

# Angular

---

# Angular

---

- Angular é uma plataforma e framework para construção da interface de aplicações usando HTML, CSS e, principalmente, JavaScript, criada pelos desenvolvedores da Google.
- Possui alguns elementos básicos que tornam essa construção interessante.
- Dentre os principais, podemos destacar os componentes, templates, diretivas, roteamento, módulos, serviços, injeção de dependências e ferramentas de infraestrutura que automatizam tarefas, como a de executar os testes unitários de uma aplicação.

# Angular

---

- Alguns outros pontos dessa plataforma que merecem destaque são o fato de que ela é open source, possui uma grande comunidade, existem várias empresas utilizando e tem muito material de estudo para quem deseja se aperfeiçoar.

# Angular CLI

---

- Angular CLI é a ferramenta oficial para inicializar e trabalhar com projetos em Angular.
- Depois de instalar a Angular CLI, você precisa executar um comando para gerar um projeto e outro para servi-lo usando um servidor de desenvolvimento local para rodar sua aplicação.
- Como ocorre com a maioria das ferramentas de front-end nos dias de hoje, a Angular CLI foi criada com base no Node.js.

# Node.js

---

- O Node.js é uma tecnologia de servidor que permite que você execute o JavaScript no servidor e crie aplicações da web no lado do servidor.
- No entanto, o Angular é uma tecnologia de front-end.
- Portanto, mesmo que você precise instalar o Node.js em sua máquina de desenvolvimento, é apenas para a execução da CLI.

# Instalando Angular

---

- Primeiro, você precisa ter o Node e o npm instalados na sua máquina de desenvolvimento. Existem muitas maneiras de se fazer isso, tais como:
  - usar o NVM (Node Version Manager) para a instalação e trabalho com muitas versões do Node em seu sistema
  - usar o gerenciador de pacotes oficial do seu sistema operacional
  - instalar a partir do site oficial da web.
  - <https://nodejs.org/en/download/>

# Instalando Angular

---

- Certifique-se de que o Node esteja instalado em seu sistema executando o comando abaixo em um prompt de comando, que deverá exibir a versão instalada do Node:
  - `$ node -v`
- Depois, execute o seguinte comando para instalar a Angular CLI:
  - `$ npm install @angular/cli`

# AngularCLI

---

- Depois de instalar a Angular CLI, você pode executar diversos comandos. Começamos conferindo a versão instalada da CLI:
  - `$ ng version`
- Um segundo comando que talvez você queira executar é o comando `help`, que ajudará a obter uma lista de comandos a serem usados:
  - `$ ng help`



# AngularCLI

---

- Se der erro, siga os seguintes passos:
  - `npm uninstall -g angular-cli`
  - `npm uninstall --save-dev angular-cli`
  - `npm install -g @angular/cli`

