**ANGULAR 1**

Versão do Node: v8.11.2

Feitosa executar( somente ) os 4 passos abaixo no respectivos diretórios:

1. Instalação dos pacotes necessários para o projeto

**C:\workspace\_angular\curso-angular\alurapic>npm install**

1. Instalação do bootstrap versão 4.1.1

**C:\workspace\_curso\curso-angular\alurapic>npm install [bootstrap@4.1.1](mailto:bootstrap@4.1.1)**

1. Instalação da biblioteca de fontes e icones

**C:\workspace\_curso\curso-angular\alurapic>npm install font-awesome@4.7.0 -S**

1. Instalação dos pacotes necessários para o projeto

**C:\workspace\_curso\curso-angular\api>npm install**

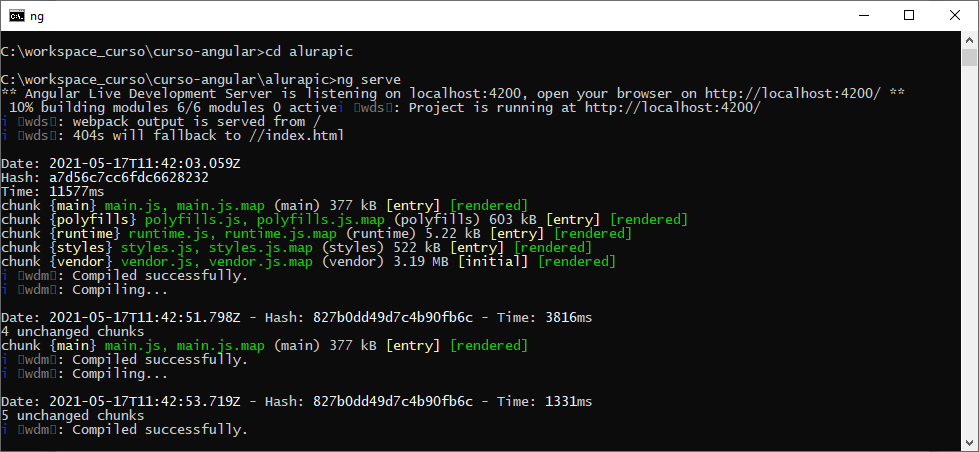
**URL da Home: <http://localhost:4200/#/home>**

**User: flavio**

**Passw: 123**

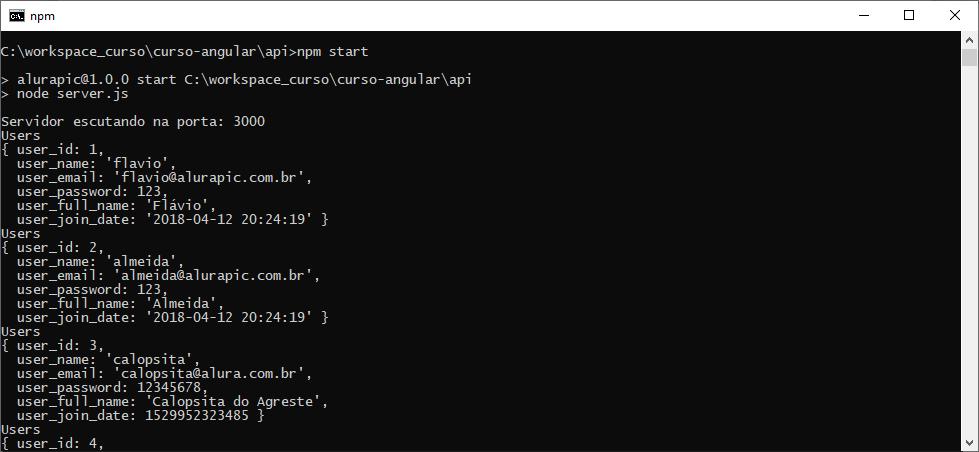
1. **Comando para startar o WebAPI( tem que startar os dois**

C:\workspace\_curso\curso-angular\alurapic>ng serve



1. C:\workspace\_curso\curso-angular\api>npm start

**http://localhost:3000/flavio/photos**



Aula 3:

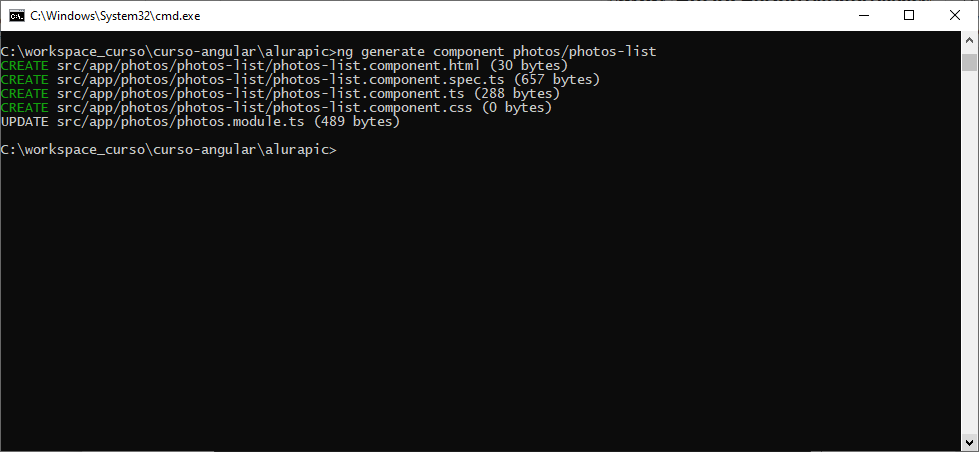
Tópicos vistos na aula:

