



Introdução Angular

Framework de desenvolvimento Frontend

Plataforma de desenvolvimento com estrutura pré-definida

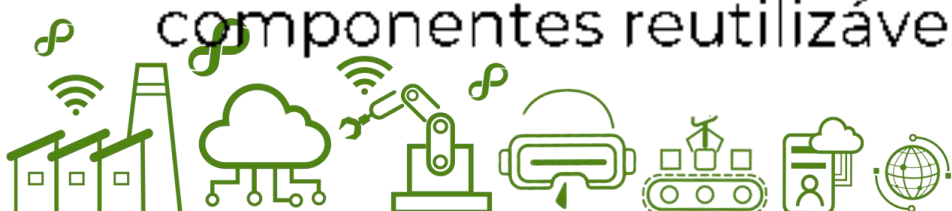
Fornece bibliotecas, ferramentas e boas práticas

Facilita a criação de aplicações de forma eficiente e padronizada

Framework web open-source mantido pelo Google

Ideal para criar aplicações web dinâmicas e SPA (Single Page Applications)

Oferece ferramentas robustas: rotas, injeção de dependência, componentes reutilizáveis e manipulação de dados.



O que é SPA (Single Page Application)

- Aplicação web que carrega uma única página HTML.
- Atualiza dinamicamente o conteúdo sem recarregar a página inteira.
- só mudam o conteúdo necessário. Isso deixa o site mais rápido e com experiência parecida com um aplicativo.
- Proporciona experiência mais rápida e fluida ao usuário
- Utiliza JavaScript e rotas no frontend para navegar entre “páginas”

Exemplo: Gmail, Google Maps, Whatsapp Web



Introdução Angular, Características:

Estrutura Pré-definida: Define como os componentes do software interagem

Reusabilidade: Permite reaproveitar código, economizando tempo e esforço

Inversão de Controle (IoC): O framework controla o fluxo, chamando o código do desenvolvedor

Configuração por Convenção: Reduz configurações explícitas com padrões pré-estabelecidos

Modularidade: Permite adicionar ou substituir componentes de forma flexível.



Introdução Angular

Vantagens

Qualidade: Incentiva boas práticas e melhora o código

Comunidade e Suporte: Acesso fácil a tutoriais, fóruns e ajuda especializada

Desvantagens

Curva de Aprendizado: Pode ser difícil para iniciantes

Sobrecarga de Código: Pode tornar o software mais pesado e complexo

Dependência: Difícil migrar ou adaptar se o framework for descontinuado



Evolução do Angular

AngularJS (1.x) – Lançado em 2010

Introduziu o Two-Way Data Binding

Facilitou o desenvolvimento de apps web dinâmicos

Angular 2 – Lançado em 2016

Reescrita completa com arquitetura baseada em componentes

Melhor desempenho e suporte para dispositivos móveis

Incompatível com AngularJS



Evolução do Angular

Angular 4, 5, 6...

Ciclo de lançamento semestral

Melhorias incrementais, novas funcionalidades e correções de bugs

Angular Atual

Evolução contínua

Maior desempenho, melhor suporte ao ECMAScript e integração com tecnologias modernas



@fpftech.educacional

Antes... Vamos precisar de NodeJs e Npm

Node.js

Plataforma de execução JavaScript no lado do servidor.

Construído sobre o motor V8 do Google Chrome.

Permite rodar JavaScript fora do navegador.

Usado para criar servidores, CLIs e ferramentas de build.

Características principais:

Assíncrono e orientado a eventos.

Alta performance para I/O

Grande ecossistema de bibliotecas via npm



Antes... Vamos precisar de NodeJs e Npm

npm (Node Package Manager)

Gerenciador de pacotes para Node.js

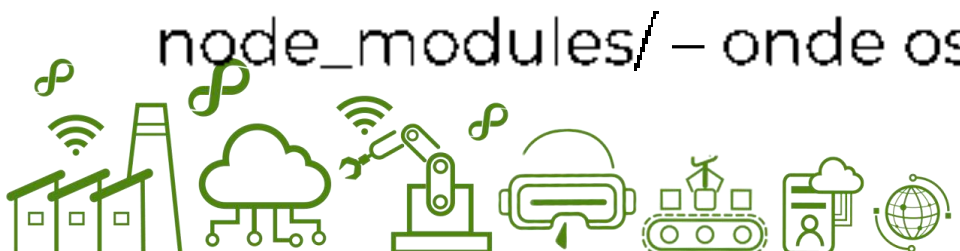
Permite instalar, atualizar e gerenciar bibliotecas JavaScript

Mantém as dependências listadas em package.json

Principais arquivos:

package.json – lista de dependências e scripts

node_modules/ – onde os pacotes são instalados



NodeJs e Npm

Verificar se Node.js e npm estão instalados:

```
node -v  
npm -v
```

Se não estão instalados, faça download em:

