

PROGRAMAÇÃO WEBII

1

Módulos

O que são? Exemplo.

O que são Módulos?



ngModule

<i>declarations</i> [...]	<i>imports</i> [...]
<i>providers</i> [...]	<i>exports</i> [...]
<i>bootstrap</i> [mainComponent]	

O que são?

- O Angular é um framework modular, ele precisa de um ou mais módulos.
- O Módulo é uma classe marcada com o `decorator@NgModule()`.
- Tem como objetivo declarar e agrupar tudo que criamos no Angular.
- Os módulos são uma ótima maneira de organizar um aplicativo e ampliá-lo com recursos de bibliotecas externas.
- **Como gerar novos módulos:** <https://angular.io/cli/generate#module-command>

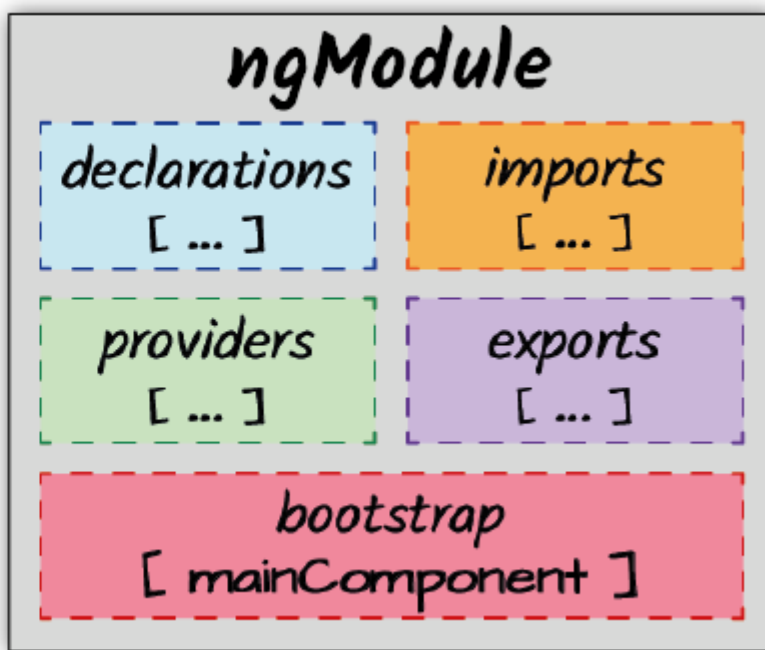
```
ng generate module <name> [options]
```

Estrutura padrão

src/app/app.module.ts

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

@NgModule({
  imports:      [ BrowserModule ],
  providers:    [ Logger ],
  declarations: [ AppComponent ],
  exports:      [ AppComponent ],
  bootstrap:    [ AppComponent ]
})
export class AppModule { }
```



Link com os dados da API do `NgModule`:

<https://angular.io/guide/ngmodule-api>

bootstrap

- Propriedade que define quais componentes são responsáveis por iniciar a aplicação
- Só existe no Módulo principal
- Já vem preenchido com o componente inicial ao criar uma aplicação com o Angular CLI
- O padrão é AppComponent porém, este nome pode mudar caso mude o prefixo ao criar o projeto Opção prefix

ngModule

declarations

[...]

imports

[...]

providers

[...]

exports

[...]

bootstrap

[*mainComponent*]

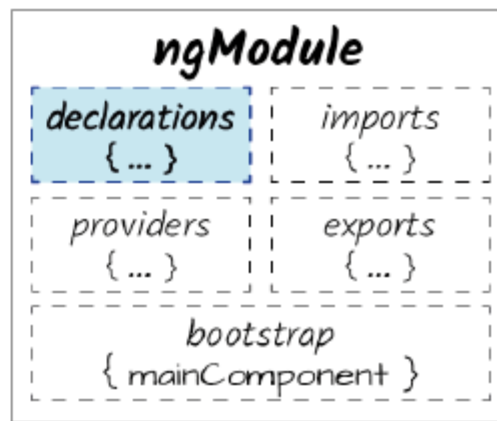
declarations

- Propriedade que define quais Componentes, Diretivas e Pipes, pertencem ao módulo.