* Consumir uma Web API através do serviço HttpClient
* Injeção de dependência e a importância de providers
* Isolamento da lógica de acesso à Web API em classe de serviço
* Tipagem do retorno da API através de interface e sua vantagem
* Componentes possuem ciclo de vida
* A fase OnInit

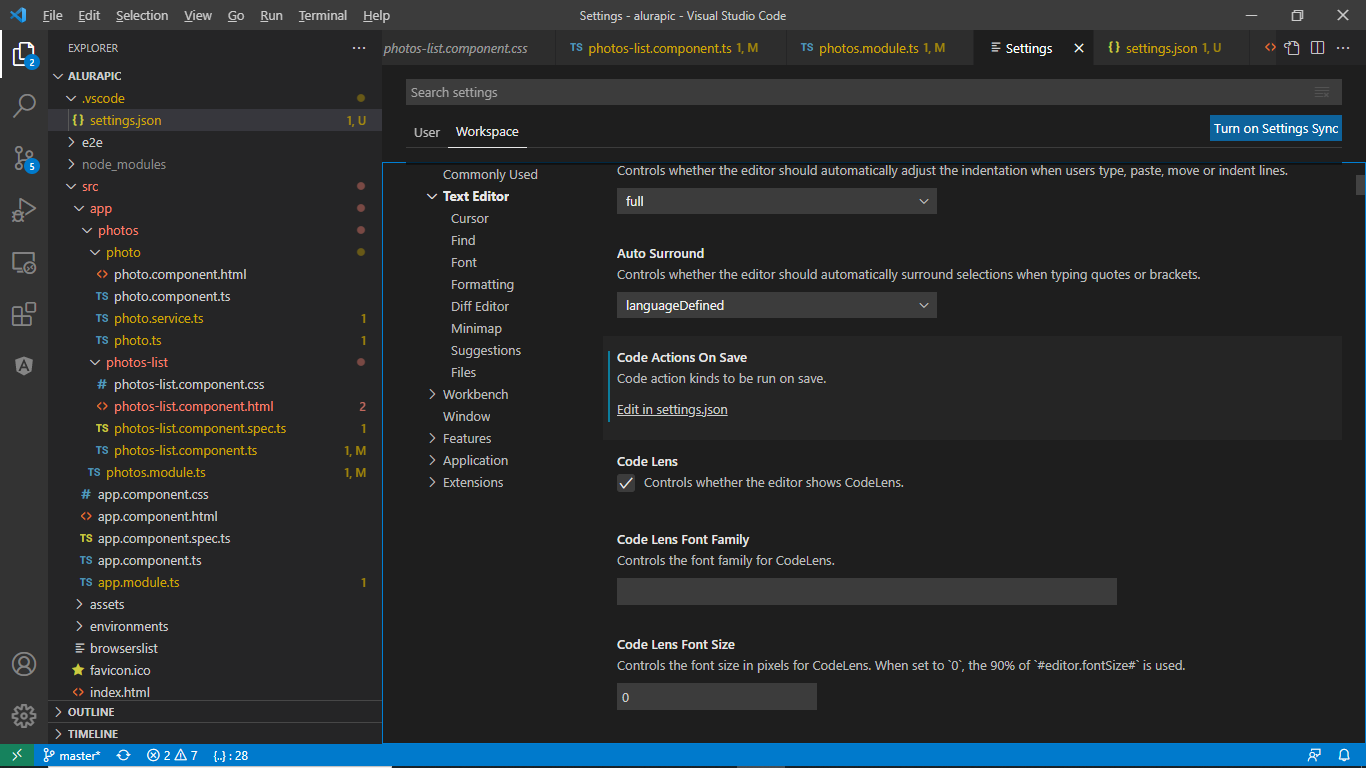
Gerando o componente pelo prompt( este comando gera os artefatos necessários ao componente)

Obs.: antes deve parar o server( Ctr+C)

C:\workspace\_curso\curso-angular\alurapic>ng generate component photos/photos-list

