



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE



Aula 2 – Typescript, instalação e criação de uma aplicação Angular

Prof.: Bruno Emerson Gurgel Gomes
IFRN – Campus Parnamirim



Introdução

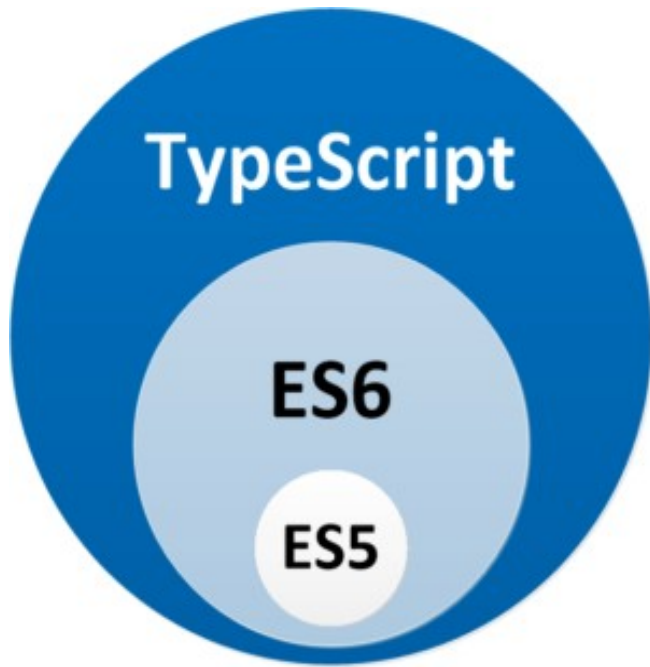
- Um conjunto de ferramentas são utilizadas para criar, rodar e distribuir a aplicação Angular
- A linguagem base do Angular é o *Typescript*
- O estudo do *framework* deve começar por entender o *Typescript* e essas ferramentas

Typescript

- **Engloba um super conjunto do javaScript**
 - Se mantém atualizado com futuras propostas para Javascript
 - Traduzido em JS compatível com ECMAScript 3 em diante
- **Trabalho mais produtivo integrado a IDE's (VS Code, VS, Atom...)**
 - Verificação estática, auto *build* automático, refatoração, etc.



TypeScript



- **ECMAScript** é uma especificação que serve de base para a linguagem JavaScript
- **TypeScript**
 - engloba o ES 5 (nov. 2009) e o ES 6 (jul. 2015)
 - incorpora inovações propostas nas ES (atualmente na versão 9)



Instalação

Duas formas:

- 1) A partir do instalador de pacotes do Node.js
- 2) Instalando o Visual Studio que já vem com o Typescript



Instalação pelo Node.js

1) Baixe e instale a versão mais recente do Node.js para a sua plataforma em <https://nodejs.org>

2) Teste a instalação no terminal (linux) ou cmd (Windows). A saída deve ser as versões instaladas dos programas

node -v

executa o node.js

npm -v

executa o gerenciador de
pacotes do node (npm)

3) Execute o comando no terminal (linux) ou command prompt (Windows):

npm install -g typescript

Opção **-g** instala o pacote de forma global, assim fica disponível para todos os projetos



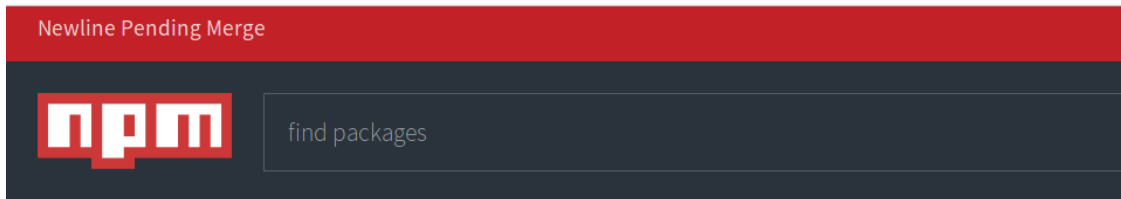
NPM <https://www.npmjs.com/>

- O Node Package Manager é o repositório de pacotes do Node.js
 - instalação de diversas ferramentas e bibliotecas
- Não é necessário trabalhar com o Node.js para usar o npm
- Utilizamos para instalar:
 - Typescript, o Angular, bootstrap e diversos outros pacotes



NPM

Parte da página do npm após a busca pelo pacote *typescript*



★ **typescript** public
build passing npm package 2.7.2 downloads 10M/month

Join the chat at <https://gitter.im/Microsoft/TypeScript>

TypeScript is a language for application-scale JavaScript. TypeScript adds optional types, classes, and modules to JavaScript. TypeScript supports tools for large-scale JavaScript applications for any browser, for any host, on any OS. TypeScript compiles to readable, standards-based JavaScript. Try it out at the [playground](#), and stay up to date via [our blog](#) and [Twitter account](#).

