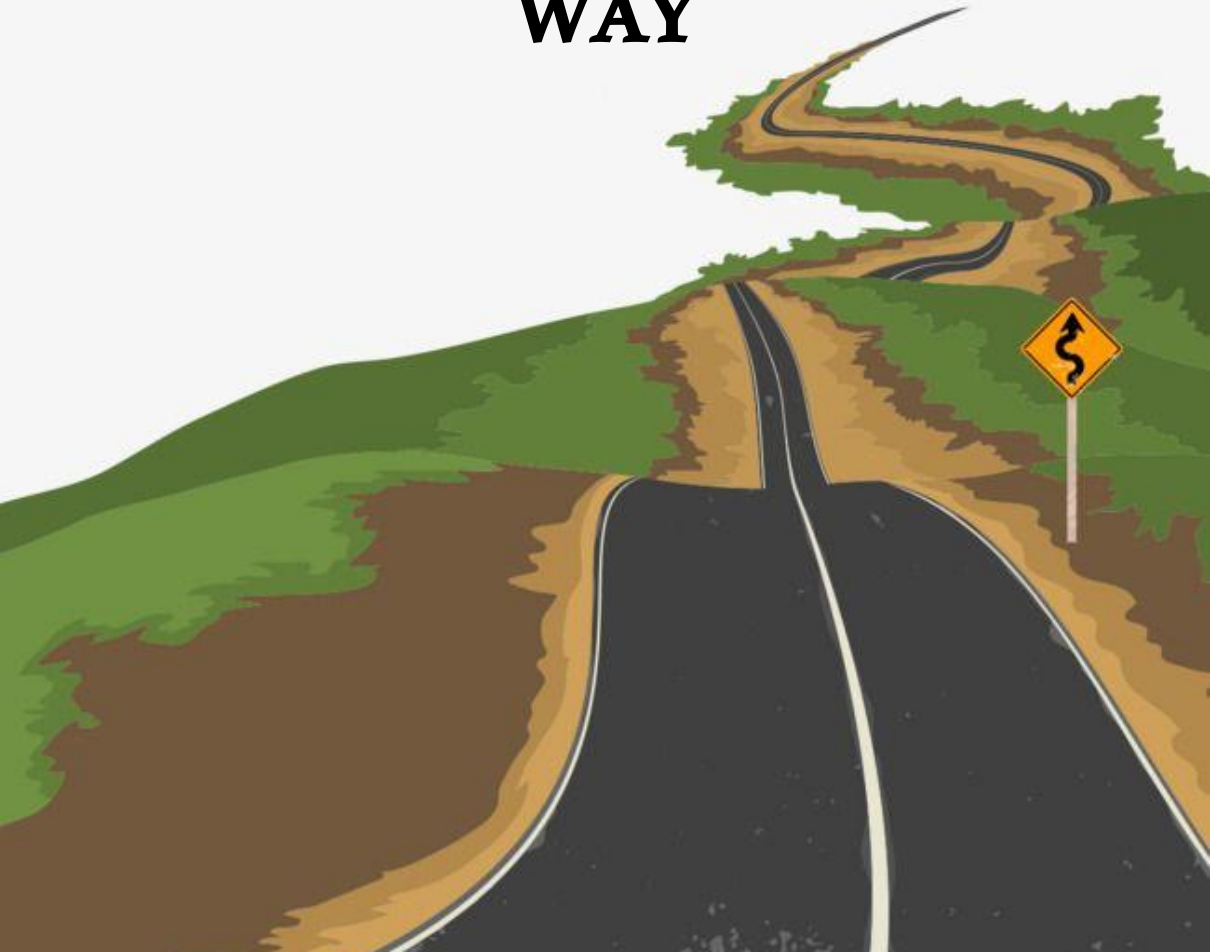


Danilo Andrade



WAY



Se você ver este ícone

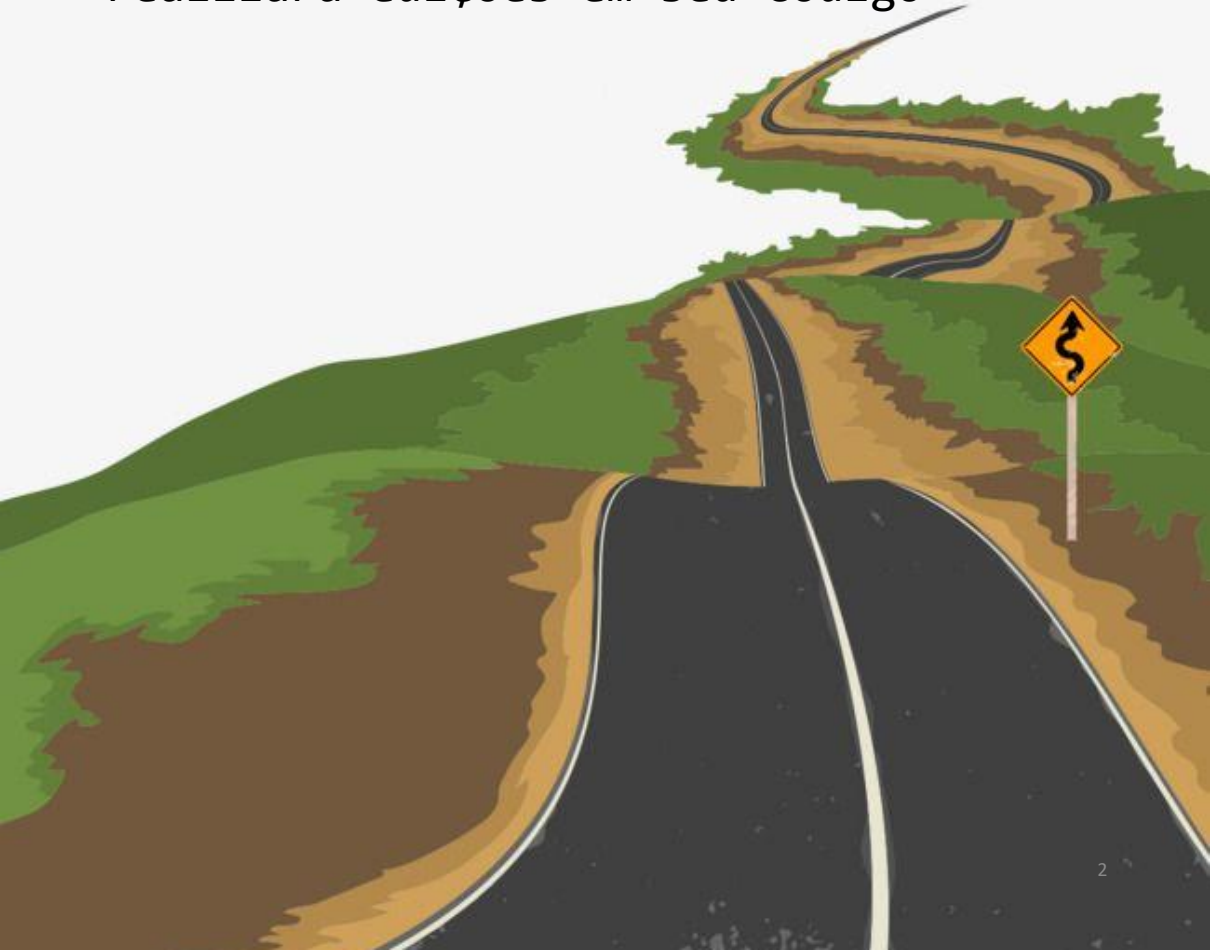


A página em questão estará tratando de explicações teóricas (mas que tem sua utilidade prática)

Se você ver este outro ícone



A página em questão estará tratando de uma atividades pratica. Portanto você realizará edições em seu código



Criando um novo projeto



Com seu terminal aberto, navegue até o diretório onde você deixará a pasta do seu projeto.

No meu caso eu vou abrir a partição I:\ que é a partição onde guardo meu projetos de programação

```
> cd I:\
```

Continua ...

Criando um novo projeto



Escreva o seguinte no terminal:

```
> ng new nome-do-projeto
```

Duas perguntas serão feitas.

Uma por vez

```
? Which stylesheet format would you like to use?
```

Use o cursor do teclado para navegar entre as opções. Clique em **Enter** para escolher. Recomendo que fique com a primeira, CSS.

Continua ...

Criando um novo projeto



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

Y para yes - Sim

N para no - Não

Dizer sim para esta questão significará que...

Dizer não para esta questão significará que...