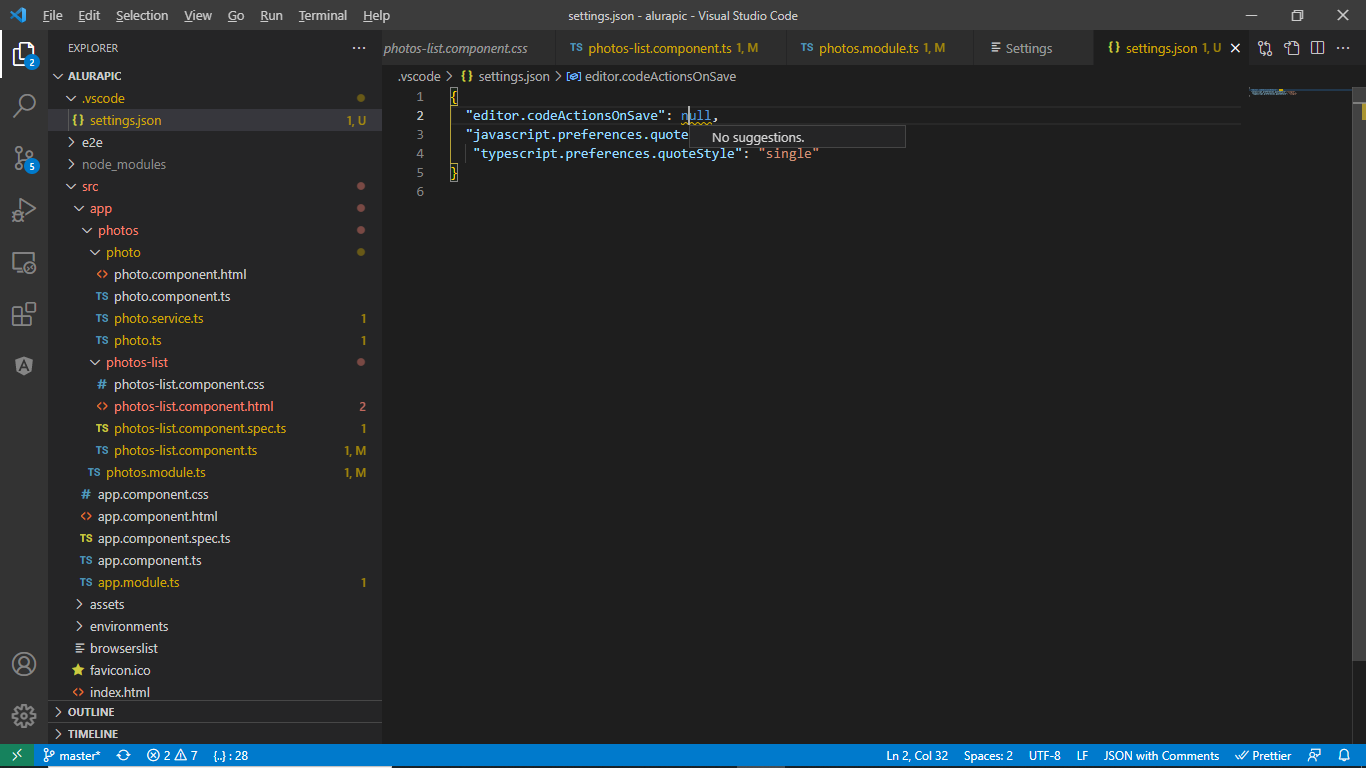


Para alterar a configuração de usuário para os imports virem com aspas simples (‘) ao inves de aspas duplas (“), inserir em File>Preferences>Settings>Workspace>Text Editor> Code actions on Save Edit in settings.json



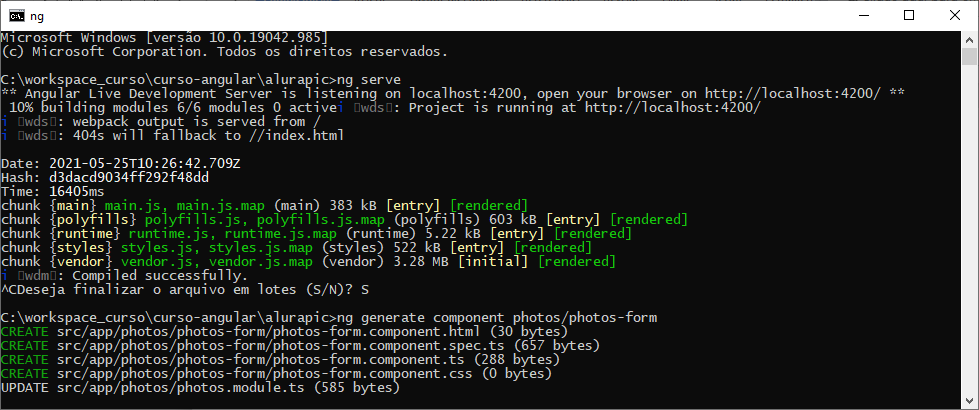
Tópicos abordados no capítulo:

* BrowserModule vs CommonModule
* Single Page Application e rotas no lado do navegador
* O módulo RouterModule
* A diretiva router-outlet como grande lacuna para exibição de outros componentes
* Módulo de rotas e definição de rotas
* Como lidar com páginas 404
* Parametrizando rotas e como obter valores do segmento parametrizado



