**HOOKS ANGULAR**

* Los Hooks son ciclos de vida dentro de un component.
* Se ejecutarán cuando se inicie o se modifique un componente, por ejemplo.

El primer Hook sería ngOnInit(), que es el primer método que se ejecuta después de cargar el constructor.

Para poder utilizarlo, debemos implementar **OnInit** en la clase.

El **constructor** se utiliza para cargar las propiedades de la clase con valores.

El método **ngOnInit()** se utilizará para crear funcionalidad en el componente como buena praxis.

Vamos a crear un component llamado **ejemplohook.component.ts** dentro de una carpeta llamada **ejemplohook.**

**EJEMPLOHOOK.COMPONENT.TS**

import { Component, OnInit } from "@angular/core";

@Component({

  selector: "app-ejemplo-hook",

  templateUrl: "./ejemplohook.component.html"

})

export class EjemploHookComponent implements OnInit {

  constructor() {

    console.log("Constructor, yo voy primero");

  }

  ngOnInit(): void {

    console.log("Ejecutando después del constructor");

  }

}

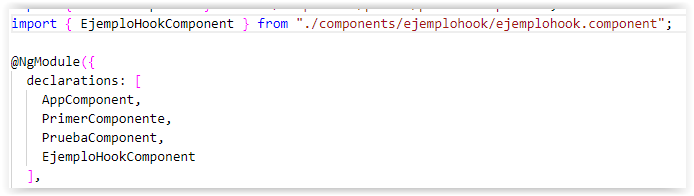
También nos creamos un nuevo archivo html llamado **ejemplohook.component.html**

**EJEMPLOHOOK.COMPONENT.HTML**

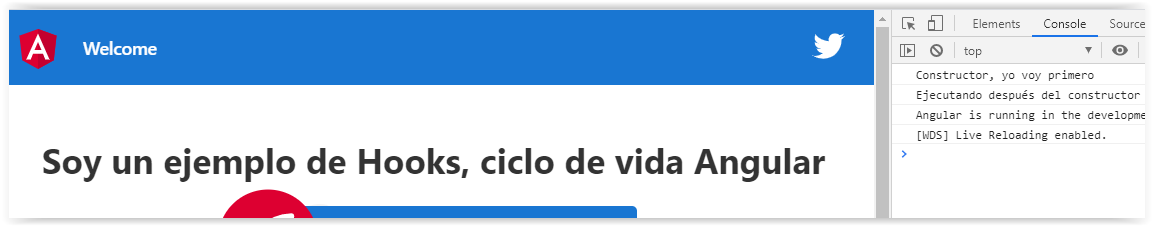
<h1>Soy un ejemplo de Hooks, ciclo de vida Angular</h1>

Y declaramos nuestro component en el archivo **app.module.ts**

**APP.MODULE.TS**



Y escribimos nuestro componente en la página inicial **app.component.html** y visualizamos la consola para mostrar el orden de ejecución del ciclo de vida.



El siguiente método es **ngDoCheck()**, que se ejecuta con cualquier cambio que exista en la vista.

Debemos realizar el import dentro de nuestro component e implementar el método en la clase.

import { Component, OnInit, DoCheck } from "@angular/core";

@Component({

  selector: "app-ejemplo-hook",

  templateUrl: "./ejemplohook.component.html"

})

export class EjemploHookComponent implements OnInit {

  constructor() {

    console.log("Constructor, yo voy primero");

  }

  ngOnInit(): void {

    console.log("Ejecutando después del constructor");

  }

  //SE EJECUTA CON CUALQUIER CAMBIO EN LA VISTA

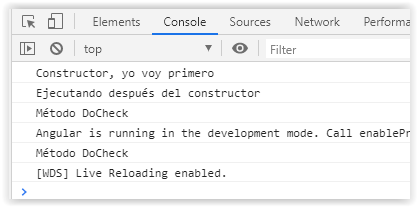
  ngDoCheck() {

    console.log("Método DoCheck");

  }

}

Al ejecutar, comprobaremos que se ejecuta en último lugar:



Ahora vamos a incluir un botón en la vista y una propiedad para mostrar un mensaje.