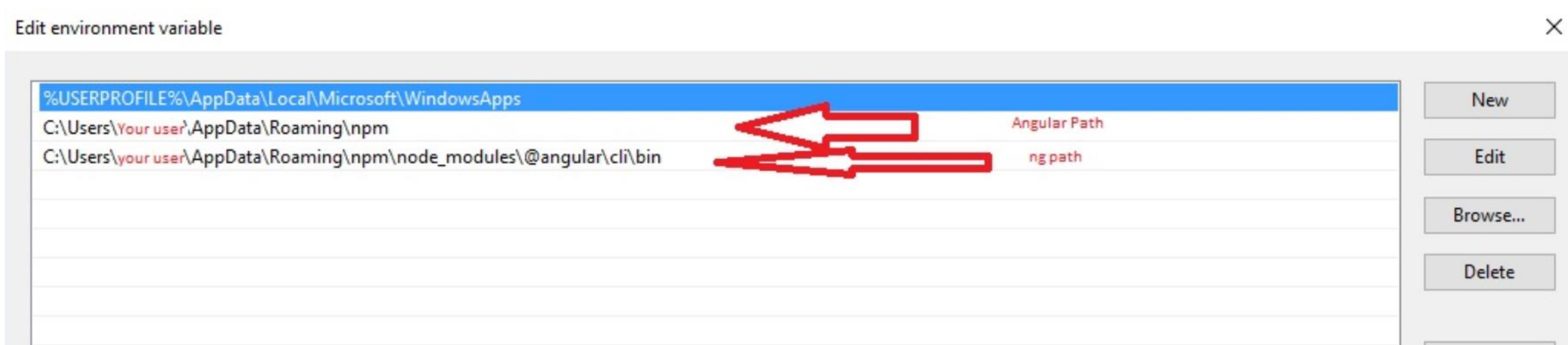
# AngularCLI

---

- Se der erro, siga os seguintes passos:
  - `npm uninstall -g angular-cli`
  - `npm uninstall --save-dev angular-cli`
  - `npm install -g @angular/cli`
    - Se der erro no install, tente:
      - `npm install -g @angular/cli --force`

# AngularCLI

- Se der erro, siga os seguintes passos:
  - Depois de instalar você entra em sistema, configuração avançadas do sistema, Variáveis de Ambiente, Path, Editar



## A CLI oferece os comandos a seguir:

---

- `add`: adiciona suporte a uma biblioteca externa em seu projeto.
- `build (b)`: compila uma aplicação do Angular em seu diretório de saída, chamado `dist/` no caminho de saída fornecido. Este comando deve ser executado dentro de um diretório de espaço de trabalho.
- `config`: obtém ou configura os valores de configuração do Angular.
- `doc (d)`: abre a documentação oficial do Angular ([angular.io](https://angular.io)) em um navegador, buscando uma palavra-chave determinada.
- `e2e (e)`: faz o `build` e serve uma aplicação do Angular, depois executa testes de ponta a ponta usando o Protractor.

## A CLI oferece os comandos a seguir:

---

- `generate (g)`: gera e/ou modifica arquivos com base em um esquema.
- `help`: lista os comandos disponíveis e suas descrições breves.
- `lint (l)`: executa as ferramentas de linting no código da aplicação em Angular em uma determinada pasta de projeto.
- `new (n)`: cria um espaço de trabalho e uma aplicação inicial em Angular.
- `run`: executa um destino personalizado definido em seu projeto.

## A CLI oferece os comandos a seguir:

---

- `serve (s)`: faz o build e serve sua aplicação, fazendo um novo build a cada alteração de arquivos.
- `test (t)`: executa os testes unitários em um projeto.
- `update`: atualiza sua aplicação e suas dependências. Consulte <https://update.angular.io/> (em inglês)
- `version (v)`: mostra a versão da Angular CLI.
- `xi18n`: extrai as mensagens i18n do código-fonte.

# Criando um projeto novo

---

- Você pode usar a Angular CLI para gerar rapidamente seu projeto em Angular executando o seguinte comando em sua interface de linha de comando:
  - `$ ng new OiMundo`
- A CLI perguntará Would you like to add Angular routing? ("Quer adicionar o roteamento do Angular?"). Você pode responder com y (Sim) ou n (Não). Não, neste caso, é a opção padrão. **Responda Sim.**
- Também será perguntado sobre o formato da folha de estilos (stylesheet) que você quer usar (por exemplo, o CSS). Escolha CSS e pressione Enter para continuar.

## Criando um projeto novo

---

- Depois disso, seu projeto estará criado com uma estrutura de diretórios e alguns arquivos com configurações e código.
- A maioria estará nos formatos TypeScript e JSON.



# Criando um novo projeto

---

- Vejamos a função de cada diretório/arquivo:
  - `/e2e/`: contém os testes end-to-end (simulação do comportamento do usuário) do site
  - `/node_modules/`: todas as bibliotecas de terceiros são instaladas nesta pasta usando `npm install`
  - `/src/`: contém o código-fonte da aplicação. A maior parte do trabalho será feita aqui
  - `/app/`: contém módulos e componentes
  - `/assets/`: contém os ativos estáticos, como imagens, ícones e estilos

# Criando um novo projeto

---

- Vejamos a função de cada diretório/arquivo:
  - `/environments/`: contém arquivos de configuração específicos do ambiente (produção e desenvolvimento)
  - `browserslist`: necessário para o autoprefixador para suporte ao CSS
  - `favicon.ico`: o favicon
  - `index.html`: o arquivo HTML principal
  - `karma.conf.js`: o arquivo de configuração para o Karma (uma ferramenta de testes)
  - `main.ts`: o arquivo inicial principal, a partir de onde o AppModule é iniciado

