

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



# Introdução ao Framework Angular

Prof. Bruno Emerson Gurgel Gomes  
IFRN – Campus Parnamirim



# Angular

- **Framework front-end**
  - Extensível por bibliotecas e ferramentas
  - **@angular/cli** para criação/*build* da aplicação
- **Projeto OpenSource (licença MIT) mantido pelo Google**
  - Site oficial: <https://angular.io>
  - Código fonte: <https://github.com/angular>
- **Desenvolvimento multi-plataforma**
  - Web, Mobile (Ionic) e Desktop

# Angular – *Single Page Applications (SPA)*

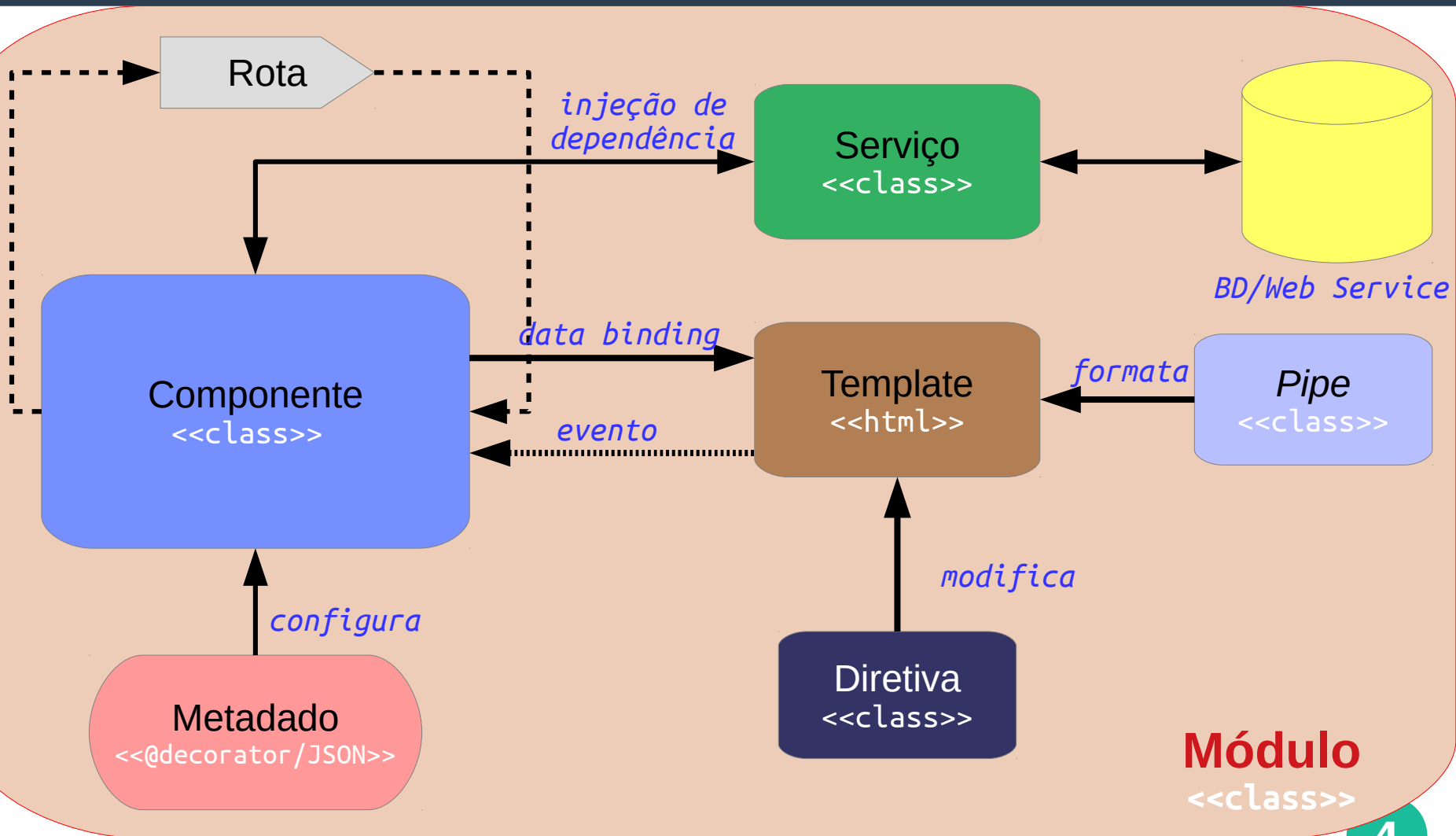
- **Características principais**

- Renderiza página em resposta a ações no cliente
- Requisição ao servidor para buscar informações (normalmente API REST/JSON)

- **Resulta em páginas mais leves**

- Carregamento de partes da aplicação a depender da ação do usuário
- Servidor não precisa montar o HTML/CSS