<https://docs.npmjs.com/downloading-and-installing-node-js-and-npm>

Alguns códigos úteis de npm:

```
npm install #instala todas as dependências  
npm install <pacote> # instala pacote específico  
npm start # executa script definido no package.json
```



Npm Angular CLI

O Angular CLI (ng) é um pacote Node.js instalado via npm:

```
ng version #verifica a versão do Angular CLI, se está instalado  
npm install -g @angular/cli #para instalar o Angular CLI
```

O Angular depende de várias ferramentas e bibliotecas que são gerenciadas pelo npm (TypeScript, Webpack, RxJS, etc.).

- npm gerencia os pacotes do projeto
- scripts como **ng serve**, **ng build** e **ng test** usam o ambiente Node.js
- Sem o Node.js e o npm, **não é possível instalar**, configurar ou rodar



Angular usa TypeScript

TypeScript é uma evolução do JavaScript, com tipagem, ou seja, você declara os tipos das variáveis, ajudando a evitar erros e facilitando a manutenção.

Estrutura bem definida: separa o código em **componentes** (pedaços da tela), **serviços** (lógica), e **módulos** (organização).

Reatividade: o conteúdo se **atualiza automaticamente** quando os dados mudam, sem precisar mexer no HTML manualmente.

Angular CLI (linha de comando para gerar código e rodar o projeto),
RxJS (programação reativa) e **HttpClient** (para comunicação com APIs).



Criando primeiro projeto

Todos os códigos que começam com “ng” são do Angular CLI:

No terminal:

```
ng new meu-projeto #cria novo projeto angular chamado “meu-projeto”  
cd meu-projeto #entra na pasta do projeto criado  
ng serve -o #roda o projeto na porta
```

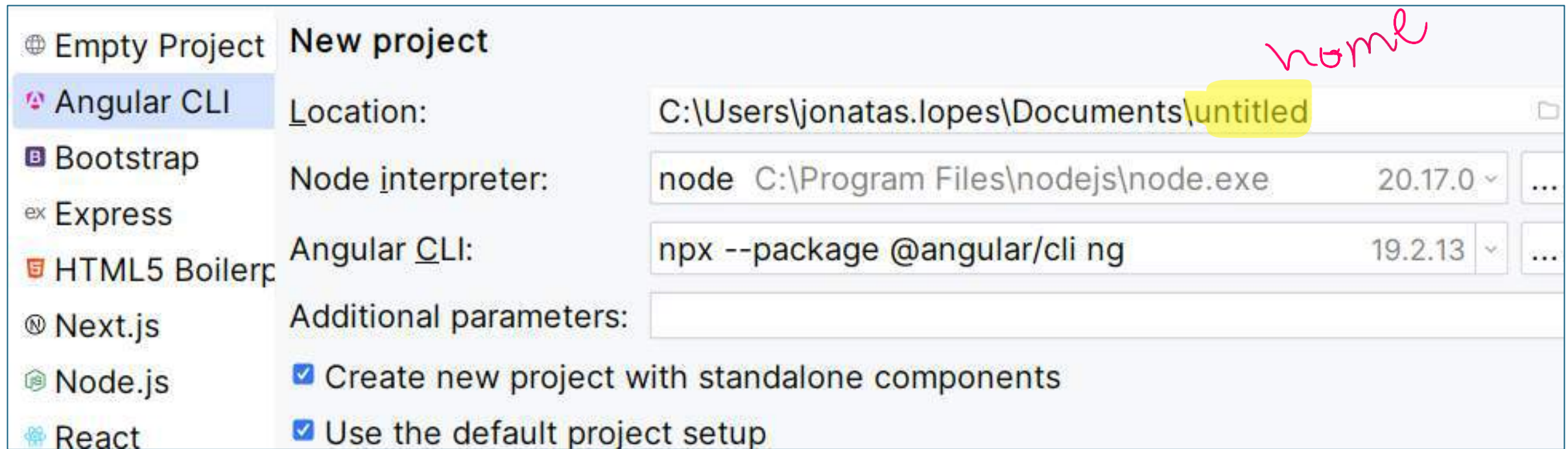
Esse comando abre o navegador automaticamente em
<http://localhost:4200>, mostrando o projeto rodando.



