

Primeira aplicação



Usamos o angular-cli para criar a estrutura inicial de uma aplicação em angular.

```
ng new <nome-aplicação> --prefix=<pre>
```

Para testar se a estrutura criada está funcionando podemos iniciar o servidor de desenvolvimento do angular.

```
ng serve
```

localhost:4200

Estrutura criada



package.json

Arquivo de configuração onde todas as dependências são listadas. Ele contém uma seção com os scripts que podem ser usados na aplicação. Também podemos ver as dependências principais do angular. Logo após podemos identificar as dependências que são usadas apenas em desenvolvimento, elas não são utilizadas em runtime.

angular.json

Neste arquivo são definidos quais arquivos de script são usados na aplicação, arquivos de estilo, os assets principais, o arquivo de index, o arquivo principal (main.ts)

webpack

É uma biblioteca responsável por criar bundles. Ele separa a aplicação em arquivos .js, ele empacota a aplicação.

styles.css

Definição do estilo principal da aplicação.

polyfills.ts

Arquivo utilizado para adicionar compatibilidade a browsers antigos.

Como o angular funciona



Quando o comando `npm start` (ou `ng serve`) é executado o webpack começa uma análise da aplicação, começando pelo `main.ts`, procurando por todos os imports, por todas as referências e os separa nos bundles.

Quando fazemos a requisição do `index.html` os bundles são injetados dinamicamente e adicionados no final da página.

O browser então recebe o `index.html` com os bundles injetados e inicia a execução da aplicação.

O main bundle carrega então o primeiro componente, o `main.ts`

```
platformBrowserDynamic().bootstrapModule(AppModule);
```



```
import { AppModule } from './app/app.module';
```

`app.module.ts`

Módulos Angular



Responsável por agrupar uma série de componentes da aplicação ou do próprio framework.

@NgModule - é um decorator. Uma função que serve para aplicar metadados em uma classe, um atributo, um método ou em um argumento de um método.

Adiciona declarations (todos os componentes), imports (quais são as dependências), providers e bootstrap (qual é o componente na lista de declarations que é responsável por iniciar a aplicação).

app.component.ts

Ele também é uma classe e importa outro decorator. Esse decorator é específico para componentes. Os atributos obrigatórios são apenas **selector** e **template**.

Módulos Angular



Dependências básicas para trabalhar no browser

```
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';
import { FormsModule } from '@angular/forms';
import { HttpClientModule } from '@angular/http';
```

Decorator. Função para aplicar metadados em uma classe, atributo ou método.

Dependências para trabalhar com formulários e chamadas Http

```
import { AppComponent } from './app.component';
```

```
@NgModule({
  declarations: [
    AppComponent
  ],
  imports: [
    BrowserModule,
    FormsModule,
    HttpClientModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
```

```
export class AppModule { }
```

Declarations - lista todos os componentes da aplicação

Imports - lista as dependências da aplicação

Bootstrap - indica qual o componente na lista dos que estão em declarations é o principal da aplicação

**Toda aplicação Angular terá pelo menos 1
módulo + 1 componente**

Módulos Angular



No body do arquivo `index.html` há um elemento com o selector do componente. Esse componente será então carregado na página `index.html`