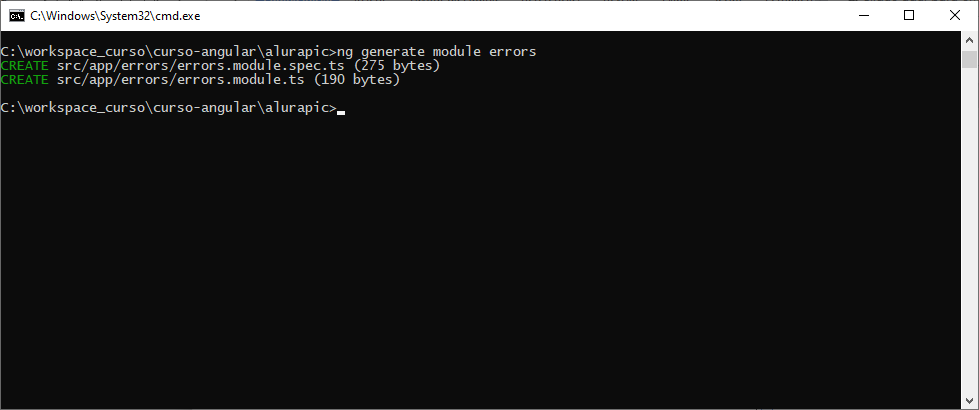
Criando form

C:\workspace\_curso\curso-angular\alurapic>ng generate component photos/photos-form



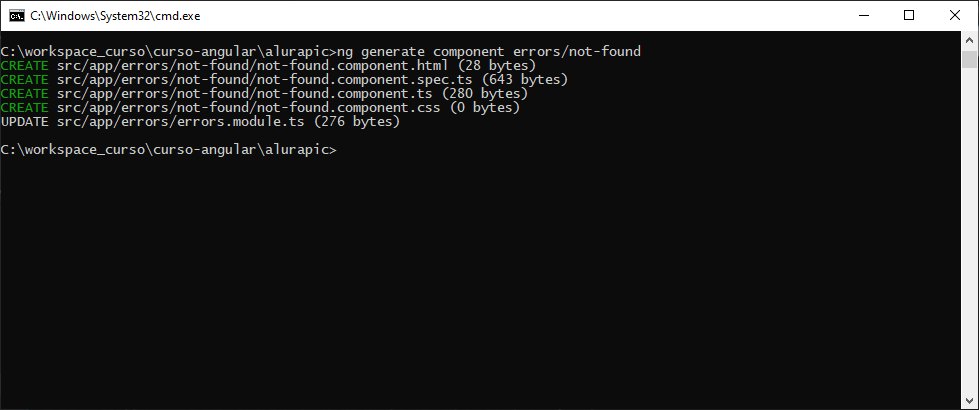
Criando um modulo de errors

C:\workspace\_curso\curso-angular\alurapic>ng generate module errors



Cirando o component not-found ( para o modulo de errors)

C:\workspace\_curso\curso-angular\alurapic>ng generate component errors/not-found

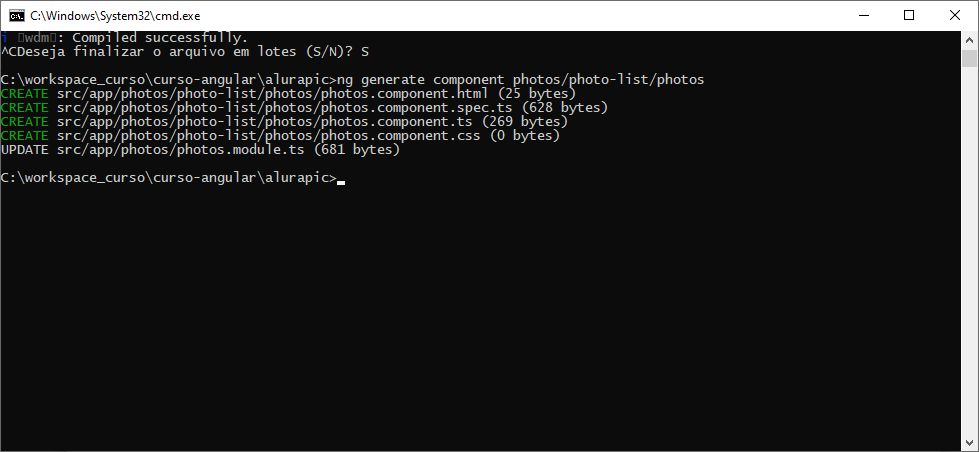


Tópicos abordados no capítulo:

* Novo componente para listar photos
* Adequação dos dados recebidos pelo componente
* Quando a fase OnInit não é suficiente
* A interface OnChanges, e como interagir com SimpleChanges

Gerar o componente de photos dentro do photo-list

C:\workspace\_curso\curso-angular\alurapic>ng generate component photos/photo-list/photos



Aprendemos que Pipes ("tubos", em português) podem gerar transformações nos dados, e que podemos criar os nossos. Para que seja um Pipe, sua classe deve ser anotada com o decorator @Pipe, além de implementar o método transform(), que possui determinada assinatura (parâmetros).