Criando primeiro projeto

No Webstorm:

Novo projeto > Angular >



Empty Project	New project	
Angular CLI	Location:	C:\Users\jonatas.lopes\Documents\untitled
Bootstrap	Node interpreter:	node C:\Program Files\nodejs\node.exe 20.17.0
Express	Angular CLI:	npx --package @angular/cli ng 19.2.13
HTML5 Boilerplate	Additional parameters:	
Next.js	<input checked="" type="checkbox"/> Create new project with standalone components	
Node.js	<input checked="" type="checkbox"/> Use the default project setup	
React		



Perguntas do primeiro projeto

Would you like to share pseudonymous usage data about this project with the Angular Team at Google under Google's Privacy Policy at <https://policies.google.com/privacy>. For more details and how to change this setting, see <https://angular.dev/cli/analytics>.

(y/N) █

Opcionais:

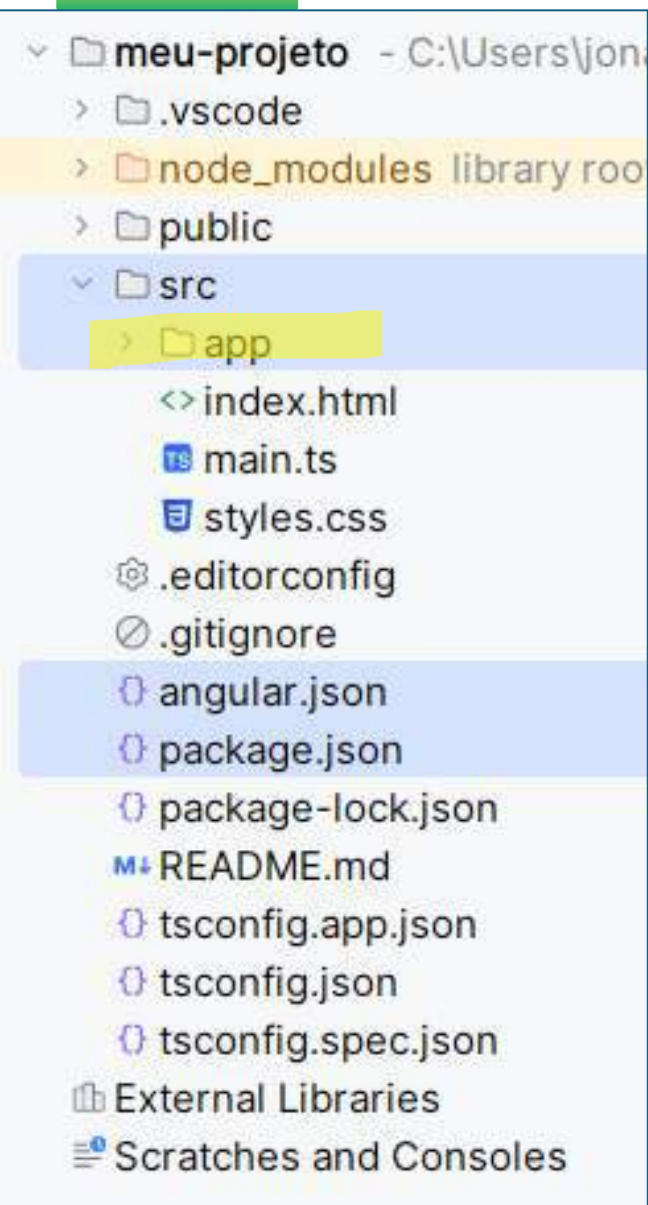
- Qual o tipo de estilo? CSS, SCSS (melhor) ... ?

Dúvidas? <https://angular.dev/tutorials/first-app>



@fpftech.educacional

Estrutura básica do projeto Angular



`src/app/` → aqui ficam os componentes e módulos (pedaços do projeto)

`src/assets/` → imagens e arquivos estáticos

`src/environments/` → configurações para diferentes ambientes (dev, produção)

angular.json → configuração geral do projeto

package.json → lista das dependências do projeto



Componentes no Angular

O que é um componente?

Um componente é a unidade básica da interface no Angular.

É como um bloco que tem:

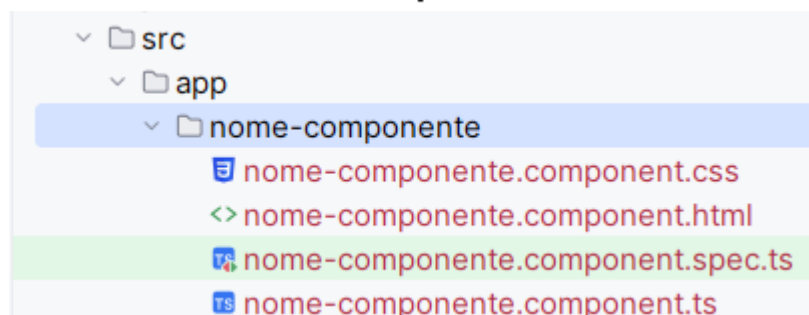
- Um **template HTML**: o que aparece na tela.
- Um arquivo **CSS/SCSS**: estilo visual.
- Um arquivo **TypeScript**: onde fica a lógica e os dados.



