

03 Angular CLI e ajustando o projeto



Transcrição

Vamos iniciar a codificação do Indexa?

Angular CLI e ajuste do projeto

Desta vez, vamos começar construindo o *front-end*. Já estamos habituados a abrir o *VS Code*, criar um arquivo `index.html`, um `style.css`, um `index.js`, começar a importar tudo e codificar, e tudo funciona, porque navegador entende o *JavaScript* que escrevemos.

No entanto, quando trabalhamos com *Angular*, o funcionamento não é exatamente assim. Vamos escrever usando *TypeScript* como linguagem principal, e o navegador não entende por padrão a linguagem do TypeScript. Dessa forma, precisamos de uma ferramenta para compilar o TypeScript para o JavaScript.

O estilo é um CSS normal com o qual estamos habituados. Porém, as coisas que vamos escrever durante essa jornada com o Indexa, o navegador não entende. Sendo

assim, são necessárias algumas ferramentas adicionais para alcançar esse objetivo.

Uma dessas ferramentas é a que chamamos de **Angular CLI**. Deixamos aberto o [link da documentação](https://angular.dev/tools/cli/setup-local)

(<https://angular.dev/tools/cli/setup-local>).

Essa documentação vai nos ensinar o que precisamos fazer para [instalar a ferramenta](https://angular.dev/tools/cli/setup-local#install-the-angular-cli) (<https://angular.dev/tools/cli/setup-local#install-the-angular-cli>) no nosso computador.

Instalando o **Angular CLI**

Note que quem precisa instalar o Angular CLI não é necessariamente o nosso projeto; vamos instalar globalmente, ou seja, dentro da pasta do *Node* no computador. Logo no início da documentação, são trazidas as instruções para a instalação. Devemos usar o seguinte comando:

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

COPIAR CÓDIGO

Com o terminal aberto, vamos digitar esse comando, mas com um detalhe. As versões do Angular são lançadas com frequência, e não necessariamente quando você começar

a codificar o Indexa estará na mesma versão usada no momento da gravação.

Então, para estarmos na mesma página e termos um cenário em comum, vamos dizer que queremos instalar o `@angular/CLI` na versão `@17.0.3`. Se você instalar exatamente essa versão, não teremos nenhum problema de comunicação, você vai escrever a mesma coisa que nós e irá funcionar.

```
npm install -g @angular/cli@17.0.3
```

[COPIAR CÓDIGO](#)

Criando um projeto *Angular*

Ao terminar a instalação, estaremos aptos para começar a utilizar as ferramentas que ele disponibiliza. Vamos usar algumas durante este curso, e a primeira delas será **criar um projeto Angular**.

Lembre-se que não é sobre criar os arquivos `index.html` e `styles.css` e começar a codificar. Vamos pedir para o Angular CLI criar esse projeto base.

Estamos no terminal, na pasta do usuário, mas não é aqui que queremos estar.

Queremos estar na área de trabalho, então executamos o comando abaixo:

```
cd Desktop/
```

[COPIAR CÓDIGO](#)

Feito isso, vamos limpar o terminal e verificar se tem algo no *desktop* com o comando `ll`.

```
ll
```

[COPIAR CÓDIGO](#)

Com o desktop limpo, vamos pedir para o Angular criar a aplicação para nós. Para isso, vamos usar o comando `ng`. `ng` é exatamente o comando que acabamos de instalar, disponibilizado pelo Angular CLI.

Para criar uma aplicação nova, vamos usar o comando `ng new`. Nesse caso, `ng` é o comando, `new` é o primeiro argumento, ou parâmetro, e o último argumento será o nome do projeto. O terminal sugere usar o `indexa`, porque já escrevemos e executamos esse comando algumas vezes antes.

```
ng new indexa
```

[COPIAR CÓDIGO](#)

Ao pressionar a tecla "Enter", surgirá uma pergunta: "Qual formato de folha de estilo você gostaria de usar?". Você quer usar CSS puro, *Sass*, *Less*? Nesse caso, vamos usar **CSS** puro.

Logo na sequência, ele pergunta se vamos lidar com renderização do lado do servidor. Nesse momento, não vamos, então responderemos "Não". Definido isso, a ferramenta vai criar o projeto base e instalar todas as dependências.

Quando trabalhamos em um projeto que tem um `package.json` que configura as dependências, ao executar o comando `npm install`, ele vai baixar tudo de acordo com o sistema operacional e com a versão do Node, e colocar na pasta "node_modules". É isso que o comando `ng` faz para nós: ele gera o projeto e instala as dependências.

Trabalhando com o Angular

Ao terminar, estaremos aptos a começar a trabalhar e entender o que o Angular gera. Vamos limpar o terminal novamente: no *Mac*, usamos o atalho "Command + K" para fazer isso; no *Windows*, com o atalho "Ctrl +

L", conseguimos algo parecido a depender do terminal usado.

Estamos na área de trabalho e, para executar essa aplicação que o `ng` acabou de criar, precisamos entrar na respectiva pasta. Então, vamos digitar o comando `cd indexa/`.

```
cd indexa/
```

[COPIAR CÓDIGO](#)

Antes não tínhamos nada e agora temos uma pasta "indexa" que é o resultado do comando `ng new`. Para executar esse projeto e subir essa aplicação, vamos usar o comando `ng serve`. Com isso, pedimos para o Angular disponibilizar o projeto que acabamos de gerar.

```
ng serve
```

[COPIAR CÓDIGO](#)

Como retorno, ele informa que gerou os arquivos `polyfills.js`, `main.js` (formato que o navegador entende) e `styles.css`. Ele informa também o tamanho de cada um e o tamanho total de todos os arquivos.

Além disso, ele diz que construiu a aplicação, o que levou 1.7 segundos, e é indicado que o ***watch mode*** está ligado, ou seja, ele presta atenção nesses arquivos que podem ser alterados. Se quisermos acessar, basta ir até o endereço

["http://localhost:4200/](http://localhost:4200/)
[\(http://localhost:4200/\)](http://localhost:4200/)".

Ao abrir esse link, temos uma aplicação com link para documentação, alguns tutoriais, documentação do CLI, e há a mensagem "*Hello, indexa*", como se fosse o "*Hello, world!*", mas com o Angular.

Conclusão

Parabéns, a sua aplicação está rodando! Preparamos os alicerces da nossa aplicação e agora temos um projeto Angular base para começar a construir o Indexa.

Agora precisaremos começar a lidar com os arquivos HTML e TypeScript que o Angular entende. Vamos fazer isso? **Te esperamos no próximo vídeo!**