# Criando um novo projeto

---

- Vejamos a função de cada diretório/arquivo:
  - polyfills.ts: polyfills necessários ao Angular
  - styles.css: o arquivo de folha de estilos (stylesheet) global do projeto
  - test.ts: um arquivo de configuração para o Karma
  - tsconfig.\*.json: os arquivos de configuração para o TypeScript
  - angular.json: contém as configurações para a CLI

# Criando um novo projeto

---

- Vejamos a função de cada diretório/arquivo:
  - package.json: contém as informações básicas do projeto (nome, descrição e dependências)
  - README.md: um arquivo em markdown que contém a descrição do projeto
  - tsconfig.json: o arquivo de configuração para o TypeScript
  - tslint.json: o arquivo de configuração para o TSlint (uma ferramenta de análise estática)

# Como servir o projeto?

---

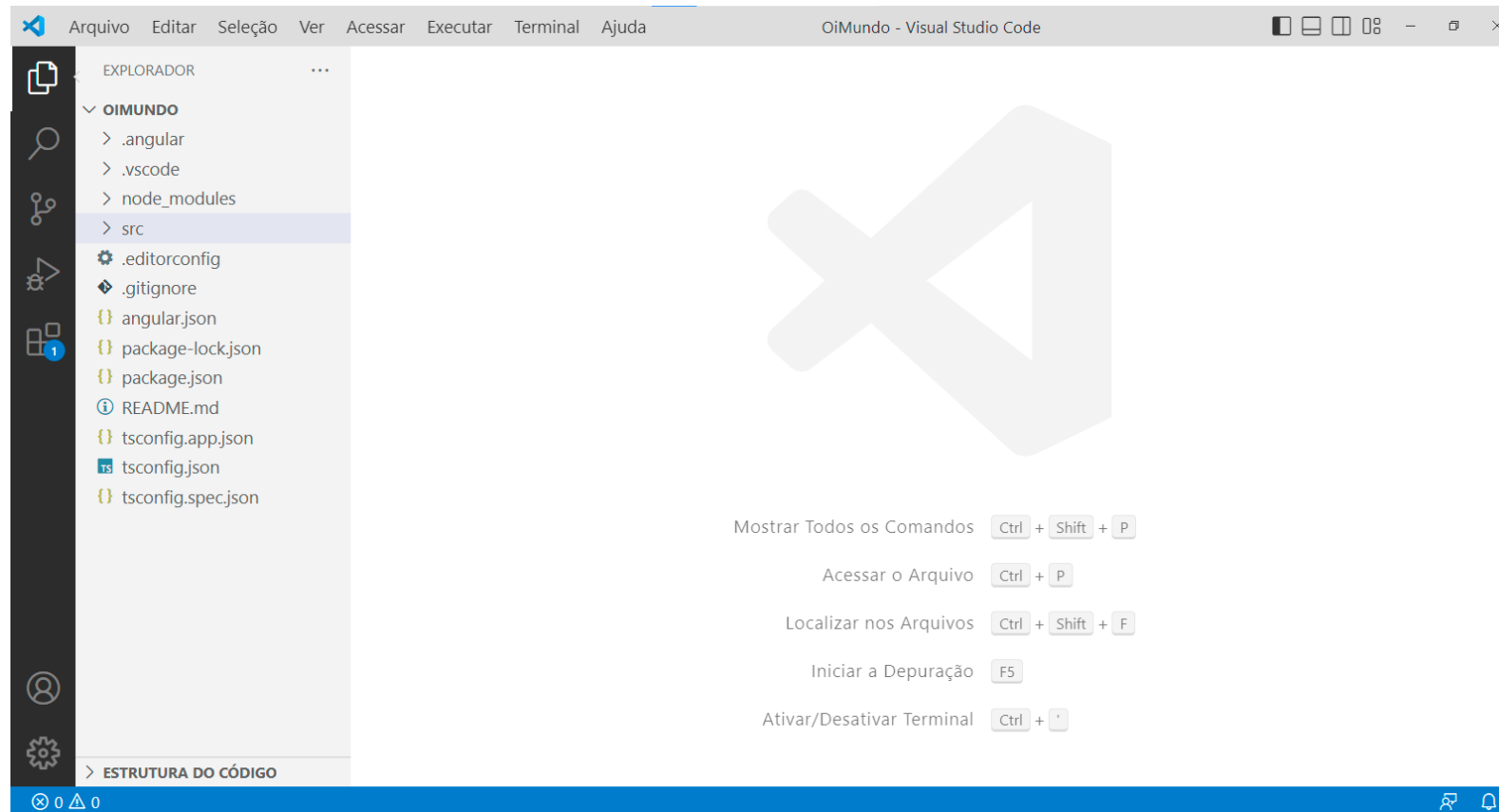
- A Angular CLI fornece um conjunto de ferramentas completo para desenvolver aplicações de front-end em sua máquina local.
- Assim, não é preciso instalar um servidor para servir o projeto — você pode, simplesmente, usar o comando `ng serve` a partir do terminal para servir seu projeto localmente.
  - `$ cd OiMundo`
  - `$ ng serve`