Para garantirmos que nossa classe implemente este método e seja avisada pelo TypeScript caso não isto não ocorra, podemos utilizar a interface:

export class FiltroPorTitulo implements PipeTransform {}

Pode ser usado o pipe com algum expressao para por exemplo deixar a palavra em caixa alta

Exemplo:

nome = ’giva’;

{{nome | uppercase}}

GIVA

PAREI EM RESOLVERS

**28/05**

**Url do json : http://localhost:3000/flavio/photos**

**Importante:**

Utilizar este metodo para não deixar a memória alocada

 ngOnDestroy(): void {

    this.debounce.unsubscribe();

  }

Este ngOnDestroy() é utilizado desalocar a memoria para a utilização this.debounce deste metódo no ngOnInit

 ngOnInit(): void {

    this.photos = this.activatedRoute.snapshot.data['photos'];

    this.debounce

    .pipe(debounceTime(300))

    .subscribe(filter => this.filter = filter);

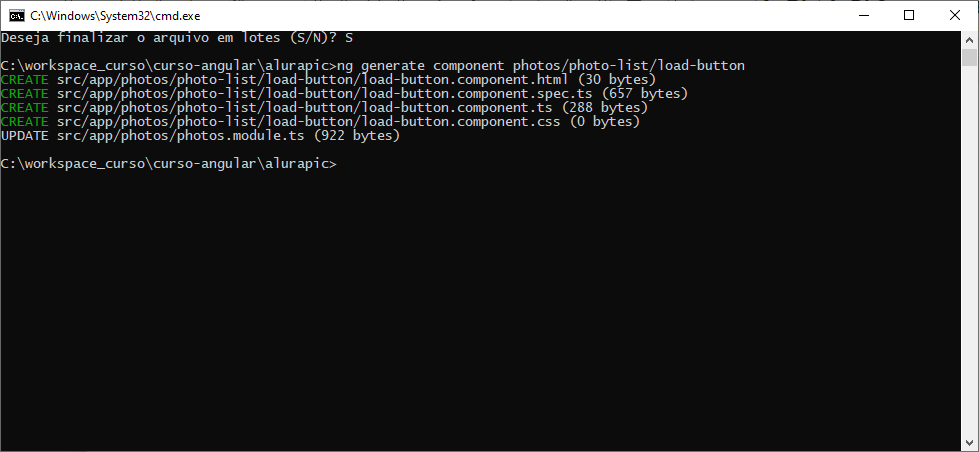
  }

Gerando o botão de load

C:\workspace\_curso\curso-angular\alurapic>ng generate component photos/photo-list/load-button

Ou pode ser abreviado

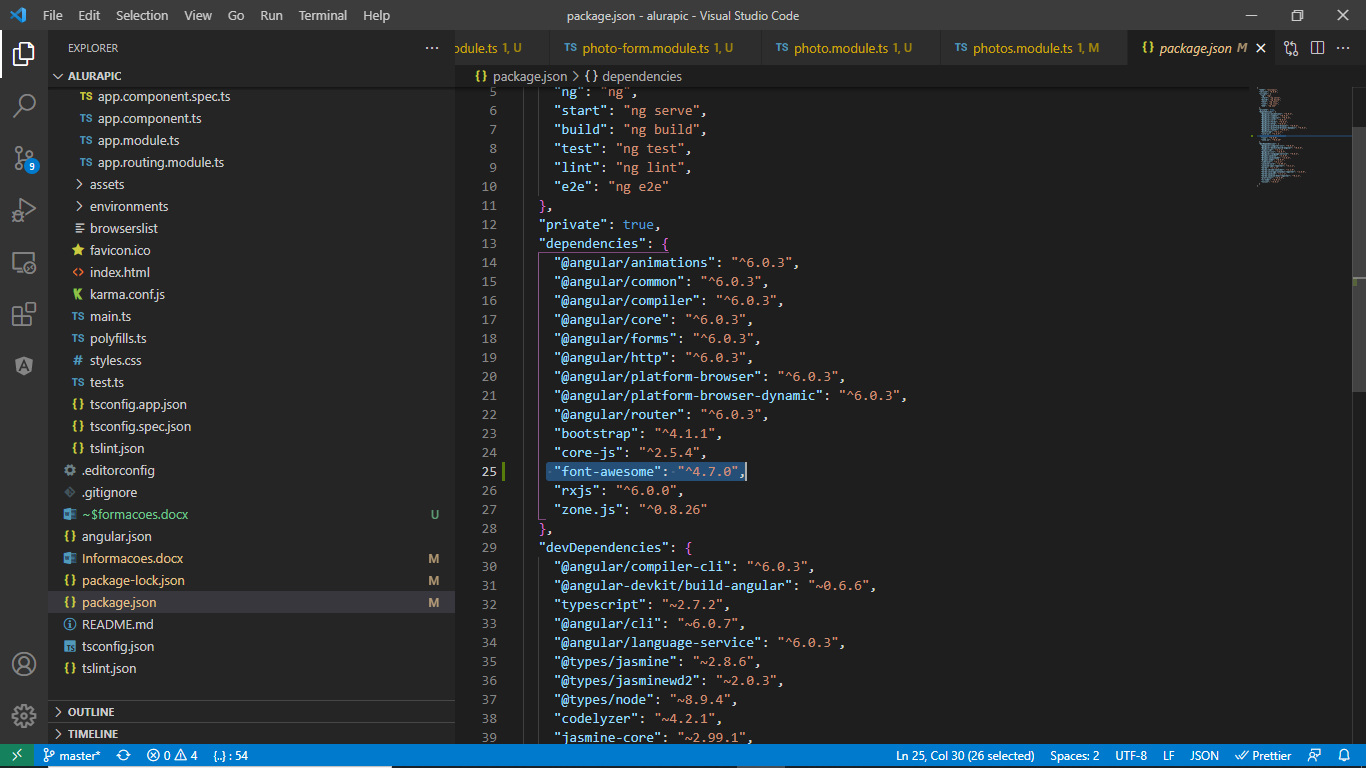
C:\workspace\_curso\curso-angular\alurapic>ng g c photos/photo-list/load-button