Gerando um componente novo

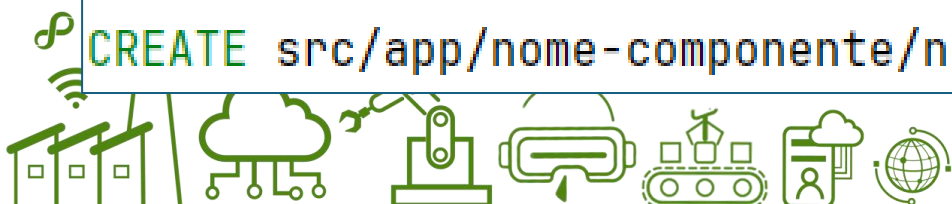
```
cuments\meu-projeto> ng generate component nome-componente --standalone
```

Isso cria uma pasta com os arquivos do componente.



Veja os códigos que criam:

```
PS C:\Users\jonatas.lopes\Documents\meu-projeto> ng generate component nome-componente
CREATE src/app/nome-componente/nome-componente.component.html (31 bytes)
CREATE src/app/nome-componente/nome-componente.component.spec.ts (672 bytes)
CREATE src/app/nome-componente/nome-componente.component.ts (260 bytes)
CREATE src/app/nome-componente/nome-componente.component.css (0 bytes)
```



Editando o componente novo

No HTML:

```
1 <h2>{{ nome }}</h2>
2 <p>{{ profissao }}</p>
3 <ul>
4   @for(hobby of hobbies; track hobby){
5     <li>{{ hobby }}</li>
6   }
7 </ul>
8
```

No JS:

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

1+ usages
@Component({
  selector: 'app-nome-componente',
  imports: [],
  templateUrl: './nome-componente.component.html',
  standalone: true,
  styleUrls: ['./nome-componente.component.css']
})
export class NomeComponenteComponent {
  nome :string  = 'Jonatas S Lopes';
  profissao :string  = 'Desenvolvedor Web Dev';
  hobbies :string[] = ['Leitura', 'Estudo', 'Tecnologia', 'Jogar'];
}
```

Obs. Html foi todo adicionado.

JS = somente adicionei a parte azul.



Editando o componente APP pra aparecer o nosso

No HTML:

Obs. Html foi todo deletado.

Depois adicionei somente 1 linha.

```
<> app.component.html x
1 <app-nome-componente></app-nome-componente>
2 |
```

No JS:

```
ts app.component.ts x
1 import { Component } from '@angular/core';
2 import { RouterOutlet } from '@angular/router';
3 import { NomeComponenteComponent } from './nome-componente';
4
5 1+ usages jonatas.lopes
6 @Component({
7   selector: 'app-root',
8   imports: [RouterOutlet, NomeComponenteComponent],
9   templateUrl: './app.component.html',
10  standalone: true,
11  styleUrls: ['./app.component.css']
12 })
13 export class AppComponent {
14   title: string = 'meu-projeto';
15 }
```



1. **Identify the main components of the system.**

1. Verificar se Node.js, Npm e Angular CLI estão instalados
2. Criar e rodar o primeiro projeto Angular
3. Gerar um componente novo, chamado **perfil**

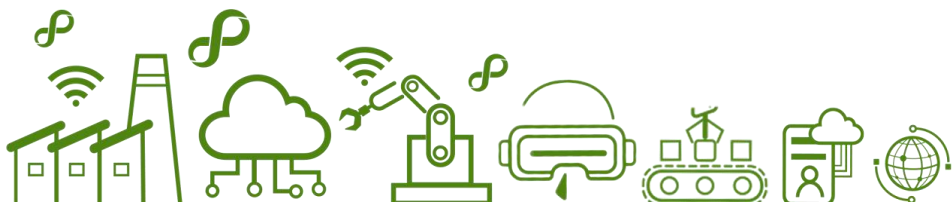
4. Preencher seu perfil com os dados (no arquivo js).

5. Criar o template HTML do seu perfil.

6. Alterar o conteúdo do arquivo `app.component.html` para puxar seu componente novo.

7. Modificar o CSS da página.

```
perfil: {...} = {  
    username: 'fulanosilva',  
    nome: 'Fulano da Silva',  
    local: 'Manaus',  
    idade: 28,  
    profissao: 'Desenvolvedor Web',  
    empresa: 'FpfTech',  
    publicacoes: 1145,  
    seguidores: 1743,  
    seguindo: 1549,  
    foto: 'https://encrypted-tbn0.gstatic.com/images?q=tbn:ANU9gbltWwYDQAAAAA&usqp=CAU'  
};
```



Obrigado!



    @fpftech.educacional