## Como servir o projeto?

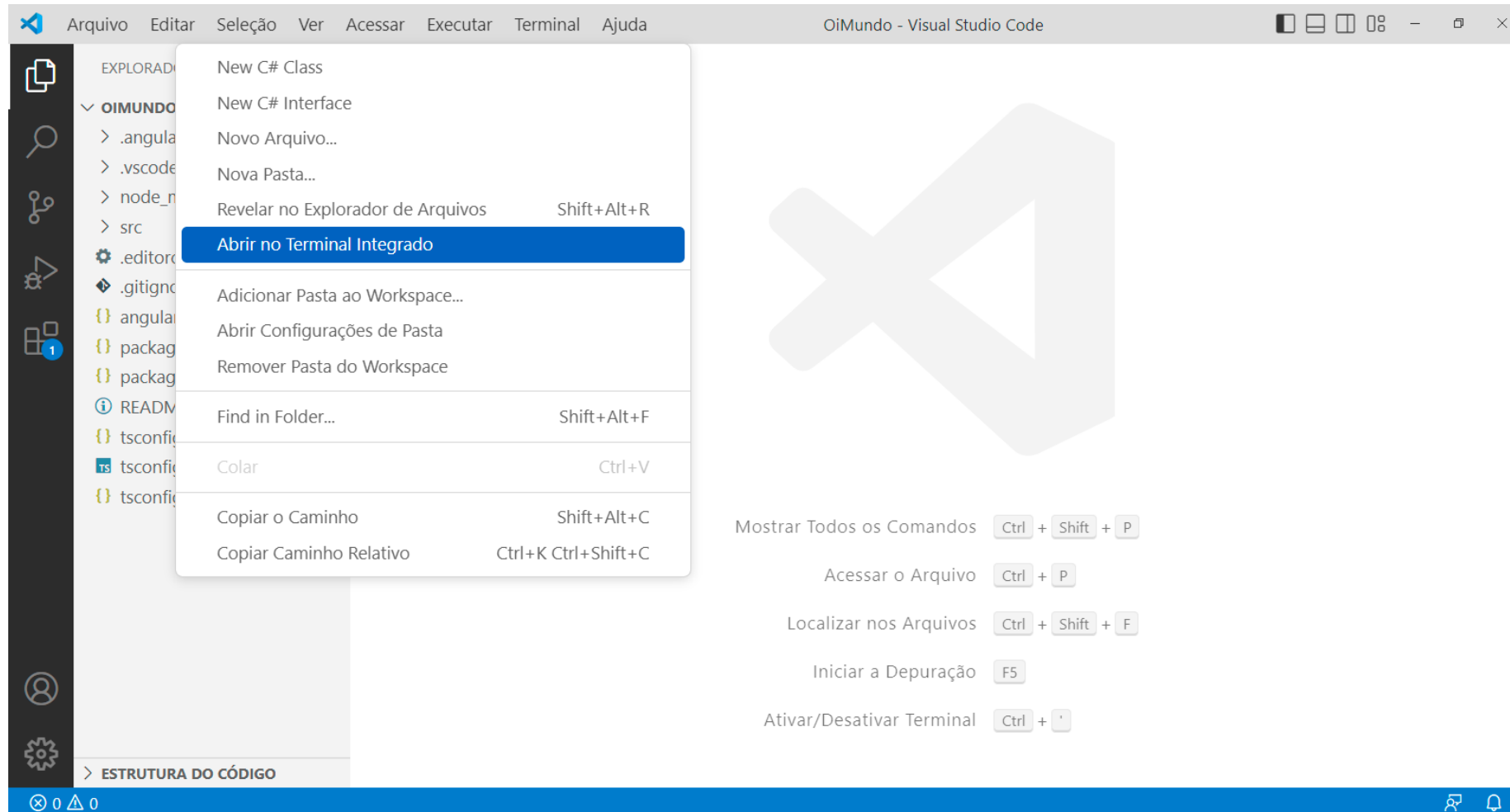
---

- Agora, você pode navegar até o endereço `http://localhost:4200/` para começar a mexer com sua aplicação de front-end.
- A página recarregará automaticamente caso você mude qualquer arquivo do código-fonte.
- Para abrir o projeto no Visual code digite:
  - `$ code .`

