

Last chance! 4 days left! [Get 20% off membership now](#)

Open in app ↗

Medium

🔍 Search

✍ Write



# Front End com Angular — IFPI Corrente.



Jhonatas G. Ribeiro

6 min read · 2 hours ago



## O que é Angular?

O Angular é um Framework desenvolvido pelo Google para criação de aplicações para web. É construído baseado em TypeScript e utiliza ‘Orientação a Objetos’ e ‘Orientação a Componentes’ para construção de interfaces web.

## Por que o Angular é tão utilizado no mundo corporativo e grandes empresas?

O Angular é comumente utilizado devido à sua robustez e conjunto de recursos abrangentes. A sua estrutura baseada em módulos permite a reutilização de componentes e manutenção de forma fácil. O uso do TypeScript como linguagem padrão fornece segurança devido à sua tipagem estática. O Angular também fornece aos desenvolvedores a integração com

diversas bibliotecas de terceiros e também a sua biblioteca própria conhecida como Angular Material .

## Angular CLI

O Angular possui uma Interface de Linha de Comando (CLI) que facilita o desenvolvimento de um novo projeto, simplificando a criação de projeto, componentes e arquivos. Com a CLI não é necessário que o desenvolvedor se preocupe com a estrutura de pastas e configuração de arquivos, já que apenas com um comando, todos os pacotes são criados e configurados de acordo com a estrutura padrão, deixando tudo pronto para o desenvolvedor colocar a mão na massa.

## Componentes

O angular é um framework baseado em componentes, por isso, é possível dividir a aplicação em pequenas partes de códigos reutilizadas em todo o projeto. Um componente Angular é composto por 3 arquivos principais:

- 1 arquivo HTML, para estruturação.
- 1 arquivo de estilo (CSS ou SCSS).
- 1 arquivo TypeScript, que controla e configura o componente.

## Criação de uma lista de tarefas utilizando o Angular

Para começar o desenvolvimento, primeiro é necessário que você tenha instalado em sua máquina o Node , NPM e o Angular CLI.

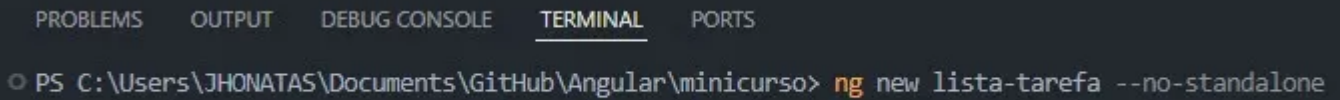
### Comando para instalar o NPM:

```
install npm -g npm
```

### Comando para instalar o Angular CLI v17:

```
npm install -g @angular/cli@17
```

## Comando para criar aplicação Angular:



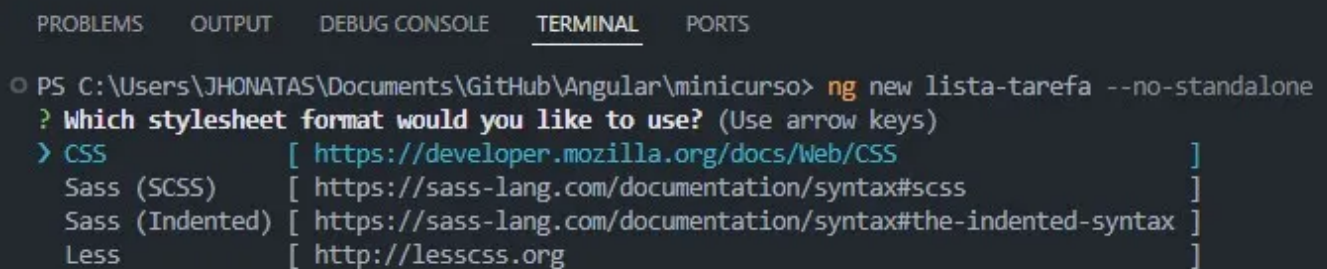
```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\JHONATAS\Documents\GitHub\Angular\minicurso> ng new lista-tarefa --no-standalone
```

```
ng new <nome-aplicacao> --no-standalone
```

o comando `-no-standalone` indica ao angular que ele deve ser criado na estrutura modular.