# Visão Geral - elementos principais



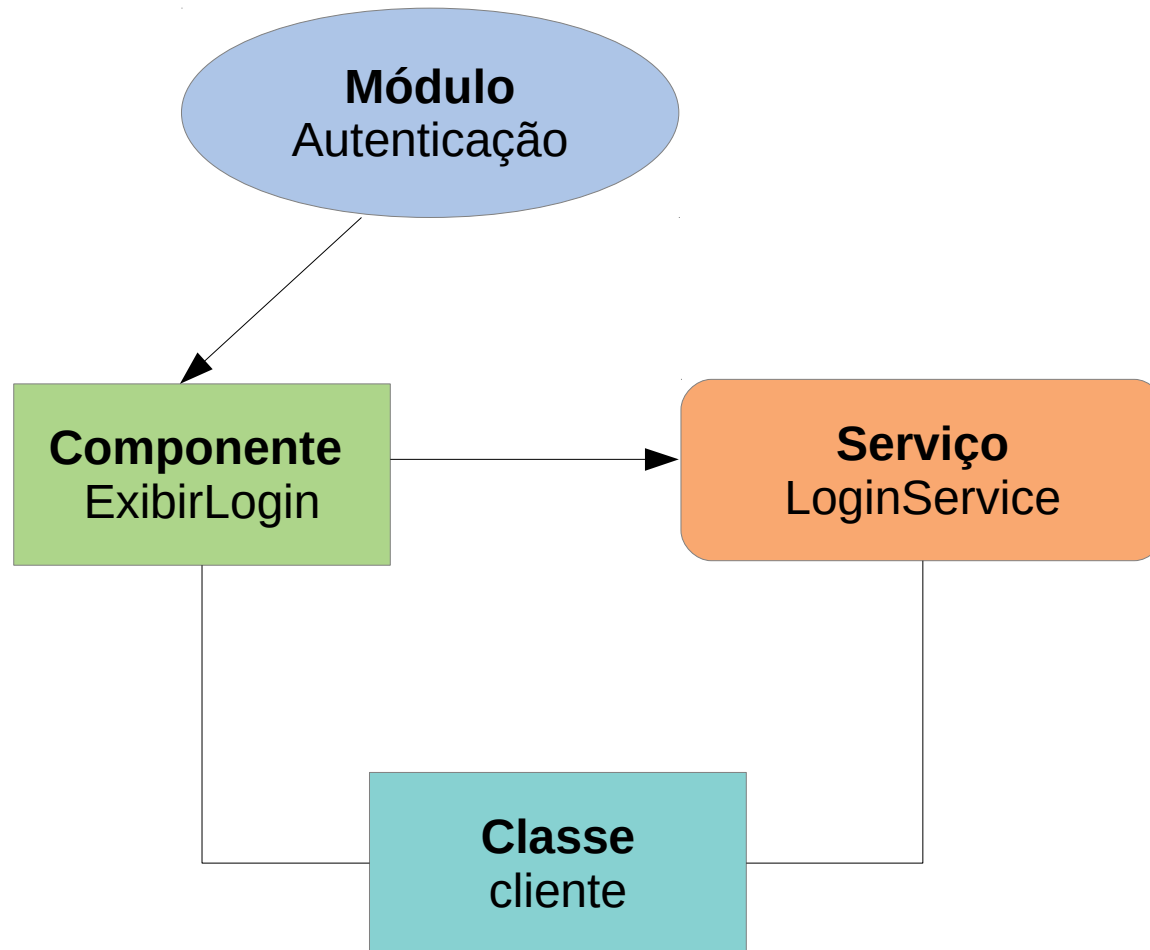
# Para facilitar: App de Loja Virtual

- **Composta por vários “subsistemas”**
  - Autenticação
  - Cadastro de clientes
  - Venda
  - Estoque, ...
- **Como esses subsistemas seriam organizados no Angular?**

# Exemplo Loja Virtual - autenticação

- No Angular, cada subsistema é um Módulo
- As “páginas dinâmicas” dentro de cada módulo são Componentes
  - Página de login
- As informações necessárias são extraídas de classes de Serviço
  - Verificar se cliente existe,
  - autorizar/negar acesso
  - Inserir id do cliente na sessão