# Visual code

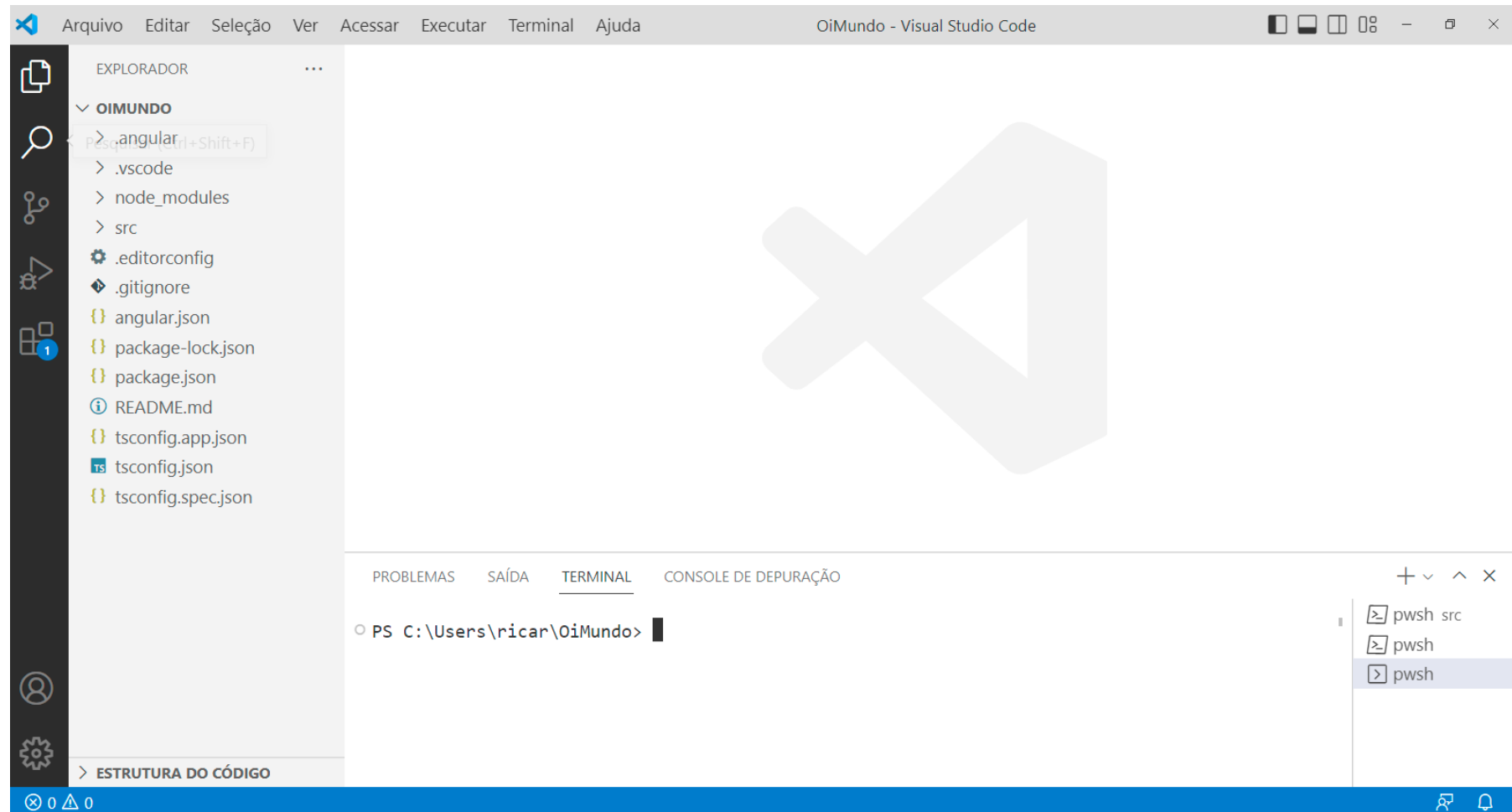


# Visual code





# Visual code



# Geração de artefatos do Angular

---

- A Angular CLI fornece um comando `ng generate` que ajuda os desenvolvedores a gerar artefatos básicos do Angular, como módulos, componentes, diretivas, pipes e serviços:
  - `$ ng generate component meu-componente`
- `meu-componente` é o nome do componente. A Angular CLI automaticamente adicionará uma referência a `components`, `directives` e `pipes` no arquivo `src/app.module.ts`.