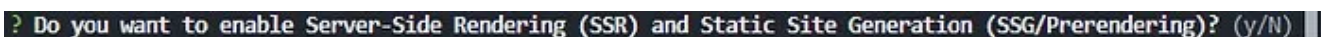
O angular irá realizar algumas perguntas na criação do projeto:

- 1º O tipo de arquivo de estilo. Usaremos CSS.



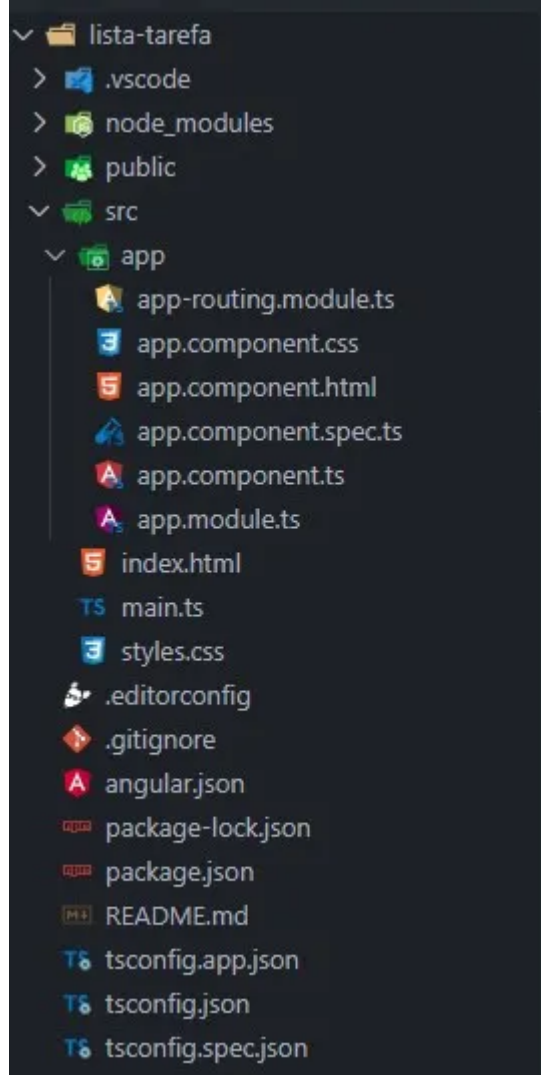
```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\JHONATAS\Documents\GitHub\Angular\minicurso> ng new lista-tarefa --no-standalone
? Which stylesheet format would you like to use? (Use arrow keys)
> CSS [ https://developer.mozilla.org/docs/Web/CSS ]
  Sass (SCSS) [ https://sass-lang.com/documentation/syntax#scss ]
  Sass (Indented) [ https://sass-lang.com/documentation/syntax#the-indented-syntax ]
  Less [ http://lesscss.org ]
```

- 2º Se iremos querer usar rotas. Não Será Necessário.



```
? Do you want to enable Server-Side Rendering (SSR) and Static Site Generation (SSG/Prerendering)? (y/N)
```

Após esses passos teremos um projeto Angular criado com a seguinte estrutura:



Com o projeto criado, precisamos construir e servir nossa aplicação. Para isso utilizaremos o comando;

```
PS C:\Users\JHONATAS\Documents\GitHub\Angular\minicurso> cd .\lista-tarefa\  
PS C:\Users\JHONATAS\Documents\GitHub\Angular\minicurso\lista-tarefa> ng serve
```

ng serve

Se tudo ocorrer bem, no terminal aparecerá no terminal a porta do servidor que está rodando.

```
Browser bundles
Initial chunk files | Names | Raw size
polyfills.js | polyfills | 90.23 kB |
main.js | main | 23.71 kB |
styles.css | styles | 95 bytes |

| Initial total | 114.04 kB

Server bundles
Initial chunk files | Names | Raw size
polyfills.server.mjs | polyfills.server | 572.95 kB |
main.server.mjs | main.server | 24.21 kB |
render-utils.server.mjs | render-utils.server | 472 bytes |

Application bundle generation complete. [6.165 seconds]

Watch mode enabled. Watching for file changes...
NOTE: Raw file sizes do not reflect development server per-request transformations.
  → Local: http://localhost:4200/
  → press h + enter to show help
```