# Exemplo Loja Virtual



# Módulo (@NgModule)

- **Organiza a aplicação**
  - A ideia é unir componentes, serviços e classes que se relacionam
- **Indica o componente principal**
- **Importa outros módulos**
- **Declara componentes, diretivas e pipes**
- **“Exporta” serviços**
- **Toda aplicação tem ao menos um módulo (AppModule)**
  - Inicia a aplicação (AppComponent)
- **Módulo principal é suficiente em aplicações de menor porte**

```
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';
import { AppComponent } from './app.component';
import { CursosComponent } from './cursos/cursos.component';
import { CursosService } from './cursos/cursos.service';

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



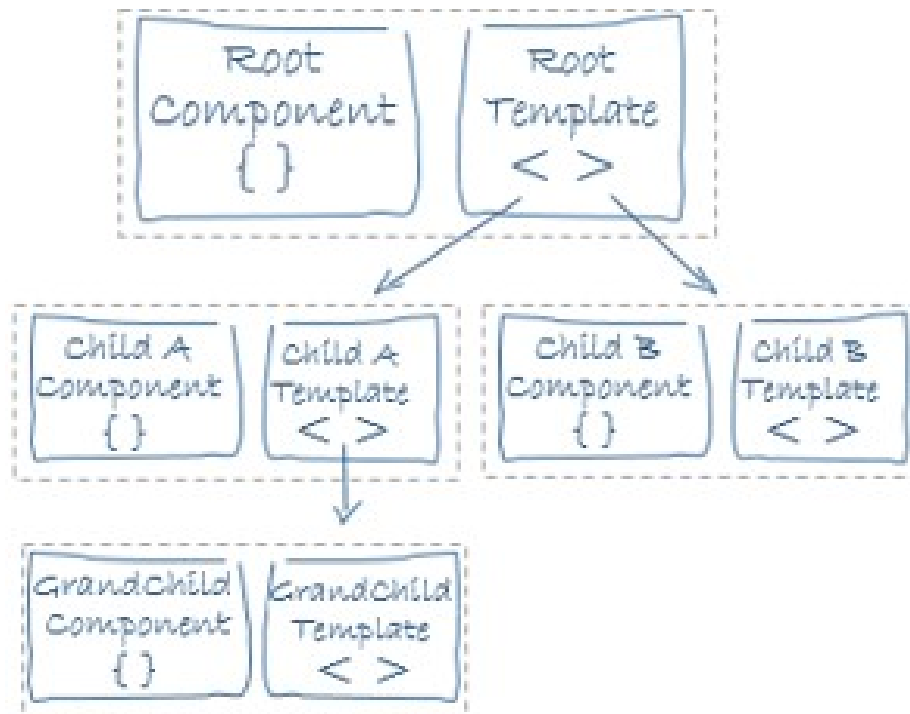
Module  
Component  
{ }

Um **componente** é um pedaço da aplicação contido em um **módulo**

# Componente (@Component)

- **Aplicação em Angular é composta por um conjunto de componentes**
- **Componente encapsula uma funcionalidade específica**
  - Formulário de cadastro
  - Menu de navegação
  - Botão de “like”
