**Dia 2 - Estilização e Data Binding**

* 1. Importando o módulo de formulários no seu projeto
  2. Abra o arquivo do módulo ao qual você deseja adicionar o módulo FormsModule. Normalmente, o arquivo do módulo é nomeado **nome-do-modulo.module.ts**.
  3. Importe o módulo FormsModule adicionando o seguinte import no início do arquivo:
* typescriptCopy code
* import { FormsModule } from '@angular/forms';
  1. Adicione o módulo FormsModule à lista de imports no decorator **@NgModule**, como mostrado abaixo:
* typescriptCopy code
* import { NgModule } from '@angular/core';
* import { FormsModule } from '@angular/forms';
* import { CommonModule } from '@angular/common';
* import { MeuComponente } from './meu-componente.component';
* @NgModule({
* declarations: [
* MeuComponente
* ],
* imports: [
* CommonModule,
* FormsModule // Adicionando o módulo FormsModule aqui
* ],
* exports: [
* MeuComponente
* ]
* })
* export class MeuModulo { }
  1. Salve o arquivo e o módulo FormsModule agora estará disponível em seu módulo.

O FormsModule é um módulo padrão do Angular que permite o uso de recursos de formulário, como o ngModel. Ele deve ser importado em qualquer módulo que utilize formulários em seu template.

* 1. Alterando o Logo

Salve a imagem abaixo dentro da pasta src/assets

Para mudar o ícone favorito (fav icon) de um aplicativo Angular para um arquivo .png, siga estes passos:

* 1. Coloque o arquivo .png no diretório "src/assets" do seu projeto Angular.
  2. Abra o arquivo "index.html" localizado em "src" e localize a seguinte linha de código:

<link rel="icon" type="image/x-icon" href="favicon.ico">

* 1. Substitua o valor do atributo "href" para o caminho relativo do seu arquivo .png, ficando assim:

<link rel="icon" type="image/png" href="assets/seu\_arquivo.png">

* 1. Salve o arquivo "index.html".

Agora, quando você executar o aplicativo Angular, o ícone favorito será o arquivo .png que você adicionou na pasta "src/assets". Certifique-se de que o nome do arquivo e o caminho estejam corretos.

* 1. Maneiras de se fazer Data biding

3.1 **Two-Way Binding**: Utilizando [( )], é possível fazer a conexão de dados bidirecional entre as propriedades do componente e a view.

para associar um item de formulario com uma variavel da classe, utilizamos a caixinha de banana

[()]

[(ngModel)]="tweetMessage"

3.2 **Event Binding**: Através de parênteses ( ), é possível associar eventos HTML a métodos do componente.

para associar um evento do html com uma função javascript é utilizado o ()

(input)="changeTweet()"

3.3 **Property Binding**: Através de colchetes [ ], é possível atribuir valores de propriedades do componente para as propriedades HTML.

para assoaciar uma propriedade do Angular a parte lógica do javascript use [ ]

[maxlength]="tweetMaxLength"

3.4 **Interpolação (Interpolation)**: Utilizando chaves duplas {{ }}, é possível inserir valores de propriedades do componente dentro do template HTML.

para escrever o valor de uma variavel no html utilize os bigodinhos duplos {{ }}

{{tweetRemain}}