Apenas aguarde até que todos os seus arquivos sejam criados

Criando um novo projeto



Agora navegue para dentro da pasta do seu projeto

```
> cd nome-do-projeto
```

Pronto, agora você poderá visualizar o conjunto de arquivos criados

Continua ...

Rodando sua aplicação em um servidor local



Escreva o seguinte no terminal:

```
> ng serve
```

E aguarde até que no terminal seja disponibilizada uma portal local para você visualizar sua aplicação.

Fique de olho, normalmente a porta local é `http://localhost:4200/`

Toda edição que você fizer em seu código poderá ser visualizada em tempo real em seu navegador



Componentes

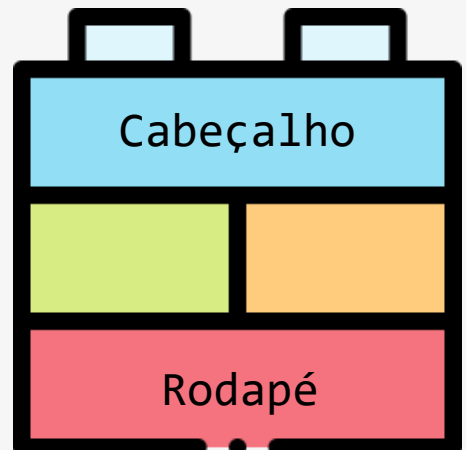


Projetos Angular são baseado em componentes

Cada componente funciona como um bloco de construção que você pode usar para construir partes específicas de uma página da web ou de um aplicativo

Cada elemento da sua página pode ser um componente diferente.

Por exemplo, o cabeçalho pode ser um componente, o rodapé outro componente.



Tudo feito em arquivos separados que depois podem ser combinados

Chamando o primeiro componente



Escreva o seguinte no terminal:

```
> ng g c componentes/home
```

```
> ng generate component componentes/home
```

O trecho componentes/home cria:

Uma pasta chamada componentes

E os arquivos do componente Home

A estrutura de arquivo de um componente



```
▼ home
  # home.component.css
  <> home.component.html
  TS home.component.spec.ts
  TS home.component.ts
```

Ao gerar uma componente são criados 4 arquivos para ela.

Cada arquivo um tem uma função que explico a seguir...

Continua ...

O arquivo .css de um componente



Este é o arquivo de estilo do componente. Ele contém as regras de estilo CSS (ou Sass/SCSS) específicas para o componente.

Você pode definir estilos para os elementos HTML dentro do componente, como cores, tamanhos, margens, etc.

A vantagem de usar estilos específicos do componente é que eles permanecem encapsulados no escopo do componente, evitando conflitos com outros estilos no aplicativo.

O arquivo .html de um componente



Este arquivo contém o modelo HTML do componente.

Ele define a estrutura e o layout visual do componente.

Dentro deste arquivo, você pode escrever HTML para criar elementos como divs, botões, formulários, etc.

Escreva algo como...

```
1 <p> Hello World! </p>
```

O arquivo .ts de um componente



Este é o arquivo TypeScript do componente.

Ele contém a lógica do componente, incluindo propriedades, métodos e qualquer outro código relacionado ao comportamento do componente.

Aqui, você define a classe do componente, importa dependências necessárias, declara variáveis, implementa métodos e lida com eventos

O arquivo .spec.ts de um componente



Escreva algo aqui

Configurando o app.routes.ts



Organize seu arquivo `app.routes.ts`
como mostrado abaixo

```
1 import { Routes } from '@angular/router';
2 import { HomeComponent } from '../components/home/home.component';
3
4 export const routes: Routes = [
5   {
6     path: "",
7     component: HomeComponent
8   }
9 ];
```

Desta forma seu `app.routes.ts` apontará
o componente Home como o arquivo
inicial de seu WebApp

Configurando o arquivo app.component.html



Remova todo o conteúdo do arquivo
app.component.html

Deixe apenas o código
mostrado ao lado

```
<router-outlet />
```


Rodando sua aplicação após as primeiras edições



Lembra o que fizemos para visualizar nossa aplicação rodando em nosso servidor local?:

```
> ng serve
```

Se você não fechou seu projeto, basta ir novamente em <http://localhost:4200/> e ver seu projeto editado

Caso você tenha fechado seu projeto, basta digitar novamente “ng serve”, clicar em **Enter**, e o servidor local irá mostrar seu aplicação