

Ciclos de Vida (Lifecycle Hooks)

Todo componente no Angular tem um conjunto de eventos de ciclo de vida (lifecycle hooks) que ocorrem quando um componente é criado, renderizado, tem o valor de suas propriedades alteradas ou é destruído. O Angular invoca uma série de métodos (hooks), que são executados no momento em que esses eventos são acionados. Uma aplicação de loja com um carrinho de compras por exemplo, poderia utilizar os lifecycle hooks para detectar mudanças, como adição ou remoção de produtos do carrinho, e atualizar esses dados na tela.

ngOnChanges

Este é o primeiro hook do ciclo de vida que é executado. Ele monitora alterações nos Inputs do seu componente. Ele é executado uma vez na criação do componente e **sempre** que houver alterações nos valores dos Inputs, informando qual o valor antigo, qual o valor atual e se foi a primeira vez que aquela propriedade foi alterada.

```
import { Component, Input, OnChanges, SimpleChanges } from '@angular/core';

@Component({
  selector: 'my-comp',
  templateUrl: './my-comp.component.html',
  styleUrls: ['./my-comp.component.css']
})
export class MyCompComponent implements OnChanges {

  @Input()
  prop1!: number

  ngOnChanges(changes: SimpleChanges): void {
    console.log(`Valor anterior de prop1 ${changes['prop1'].previousValue}`)
    console.log(`Valor atual de prop1 ${changes['prop1'].currentValue}`)
    console.log(`É a primeira alteração do valor de prop1 ${changes['prop1'].firstChange}`)
  }
}
```

Como pode notar, implementamos uma interface chamada *OnChanges*. Essa interface nos obriga a implementar o método do primeiro ciclo de vida executado, o método *ngOnChanges*. Esse método não possui retorno e recebe um parâmetro do tipo *SimpleChanges*. Ele vai ser executado uma vez na inicialização do componente e toda vez que o valor dos inputs do seu componente for alterado. Para acessarmos os valores que mudaram, nós chamamos o parâmetro *changes* e, entre colchetes e aspas informamos o nome da propriedade. Após isso, podemos acessar as propriedades *previousValue*, que informa o valor anterior, *currentValue*, que informa o valor atual do input e a propriedade *firstChange*, que informa se foi a primeira alteração de valor do input.