**INSTALAÇÃO ANGULAR CLI**

Instale a última versão do Angular CLI com o seguinte comando (talvez seja necessário executar o comando como administrador do sistema):

npm install -g @angular/cli@latest

\*\*Requer NodeJS versão 12 ou superior.

Para verificar se a instalação está correta, execute o seguinte comando para verificar a instalação:

ng --version

Você deverá receber uma mensagem no terminal com a versão do Angular CLI, que deverá ver na mensagem exibida a versão do Angular, algo parecido com "Angular CLI: 13.0.4", significando que a versão 13.x.x foi instalada no sistema.

Acompanhe normalmente as aulas para criar o seu projeto, pois o mesmo já estará utilizando por padrão o Angular 13.

\*\*O arquivo zip em anexo projetos-final-angular-13.zip contém a versão final de todos os projetos configurados para o Angular 13, que poderão ser utilizados no lugar dos arquivos demonstrados no vídeo anterior (formação-angular.zip).

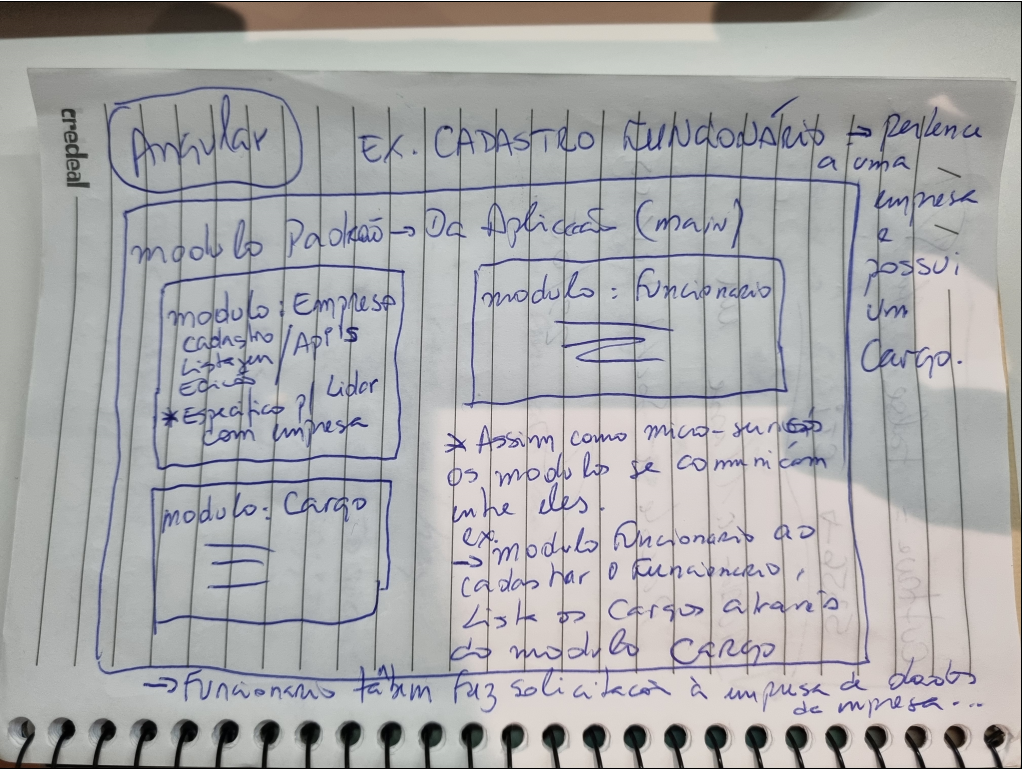
**Notas para o Angular CLI 13 do Angular 13**

Nos próximos vídeos mencionarei sobre um arquivo chamado .angular-cli.json, que é o arquivo de configuração do Angular CLI (explicarei sobre ele em detalhes no vídeo).

Na versão 13 do Angular e Angular CLI, o arquivo .angular-cli.json que mencionarei nos vídeos a seguir foi renomeado para "angular.json", portanto caso você esteja seguindo o curso com o Angular 13, sempre que eu mencionar o primeiro arquivo, faça as alterações no angular.json.

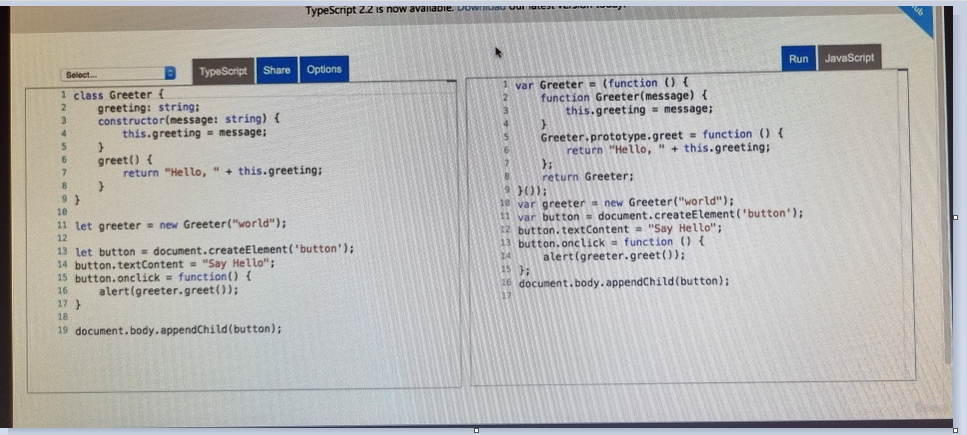
Embora a estrutura de ambos seja um pouco diferente, ela não impactará em nada, pois indicarei como fazer as alterações necessárias nele da forma correta.