Instalação da biblioteca de fontes e icones

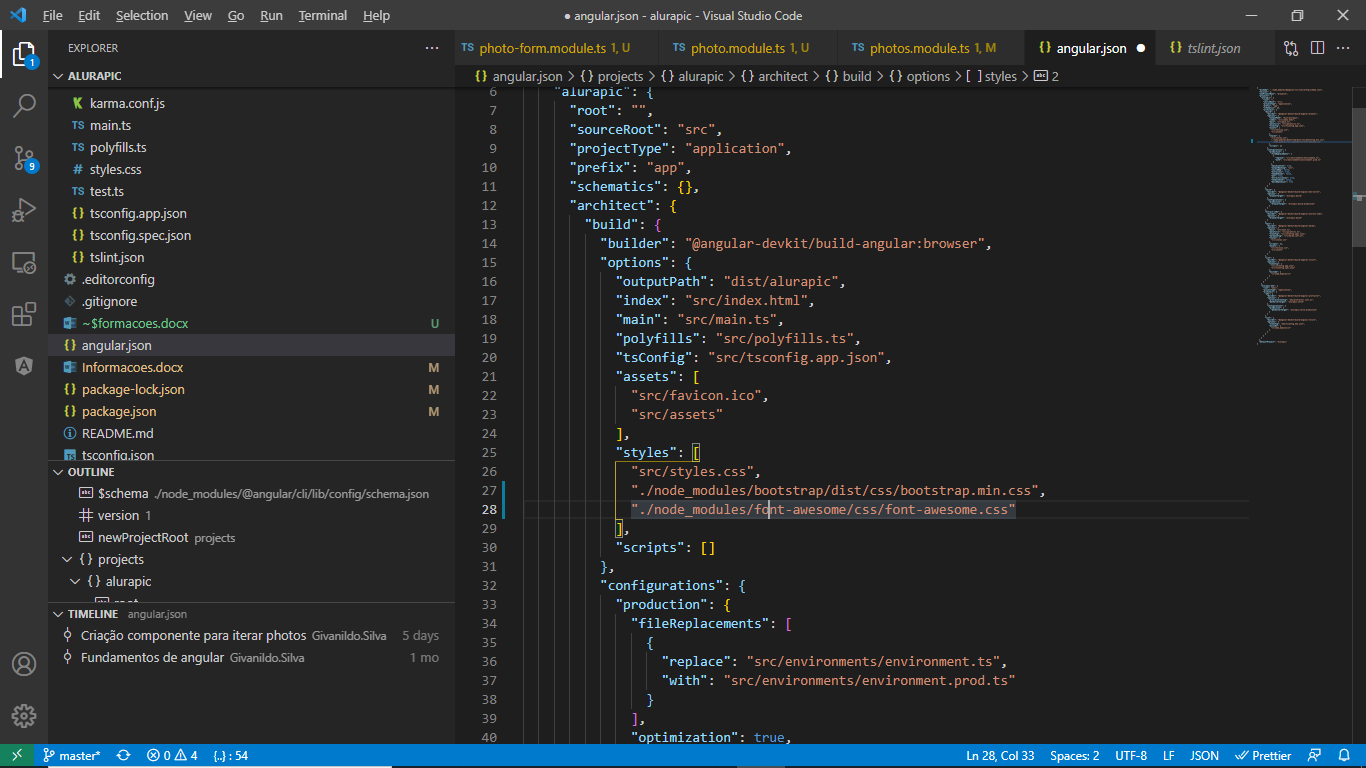
C:\workspace\_curso\curso-angular\alurapic>npm install font-awesome@4.7.0 -S