Pronto, agora temos uma aplicação angular criada e sendo executada em servidor local, mas não acabou por aqui. Agora iremos de fato colocar a mão na massa.

## Criando componente da nossa lista de tarefas

Para criarmos um componente angular utilizaremos o seguinte comando

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\JHONATAS\Documents\GitHub\Angular\minicurso\lista-tarefa> ng generate component corpo-lista
```

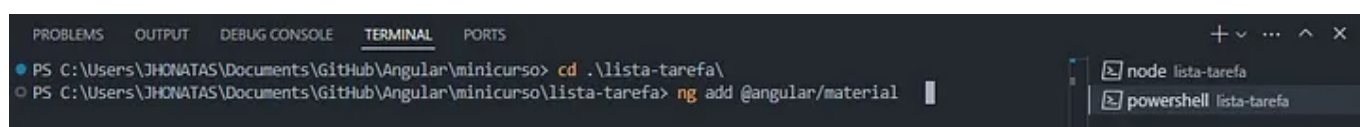
```
ng generate component <nome-component>
```

Se tudo ocorrer bem, o componente será criado juntamente com seus respectivos arquivos base.

```
PS C:\Users\JHONATAS\Documents\GitHub\Angular\minicurso\lista-tarefa>
CREATE src/app/corpo-lista/corpo-lista.component.html (27 bytes)
CREATE src/app/corpo-lista/corpo-lista.component.spec.ts (649 bytes)
CREATE src/app/corpo-lista/corpo-lista.component.ts (228 bytes)
CREATE src/app/corpo-lista/corpo-lista.component.css (0 bytes)
UPDATE src/app/app.module.ts (685 bytes)
```



No desenvolvimento iremos utilizar a biblioteca do Angular Material, para isso precisaremos adicionar essa biblioteca à nossa aplicação utilizando o seguinte comando:



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

Também iremos utilizar um serviço angular como nosso ‘Banco de dados’, para isso precisamos criar esse arquivo com o seguinte comando CLI

```
ng generate service <nome-service>
```

Com isso teremos esse arquivo para armazenar e manipular nossos dados. Abaixo deixarei o modelo de cada arquivo bem como seus respectivos imports necessários no app.module

## HTML

```
<div class="container">
  <div class="header">
    <form class="example-form">
```

```

    <mat-form-field class="example-full-width">
      <mat-label>Tarefa:</mat-label>
      <input name="texto" matInput placeholder="Atividade de Algoritmos" [(ngModel)]="texto">
    </mat-form-field>
    <button mat-flat-button (click)="adicionar()">Adicionar Tarefa</button>
  </form>
</div>
<hr>
<div class="body">
  <h6>Tarefas não concluídas</h6>
  @for(tarefa of listaTarefas;track tarefa){
  <div class="lista">
    <span>{{tarefa.texto}}</span>
    <span>
      <button mat-fab (click)="concluir(tarefa.id)">
        <mat-icon>check</mat-icon>
      </button>
      <button mat-fab (click)="excluir(tarefa.id)">
        <mat-icon>delete</mat-icon>
      </button>
    </span>
  </div>
  <hr>
  }@empty {
  <span>Ainda não há tarefas</span>
  }
  <hr>
  <h6>Tarefas Feitas</h6>
  @for(tarefa of listaTarefasFeitas;track tarefa){
  <div class="lista">
    <span>{{tarefa.texto}}</span>
    <span>
      <button mat-fab (click)="excluir(tarefa.id)">
        <mat-icon>delete</mat-icon>
      </button>
    </span>
  </div>
  <hr>
  }@empty {
  <span>Ainda não há tarefas feitas</span>
  }
</div>
</div>

```

## CSS