```
declarations: [  
  HeaderComponent, // Adiciona um componente de Header ao App  
  BlueTextDirective, // Adiciona uma diretiva para deixar o texto Azul  
  CreditCardPipe // Adiciona um Pipe que formatar o numero do cartão de crédito  
],
```

OBSERVAÇÃO:

Quando a quantidade de declarações se torna muito grande, é comum a criação de novos módulos para agrupar os itens com funcionalidades em comum.



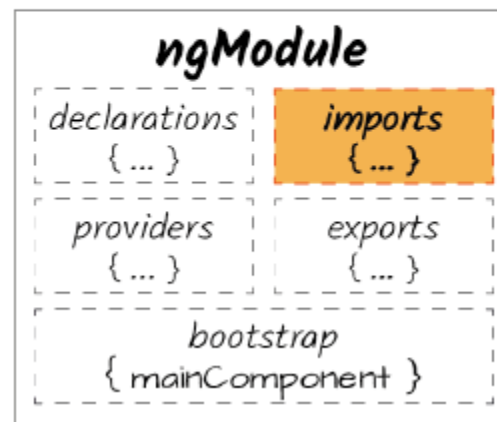
imports

- Propriedade que define quais módulos a sua aplicação precisa importar para ter acesso aos Componentes, Diretivas e Pipes que foram exportados.
- Geralmente utilizado para importar bibliotecas de terceiros.

```
imports: [  
  HttpClientModule, // Módulo para tratar requisições HTTP.  
  FinanceiroModule, // Adiciona os componentes da área financeira.  
],
```

MÓDULOS OBRIGATÓRIOS:

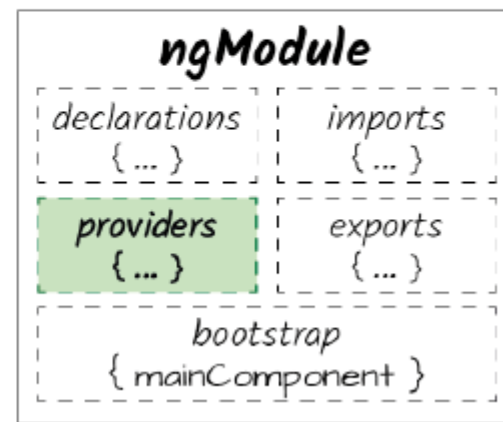
- ✓ **BrowserModule:** Deve ser importado no módulo principal da aplicação.
- ✓ **CommonModule:** Deve ser importado nos demais módulos.



providers

- Propriedade que define os serviços que serão utilizados pelos componentes.
- Os serviços são o elo de ligação entre a aplicação e os dados.

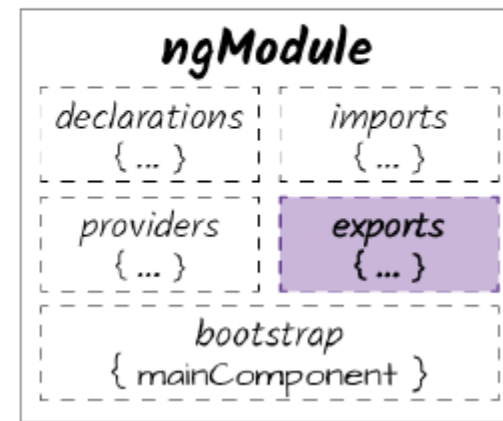
```
providers: [  
  ClientesService, // Serviço responsável por manipular clientes com o BD.  
  DadosService, // Serviço para armazenamento temporário de dados.  
],
```



exports

- Propriedade que define quais itens (componentes, diretivas e pipes) serão visíveis por outros módulos quando este for importado.
- Geralmente só é declarado quando o módulo foi criado para ser utilizado em outros.

```
exports: [  
  ModalComponent, // Exporta o componente para ser utilizado em outros módulos.  
  FotosDirective, // Exporta uma diretiva de manipulação de fotos.  
  CepPipe // Exporta um pipe de formatação de CEP.  
],
```



Exemplo

```
@NgModule({
  declarations: [
    AppComponent,
    HeaderComponent,
    FooterComponent,
    HomeComponent,
    ListarProdutosComponent,
    CadastrarProdutoComponent,
    AtualizarProdutoComponent
  ],
  imports: [
    BrowserModule,
    AppRoutingModule,
    FormsModule,
    SharedModule,
    HttpClientModule,
    BrowserAnimationsModule, // required animations module
    ToastrModule.forRoot(), // ToastrModule added
    SweetAlert2Module.forRoot(),
  ],
  providers: [
    { provide: LOCALE_ID, useValue: 'pt-BR' },
  ],
  bootstrap: [AppComponent]
```

exemplo de modularização

