "build": {
26         "options": {
43           "src/styles.scss"
44         },
45         "scripts": []
46       },
47       "configurations": {
48         "production": {
49           "budgets": [
50             {
51               "type": "initial",
52               "maximumWarning": "500kB",
53               "maximumError": "1MB"
54             },
55             {
56               "type": "anyComponentStyle",
57               "maximumWarning": "4kB",
58               "maximumError": "8kB"
59             }
60           ],
61           "outputHashing": "all",
62           "fileReplacements": [
63             {
64               "replace": "src/environments/environment.ts",
65               "with": "src/environments/environment.production.ts"
66             }
67           ]
68         },
69         "development": {
```

- 7) Para testar, suba a aplicação Angular com:  
ng serve --configuration=production

## 8) Instalou o docker desktop.

### Comandos docker:

```
#comando para criar o container da aplicacao
#docker build --tag cursoangular-app .

#comando para subir o container da aplicacao
#docker run -p 4200:80 --name cursoangular-app-container -d cursoangular-app

#comando para verificar logs do container da aplicacao
#docker logs cursoangular-app-container

#comando para remover o container
#docker rm -f meu_container
```