Installing

For the latest stable version:

```
npm install -g typescript
```



Exemplo

```
let nome: string = 'Bruno Gomes'

function ola(nm: string): string {
    return 'Seja bem vindo ' + nm + '!'
}

console.log(ola(nome))
```

- 1) Abra o seu editor de texto e salve o código com algum nome e extensão .ts (exemplo, *ola.ts*)
- 2) Execute na linha de comando o compilador *typescript* (tsc). A saída deve ser um arquivo .js (ola.js).
- 3) Rode o programa ola.js gerado com o Node.js (node) na linha de comando ou insira em uma página HTML e execute com o seu navegador Web.

```
tsc ola.ts
```

```
node ola.js
```



@angular/cli

- **Deve ser instalada após o Typescript**
- **Ferramentas para desenvolvimento da aplicação angular**
 - criação, execução, testes e construção
- **Instalação**

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

@angular/cli - comandos básicos

Comando	significado/função
ng new <i>nomeapp</i>	Cria uma nova aplicação. É cria uma pasta <i>nomeapp</i> contendo todo o código básico de uma aplicação Angular.
ng serve	Constrói a aplicação e cria um domínio temporário (http://localhost:4200) para que o desenvolvedor possa rodar e testar. Sempre que a aplicação é modificada a página é alterada automaticamente.
ng g c <i>nomecomp</i>	Cria uma pasta <i>nomecomp</i> com os arquivos básicos para um novo componente.
ng g s <i>nomeserv</i>	Cria um arquivo <i>nomeserv.ts</i> com esqueleto de código para criação de um novo serviço.
ng build	Compila e constrói a aplicação para que ela seja distribuída. É criada uma pasta dist com os arquivos da aplicação. Opção --prod (versão final de “produção”) Opção --dev (versão de desenvolvimento)

Primeira aplicação (teste) !

Os comandos a seguir devem ser feitos em linha de comando (*cmd*, *shell*, etc.)

Crie a aplicação

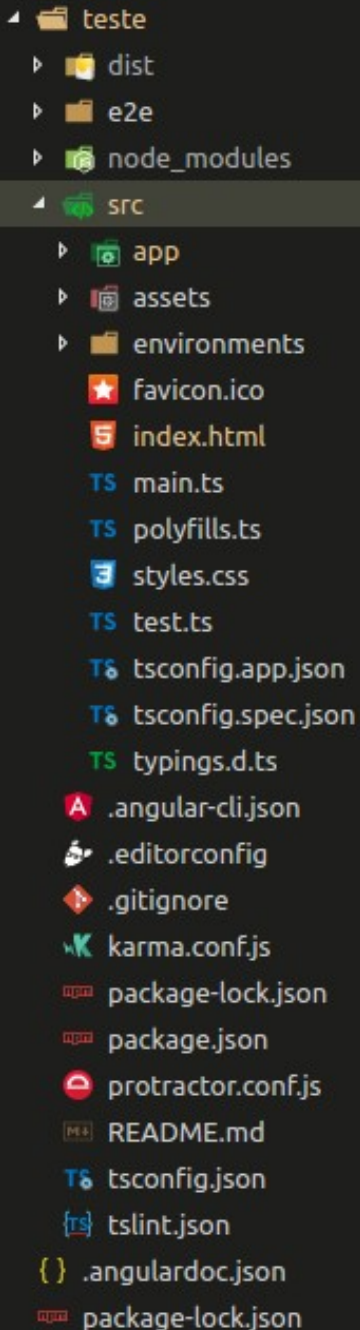
ng new teste

Mude para o diretório da aplicação e rode o servidor de teste

ng serve

Acesse a URL **<http://localhost:4200>** para ver a aplicação

Estrutura básica gerada!!