Osucesso da instalação da biblioteca pode ser verificada no package.json



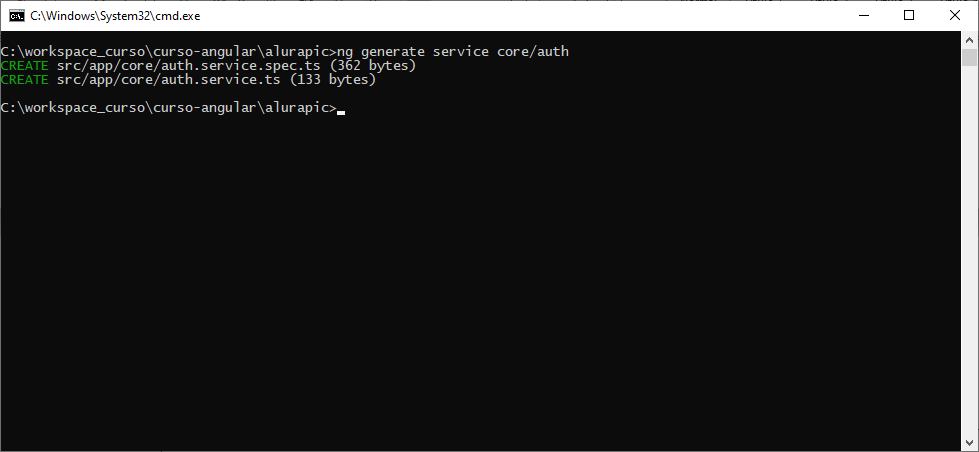
Registrar o caminho do font-awesome( caminho no node\_modules) no arquivo angular.json

 "./node\_modules/font-awesome/css/font-awesome.css"



Gerando um serviço de autenticação auth

C:\workspace\_curso\curso-angular\alurapic>ng generate service core/auth



User: flavio

Password: 123

**ANGULAR 2**

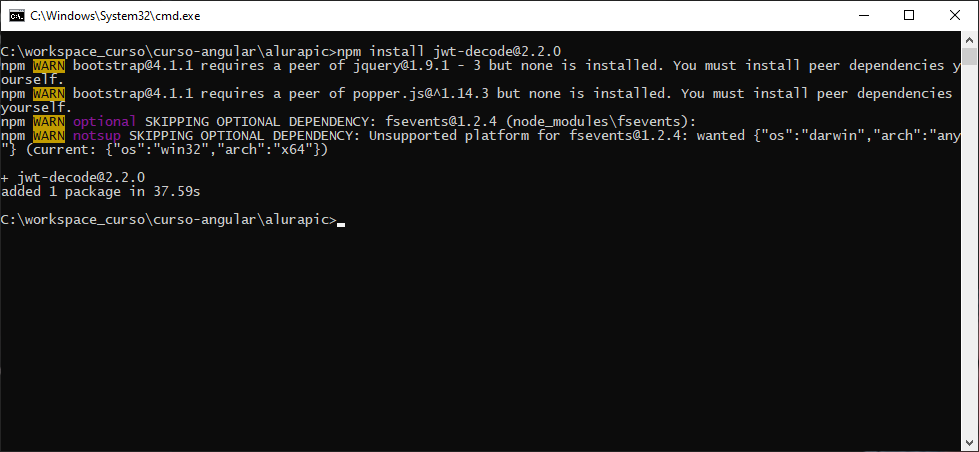
Vimos nesta aula:

* criação de um componente de login;
* validação de formulário;
* como componentizar mensagens de validação;
* como enviar crendenciais para a API;
* como redirecionar e obter referências do template (ViewChild);
* descobrindo a plataforma de execução.

Vimos nesta aula:

* como acessar o header de resposta;
* armazenamento de token;
* a segurança do token;
* cabeçalho da nossa aplicação.

Instalação do jwt



**Obs.: Com o Async pipe conseguimos capturar a emissão do Observable diretamente do nosso template.**

**Obs.: au utilizar [AuthGuard] deve-se reiniciar o angular CLI**

**O guarda de rotas serve para darmos consistência para nossa aplicação, liberando acesso apenas para as rotas que fazem sentido para nosso usuário.**