**ESTRUTURA DO ANGULAR**



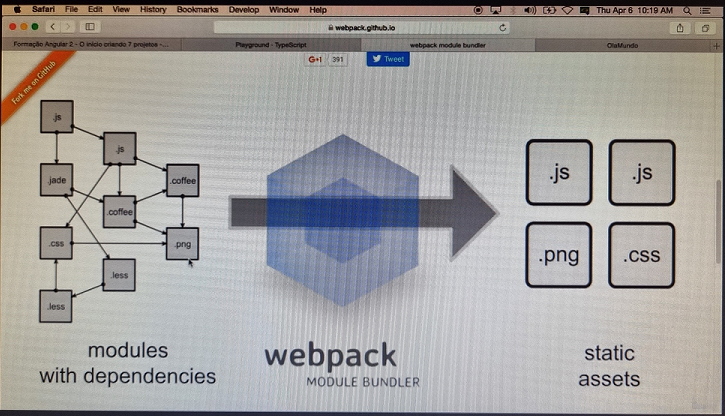
A ideia do angular é criar pequenas unidades de código com um propósito e que nós chamaremos de módulo. Sempre partiremos de módulo e o angular sempre tem um módulo principal que vai conseguir agregar. Desse modo o teu código vai ficar muito bem acoplado de fácil manutenção e muito bem organizado. É muito importante que você tenha em mente esse conceito e essa organização de módulos do angular.

**TYPE SCRIPT**



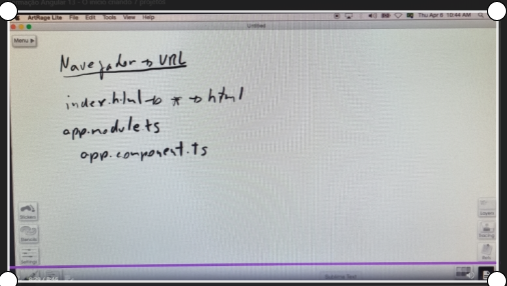
Forma amigável em escrever código fontes que será compilado para o java script.

**WEBPACK**



É o gerenciador de módulos. Ele vai pegar todo o nosso código e vai fazer uma mágica com eles e migrar para uma forma onde o navegador irá conseguir interpretar e rende rizar a página.

**CARREGAMENTO DE UMA PÁGINA**



Assim você tem lá o seu navegador e você acessou determinada URL, o index.htm é carregado.

app.module.ts -> é executado o arquivo de bootstrap porque está condicionado a executar o nosso arquivo App Mode que é o arquivo de módulo principal do nosso sistema. Esse arquivo ele vai ter uma referência para o componente que é o app componente que é o outro arquivo que contém a nossa página.

app.component.ts -> esse app concorrente ele contém o HTML que contém apenas a instrução "Olá Mundo".

O index.html possui uma tag principal de aplicação então quando esse modulo e componente forem carregados, aquela tag que está aqui no meu HTML ela será automaticamente substituída pelo HTML do componente e é por isso que nós visualizamos a página ou aquela mensagem do Olá mundo.

**CALCULADORA**

Procedimentos pós criação

Após criar o projeto com o comando “ng new calculadora”, executar o comando ng server para verificar se a aplicação está executando.

No arquivo de teste /e2e/src/app.e2e-spec.ts comentar a linha que realiza o teste do título da página.

  it('should display welcome message', async () => {

    await page.navigateTo();

    expect(await page.getTitleText()).toEqual('calculadora app is running!');

  });

No diretório do componente padrão, devemos comentar o teste no arquivo /src/app/app.component.spec.ts.

  it(`should have as title 'calculadora'`, () => {

    const fixture = TestBed.createComponent(AppComponent);

    const app = fixture.componentInstance;

    expect(app.title).toEqual('calculadora');

  });

  it('should render title', () => {

    const fixture = TestBed.createComponent(AppComponent);

    fixture.detectChanges();

    const compiled = fixture.nativeElement;

    expect(compiled.querySelector('.content span').textContent).toContain('calculadora app is running!');

  });

No arquivo /src/app/app.component.ts retirar a propriedade title do AppComponent

export class AppComponent {

  //title = 'calculadora';

}

Na view do projeto, removemos a referência da propriedade “title” -> /src/app/app.component.html

<h1>

  {{ title }}

</h1>