  - Listagem, ...

# App Angular - conjunto de componentes



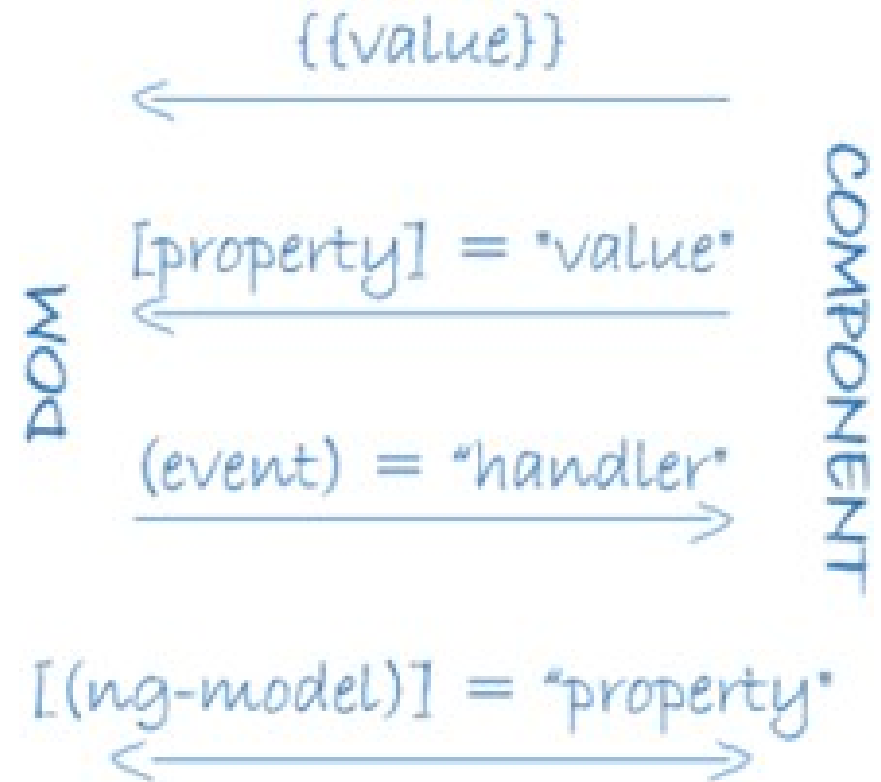
# Componente (@Component)

- **Classe Typescript**
- **Controla o conteúdo e comportamento do *Template***
  - HTML/CSS + diretivas
- **Configuração por meio de metadados**
- **Lógica de negócio implementada em Serviços**

```
@Component({
  selector: 'app-ifrn-cursos',
  templateUrl: './cursos.component.html',
  styleUrls: ['./cursos.component.css']})
export class CursosComponent implements
OnInit {
  private cursos: string[];
  constructor(private cursosService: CursosService {
    this.cursos = [];
  }
  ngOnInit() { this.cursos =
    this.cursosService.getCursos();
  }
  onAddCurso(cc: string) {
    this.cursosService.addCurso(cc);
  }
}
```

# Componente ↔ Template

- **Data binding:** Informações do componente para o template
- **Eventos:** Ações e dados do *template* para o componente



# Metadados

- **Configura o processamento das classes**
  - Módulos, componentes e serviços
- **Uso de *decorators* (@)**
  - Metadados nos *decorators* são objetos JSON