**Vimos nesta aula:**

**criação de um componente de registro;**

**validação de formulário de registro;**

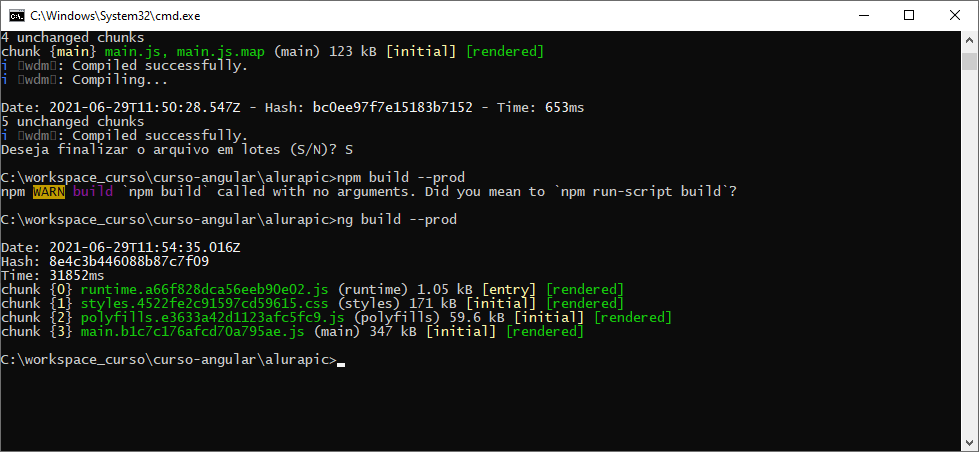
**como criar nosso próprio validador;**

**validando de maneira assíncrona;**

**submissão de dados.**

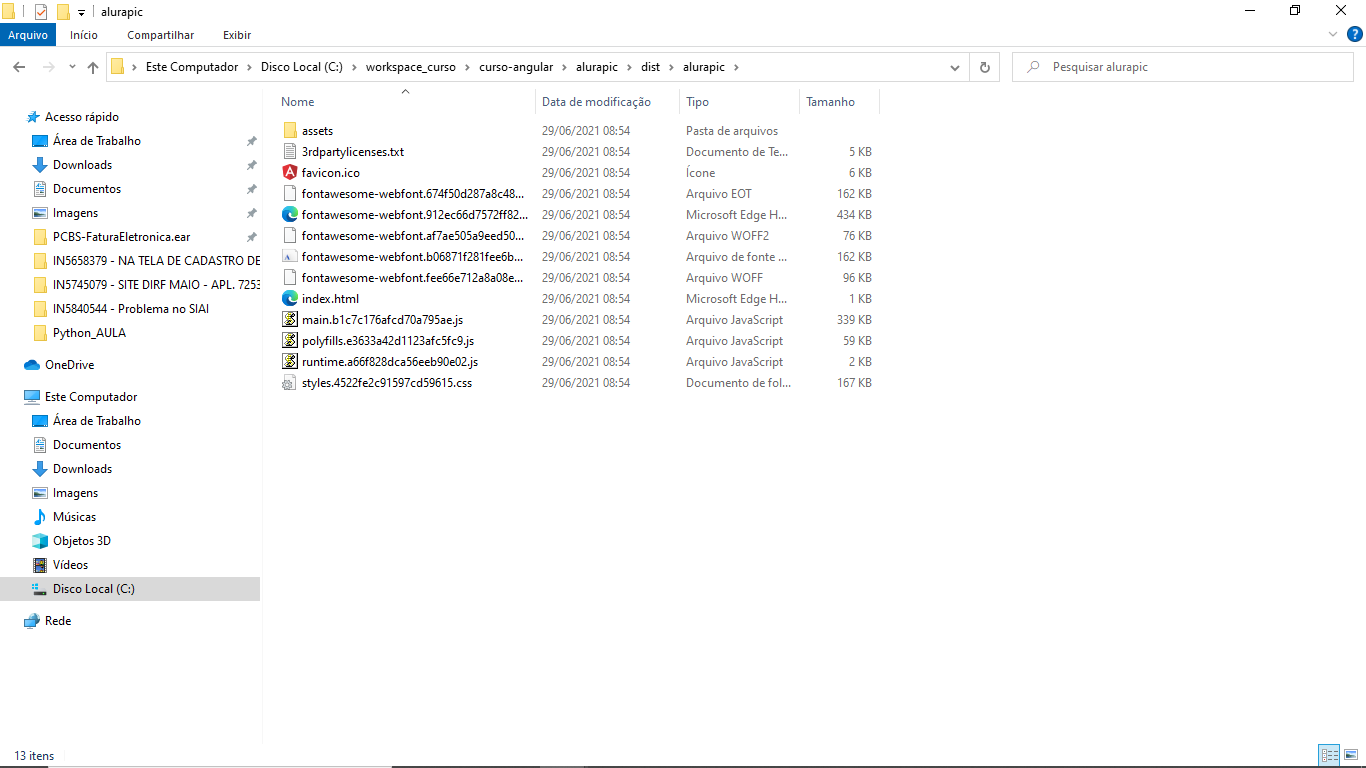
**Para build do projeto**

**C:\workspace\_curso\curso-angular\alurapic>ng build --prod**



Gera a pasta dist\alurapic ( onde estão os arquivos que ficarão no servidor)

C:\workspace\_curso\curso-angular\alurapic\dist\alurapic



Vimos nesta aula:

* rotas filhas;
* como buildar o projeto;
* Lazy loading e code splitting;
* interceptadores e envio de token.

Condtrução da aplicação

C:\workspace\_curso\curso-angular\alurapic>ng build --prod

Cria dentro dist\alurapic