# Geração de artefatos do Angular

---

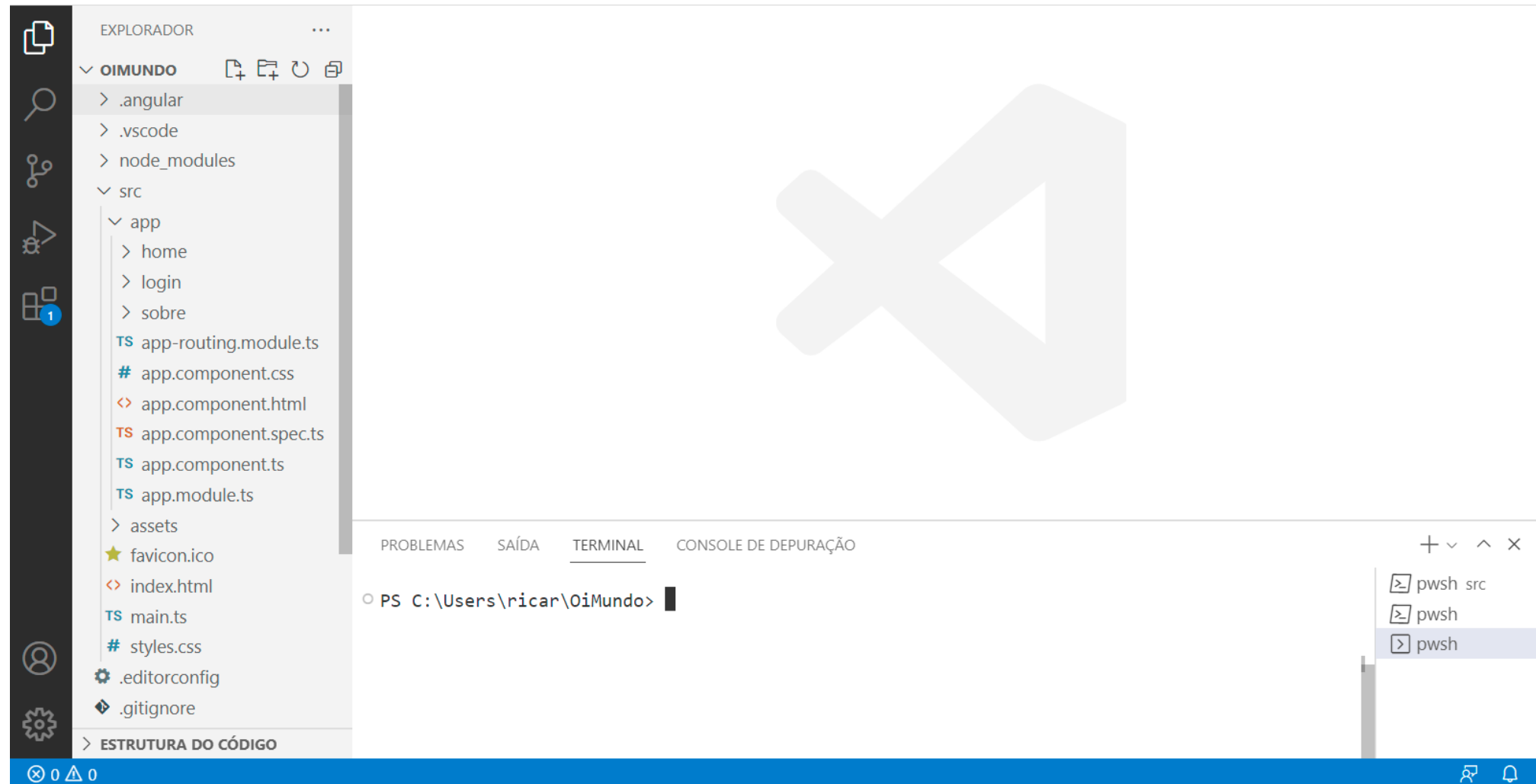
- Se quiser adicionar seu componente, diretiva ou pipe para outro módulo (que não seja o módulo principal da aplicação, `app.module.ts`), você pode simplesmente prefixar o nome do componente com o nome do módulo e uma barra assim :
  - `$ ng g component meu-modulo/meu-componente`
- `meu-modulo` é o nome de um módulo existente.