```

*{
  margin : auto;
}
.container {

```

```

background-color : rgba ( 0 , 0 , 0 , 0.329 );
max-width : 50% ;
max-height : max-content;
margin-top : 3% ;
border-radius : 2% ;
}
.example-form , .header {
width : 85% ;
display : flex;
}
.example-full-width , .body {
width : 100% ;
margin : 2% ;
}
.lista {
width : 100% ;
display : flex;
justify-content : space-between;
margin : 1% ;
}
h6 , span {
text-align : center;
}
button {
margin : 1px
}

```

## TypeScript

```

export class CorpoListaComponent {
  texto!: any
  listaTarefas!: any
  listaTarefasFeitas!: any
  constructor ( private service: ServiceService ) {
    this . listaTarefas = service . getTarefas ()
    this . listaTarefasFeitas = service . getTarefasFeitas ()
  }
  excluir ( indice: number ) {
    this . service . excluirTarefa (indice)
    this . listaTarefas = this . service . getTarefas ()
    this . listaTarefasFeitas = this . service . getTarefasFeitas ()
  }
  adicionar () {
    this . service . adicionarTarefa ( this . texto )
    this . texto = ''
    this . listaTarefas = this . service . getTarefas ()
    this . listaTarefasFeitas = this . service . getTarefasFeitas ()
  }
  concluir ( indice: number ) {
    this . service . concluirTarefa (indice)
    this . listaTarefas = this . service . getTarefas ()
  }
}

```



```
    this . listaTarefasFeitas = this . service . getTarefasFeitas ()  
  }  
}
```

## Service

```
exportar classe ServiceService {  
  construtor () { }  
  private Tarefalista : any = []; id  
  privado = 0 ;  
  getTarefas () {  
    return this.listaTarefa.filter((item: any) => item.feita == false);  
  }  
  getTarefasFeitas() {  
    return this . listaTarefa . filter ( ( item: any ) => item. feita == true );  
  }  
  adicionarTarefa ( texto: string ) {  
    this . listaTarefa . push ({  
      id : this . id ,  
      texto : texto,  
      feita : false  
    });  
    this . id ++;  
  }  
  excluirTarefa ( id: number ) {  
    this . listaTarefa = this . listaTarefa . filter ( ( item: any ) => item. id !=  
  }  
  concluirTarefa ( id: number ) {  
    this . listaTarefa . forEach ( ( item: any ) => {  
      if (item. id === id) {  
        item. feita = true;  
      }  
    });  
  }  
}
```

## app.module

```
import { NgModule } from '@angular/core';  
import { BrowserModule, provideClientHydration } from '@angular/platform-browser';  
import { AppRoutingModule } from './app-routing.module';  
import { AppComponent } from './app.component';  
import { provideAnimationsAsync } from '@angular/platform-browser/animations/asy  
import { CorpoListaComponent } from './corpo-lista/corpo-lista.component';  
import { FormsModule } from '@angular/forms'
```

```
import { MatInputModule } from '@angular/material/input';
import { MatFormFieldModule } from '@angular/material/form-field';
import { MatIconModule } from '@angular/material/icon';
import { MatDividerModule } from '@angular/material/divider';
import { MatButtonModule } from '@angular/material/button';
import { MatListModule } from '@angular/material/list';
import { MatTableModule } from '@angular/material/table';

@NgModule({
  declarations: [
    AppComponent,
    CorpoListaComponent,
  ],
  imports: [
    BrowserModule,
    AppRoutingModule,
    FormsModule,
    MatInputModule,
    MatFormFieldModule,
    MatIconModule,
    MatButtonModule,
    MatDividerModule,
    MatListModule,
    MatTableModule,
  ],
  providers: [
    provideClientHydration(),
    provideAnimationsAsync()
  ],
  bootstrap: [AppComponent]
})
export class AppModule { }
```



**Written by Jhonatas G. Ribeiro**

0 Followers

Edit profile