```
@Component({  
  selector: 'app-ifrn-cursos',  
  templateUrl: './cursos.component.html',  
  styleUrls: ['./cursos.component.css']})
```

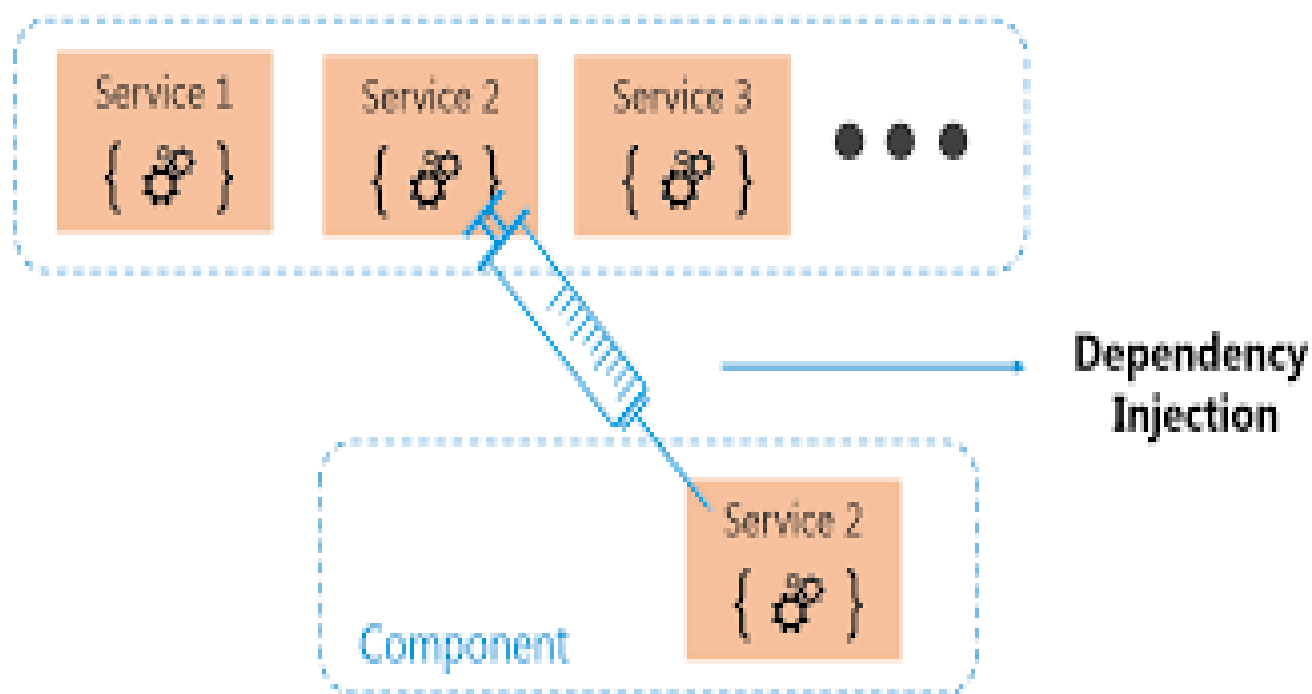
# Serviços (@Injectable)

- Implementa a lógica de negócio do componente
- Faz a ligação entre webservices REST ou bases de dados
- É criado no componente por meio de *injeção de dependência* em seu construtor

```
import { Injectable } from '@angular/core';
@Injectable()
export class CursosService {
  private cursos: string[] = ['Angular', 'Java', 'MySQL'];
  constructor() { console.log('CursosService'); }
  getCursos() { return this.cursos; }
  addCurso(cc: string) { this.cursos.push(cc); }
}
```

# Componentes e serviços

edureka!





# Referências

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

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

**GRONER, Loiane.** Curso de Angular 2. Disponível em: <http://https://loiane.training/>. Acesso em 01 mar. 2018.