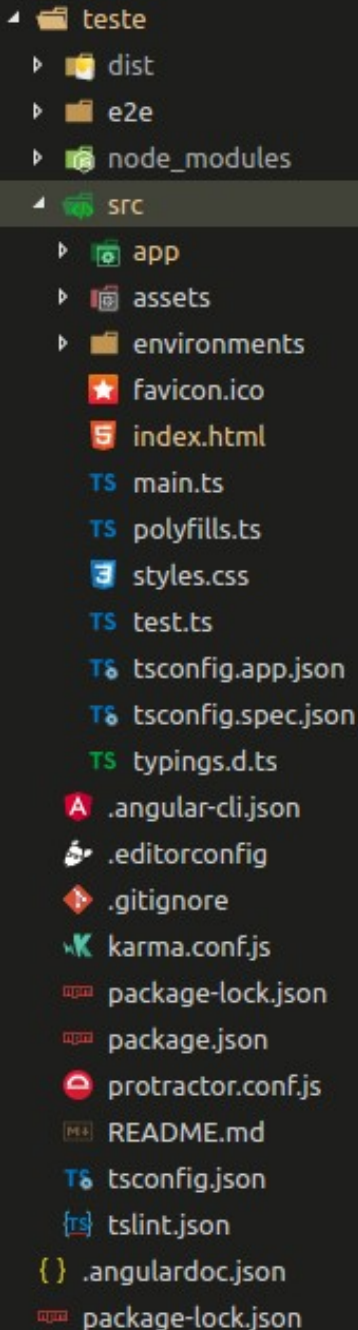


A screenshot of a file explorer showing the project structure. The 'src' folder is expanded, revealing a list of files and subfolders. The files include 'app', 'assets', 'environments', 'favicon.ico', 'index.html', 'main.ts', 'polyfills.ts', 'styles.css', 'test.ts', 'tsconfig.app.json', 'tsconfig.spec.json', 'typings.d.ts', '.angular-cli.json', '.editorconfig', '.gitignore', 'karma.conf.js', 'package-lock.json', 'package.json', 'protractor.conf.js', 'README.md', 'tsconfig.json', 'tslint.json', '.angulardoc.json', and another 'package-lock.json' at the bottom.

- teste
 - dist
 - e2e
 - node_modules
 - src
 - app
 - assets
 - environments
 - favicon.ico
 - index.html
 - main.ts
 - polyfills.ts
 - styles.css
 - test.ts
 - tsconfig.app.json
 - tsconfig.spec.json
 - typings.d.ts
 - .angular-cli.json
 - .editorconfig
 - .gitignore
 - karma.conf.js
 - package-lock.json
 - package.json
 - protractor.conf.js
 - README.md
 - tsconfig.json
 - tslint.json
 - .angulardoc.json
 - package-lock.json

- **Diversos arquivos e pastas**
 - Configuração,
 - Módulo e componentes,
 - Organização dos pacotes, classes e APIs necessárias,
 - Código fonte, etc.

Estrutura da aplicação



```
teste
├── dist
├── e2e
├── node_modules
└── src
    ├── app
    ├── assets
    ├── environments
    ├── favicon.ico
    ├── index.html
    ├── main.ts
    ├── polyfills.ts
    ├── styles.css
    ├── test.ts
    ├── tsconfig.app.json
    ├── tsconfig.spec.json
    ├── typings.d.ts
    ├── .angular-cli.json
    ├── .editorconfig
    ├── .gitignore
    ├── karma.conf.js
    ├── package-lock.json
    ├── package.json
    ├── protractor.conf.js
    ├── README.md
    ├── tsconfig.json
    ├── tslint.json
    ├── .angulardoc.json
    └── package-lock.json
```

- **Pasta *src***

- Contém pastas e arquivos de desenvolvimento
- Módulo e componente AppComponent (<app-root>)
- Todo o código Typescript (componentes, classes, etc.) que o desenvolvedor criar
- Códigos html e css de componentes, etc.

- **index.html**

- Página raiz. Normalmente não é necessário editar.
- Código HTML será inserido nos templates (.html) dos componentes.

- **styles.css**

- Estilos 'globais' da aplicação

Build e execução da aplicação

Passos necessários para rodar a versão de distribuição!

Faça o build da aplicação

```
ng build --prod
```

Instale o pacote http-server

```
npm install -g http-server
```

Vá ao diretório dist e digite no terminal

```
http-server
```

Abra o navegador e digite a URL

```
http://localhost:8080
```

Referências

ANGULAR. Angular. Disponível em:
<https://angular.io>. Acesso em: 01 mar. 2018.

ANGULAR CLI. Angular CLI. Acesso em: 01 mar. 2018.
Disponível em:
<https://github.com/angular/angular-cli/wiki>

GUEDES, Tiago. Crie aplicações com Angular: o novo framework do Google. Casa do Código, 2017.