# Angular

---

- Digite:
  - \$ ng g c home
  - \$ ng g c login
  - \$ ng g c sobre

# Angular



# Angular

---

TS app-routing.module.ts ×



src > app > TS app-routing.module.ts > ...

```
1 import { NgModule } from '@angular/core';
2 import { RouterModule, Routes } from '@angular/router';
3 import { LoginComponent } from '../login/login.component';
4 import { SobreComponent } from '../sobre/sobre.component';
5 import { HomeComponent } from '../home/home.component';
6
7 const routes: Routes = [
8   {path: 'login', component: LoginComponent},
9   {path: 'sobre', component: SobreComponent},
10  {path: 'home', component: HomeComponent}
11 ];
12
13 @NgModule({
14   imports: [RouterModule.forRoot(routes)],
15   exports: [RouterModule]
16 })
```

# Angular

---

<> index.html ×



src > <> index.html > html > body > search

```
1  <!doctype html>
2  <html lang="en">
3  <head>
4    <meta charset="utf-8">
5    <title>Olá mundo</title>
6    <base href="/">
7    <meta name="viewport" content="width=device-width, initial-scale=1">
8    <link rel="icon" type="image/x-icon" href="favicon.ico">
9  </head>
10 <body>
11   <menu></menu>
12   <search></search>
13   <app-root></app-root>
14 </body>
15 </html>
16
```

# Angular

---

- Coloque texto no html de cada componente criado

<> sobre.component.html × TS sobre.component.ts

src > app > sobre > <> sobre.component.html > ...

Go to component

1 <p>Sobre o angular!</p>

2 <h1>Tudo isso fica bem legal pois as coisas vão se conectando</h1>



# Angular

---

- O app.component é onde é iniciado todo o projeto.

```
<> app.component.html •
src > app > <> app.component.html > router-outlet
Go to component
1 <h1>{{title}}</h1>
2 <nav>
3   <a routerLink="/">Index</a> <br>
4   <a routerLink="/home">Home </a><br>
5   <a routerLink="/login">Login </a><br>
6   <a routerLink="/sobre">Sobre</a><br>
7 </nav>
8 <router-outlet></router-outlet>
```

## Atividade:

---

- Assista os vídeos:
  - <https://www.youtube.com/watch?v=Yf0rC7dERjg>
- Assista, pelo menos, os 8 primeiros vídeos desta playlist
  - [https://www.youtube.com/watch?v=vJt\\_K1bFUeA&list=PLnDvRpP8Bnex2GQEN0768\\_AxZg\\_RalGmw](https://www.youtube.com/watch?v=vJt_K1bFUeA&list=PLnDvRpP8Bnex2GQEN0768_AxZg_RalGmw)

# Atividade

---

- Farei uma prova em cima do que foi abordado nos vídeos

# Atividade

---

- Farei uma prova em cima do que foi abordado nos vídeos
- **BRINCADEIRA!**

# Atividade

---

- Farei uma prova em cima do que foi abordado nos vídeos
- **BRINCADEIRA!**
- Mas seria bom vocês assistirem... hehe