

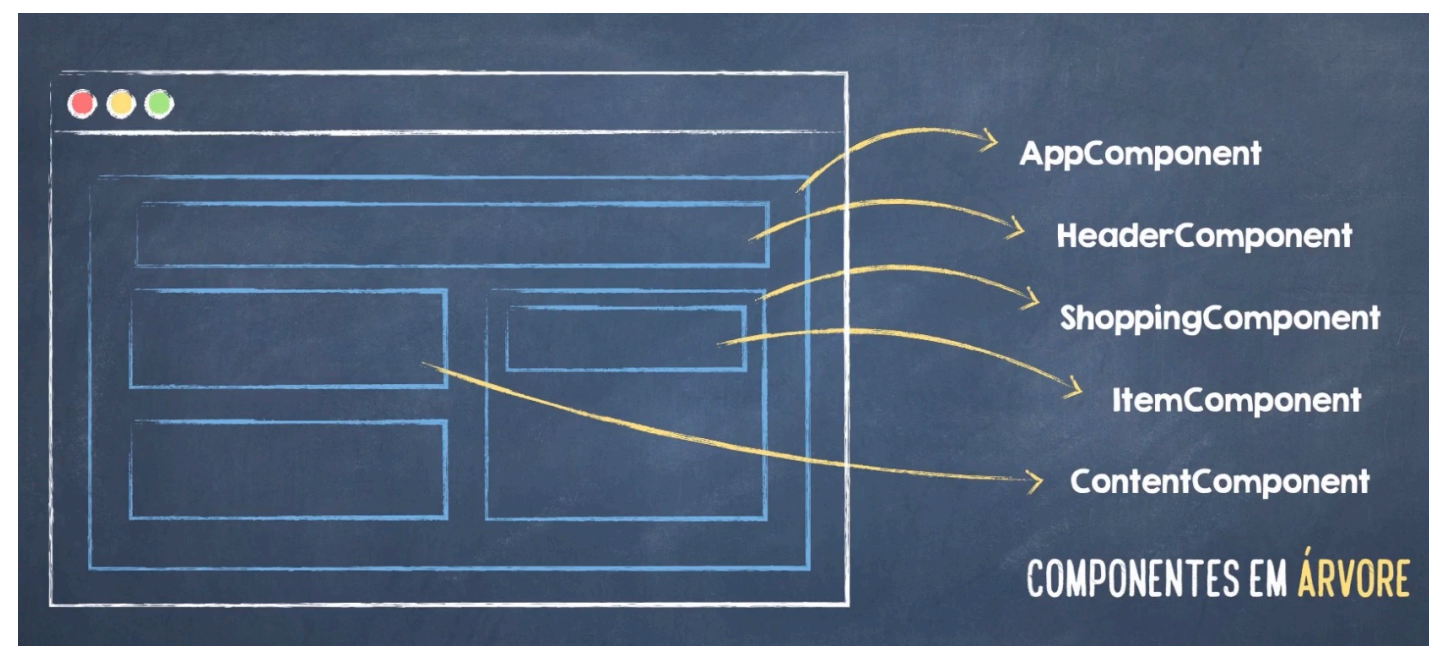
# Componentes



**Componentes são partes independentes e reutilizáveis. Em angular um componente é uma classe que tem um ciclo de vida, possui um template para determinar sua aparência e um selector (tag) para ser usado por outras partes da aplicação.**

**No angular uma página é composta por partes menores que podem funcionar de forma independente ou se comunicarem entre si através de eventos.**

**O framework está estruturado de forma que há sempre um componente raiz (pai) seguido pelos seus filhos.**



# Componentes

## Estrutura



```
export class MeuComponente {  
  constructor () { }  
  
}
```

# Componentes

## Estrutura



```
import { Component } from '@angular/core'
```

```
@Component({  
  selector: 'app-componente',  
  templateUrl: './meucomponente.html' } )
```

—————→ Decorator

```
export class MeuComponente {  
  constructor () { }  
  
}
```

Uma vez que o componente foi criado é necessário informar em que módulo o componente vai estar.



# Componentes

Criação com o angular-cli

**ng generate component header --skip-tests**

O generate automaticamente adiciona o novo componente no módulo principal da aplicação.

No fonte do componente podemos identificar o método `ngOnInit()`. Ele é um método que sempre é iniciado durante o ciclo de vida do componente.



# Componentes

Criação com o angular-cli

**1 - podemos alterar o conteúdo do componente de cabeçalho adicionando o que precisarmos. Para isso vamos usar HTML + CSS.**

**2 - em seguida vamos usar esse componente a partir do componente principal da aplicação. Para isso basta usar a tag do componente que criamos <tag></tag>**

**3 - também é possível adicionarmos código css ao nosso novo componente através do seu arquivo de css.**



# Property bind

É uma ligação entre uma propriedade de um elemento a uma expressão angular definida em um componente ou em um método.

Sintaxe: [ ] ao redor da propriedade do elemento que você quer atribuir o valor.

```
// no componente  
  
aluno = {name: 'João da Silva'}
```

```
<!-- no template do componente -->  
  
<input type="text" [value]="aluno.name">
```

**Sempre que o valor do aluno (no componente) mudar, o valor no input também vai mudar, mostrando que eles estão ligados!**



# Property bind

## Mais um exemplo

```
// no componente
```

```
aluno = {name: 'João da Silva', isAprovado: true}
```

```
<!-- no template do componente -->
```

```
<input type="text" [value]="aluno.name">
```

```
<div [hidden]="!aluno.isAprovado">
```

```
  Você foi APROVADO
```

```
</div>
```

```
<div [class.light]="aluno.isAprovado"></div>
```



```
<div class="light"></div>
```