**TS**

import { Component, OnInit, **DoCheck** } from "@angular/core";

import { THIS\_EXPR } from "@angular/compiler/src/output/output\_ast";

@Component({

  selector: "app-ejemplo-hook",

  templateUrl: "./ejemplohook.component.html"

})

export class EjemploHookComponent implements OnInit, **DoCheck** {

  public mensaje: string;

  constructor() {

    console.log("Constructor, yo voy primero");

    this.mensaje = "Primer mensaje Angular";

  }

  ngOnInit(): void {

    console.log("Ejecutando después del constructor");

  }

  //SE EJECUTA CON CUALQUIER CAMBIO EN LA VISTA

  ngDoCheck() {

    console.log("Método DoCheck");

  }

  cambiarMensaje() {

    this.mensaje = "Has pulsado el botón!!!";

  }

}

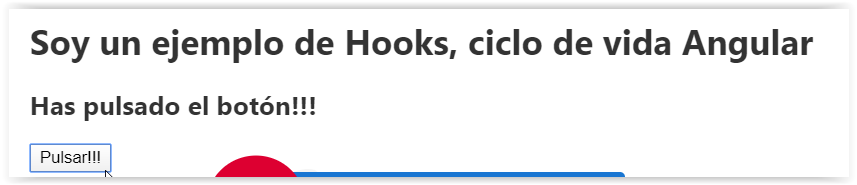
**HTML**

<h1>Soy un ejemplo de Hooks, ciclo de vida Angular</h1>

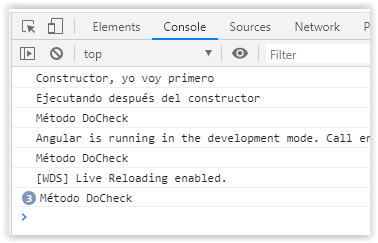
<h2>{{ mensaje }}</h2>

<button (click)="cambiarMensaje()">Pulsar!!!</button>

Y visualizaremos que se ha cambiado el mensaje



Y también se ha ejecutado **DoCheck()**



El otro Hook sería **OnDestroy()** que se ejecuta cuando se va a eliminar el component, al eliminar la instancia.

**TS**

import { Component, OnInit, DoCheck, OnDestroy } from "@angular/core";

import { THIS\_EXPR } from "@angular/compiler/src/output/output\_ast";

@Component({

  selector: "app-ejemplo-hook",

  templateUrl: "./ejemplohook.component.html"

})

export class EjemploHookComponent implements OnInit, DoCheck, OnDestroy {

  public mensaje: string;

  constructor() {

    console.log("Constructor, yo voy primero");

    this.mensaje = "Primer mensaje Angular";

  }

  ngOnInit(): void {

    console.log("Ejecutando después del constructor");

  }

  //SE EJECUTA CON CUALQUIER CAMBIO EN LA VISTA

  ngDoCheck() {

    console.log("Método DoCheck");

  }

  cambiarMensaje() {

    this.mensaje = "Has pulsado el botón!!!";

  }

  ngOnDestroy() {

    console.log("Se va a eliminar el Componente OnDestroy()");

  }

}

Para implementar la acción, utilizaremos el componente de **primercomponente** creado anteriormente.

Lo que haremos será ocultar el componente al pulsar un botón.

**PRIMERCOMPONENTE.COMPONENT.TS**

import { Component } from "@angular/core";

@Component({

  //INDICAMOS EL NOMBRE DEL COMPONENTE

  selector: "primercomponente",

  //INDICAMOS LA PLANTILLA PARA EL COMPONENT

  templateUrl: "./primercomponente.component.html"

})

export class PrimerComponente {

  public titulo: string;

  public descripcion: string;

  public anyo: number;

  public ocultar: boolean;

  //TENEMOS UN CONSTRUCTOR PARA LA PRIMERA VEZ

  //DE CARGA DEL COMPONENT

  constructor() {

    console.log("Creando el primer component");

    this.titulo = "Propiedad título del Componente";

    this.descripcion = "Aprendiendo a crear componentes en Angular";

    this.anyo = 2020;

    this.ocultar = false;

  }

  ocultarComponente() {

    this.ocultar = true;

  }

}

**PRIMERCOMPONENTE.COMPONENT.HTML**

<h1>Soy el primer componente....</h1>

<h1>{{ titulo }}</h1>

<p>{{ descripcion }}</p>

<p>{{ anyo }}</p>

<hr />

<button (click)="ocultarComponente()">Ocultar componente Hook</button>

<div \*ngIf="!ocultar">

  <app-ejemplo-hook></app-ejemplo-hook>

</div>



Y visualizaremos el ciclo de vida

