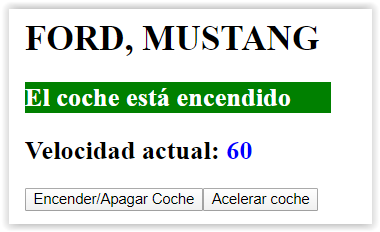
**ENVIAR INFORMACION ENTRE PARENT E HIJO**

* Vamos a visualizar cómo podemos enviar información desde un parent a un hijo.
* Para ello, nos crearemos un componente Coche que podrá acelerar y tendrá encendido y apagado.
* Cuando veamos que el coche es funcional, nos crearemos un componente llamado **Concesionario** que contendrá un conjunto de Coches y que enviará los coches al hijo.



Comenzamos creando una carpeta llamada **models** dentro de **src/app**

Dentro de la carpeta, creamos un nuevo fichero llamado **coche.ts**

**COCHE.TS**

export class Coche {

  constructor(

    public marca: string,

    public modelo: string,

    public velocidad: number,

    public velocidadmaxima: number,

    public aceleracion: number,

    public estado: boolean

  ) {}

}

Creamos un nuevo **component** llamado **Coche**, dónde utilizaremos nuestro modelo recién creado.

**COCHE.COMPONENT.TS**

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

import { Coche } from "./../../models/coche";

@Component({

  selector: "app-coche",

  templateUrl: "./coche.component.html",

  styleUrls: ["./coche.component.css"]

})

export class CocheComponent implements OnInit {

  public car: Coche;

  public mensaje: string;

  acelerarCoche(aceleracion) {

    if (this.comprobarEstado() == false) {

      this.car.velocidad = 0;

      alert("El coche está apagado!!!");

    } else {

      this.car.velocidad = this.car.velocidad + aceleracion;

      if (this.car.velocidad >= this.car.velocidadmaxima) {

        this.car.velocidad = this.car.velocidadmaxima;

      }

    }

  }

  encenderCoche() {

    this.car.estado = !this.car.estado;

    this.comprobarEstado();

  }

  private comprobarEstado() {

    if (this.car.estado == false) {

      this.mensaje = "El coche está apagado";

      this.car.velocidad = 0;

      return false;

    } else {

      this.mensaje = "El coche está encendido";

      return true;

    }

  }

  constructor() {

    this.car = new Coche("DMG", "DELOREAN", 0, 180, 20, false);

    this.comprobarEstado();

  }

  ngOnInit(): void { }

}

Y dibujamos el coche en la vista y comprobamos su funcionalidad.

**COCHE.COMPONENT.CSS**

.rojo {

  color: white;

  width: 50%;

  background-color: red;

}

.verde {

  color: white;

  width: 50%;

  background-color: green;

}

.azul {

  color: blue;

}

.rosa {

  color: fuchsia;

}

**COCHE.COMPONENT.HTML**

<div>

  <h1>{{ car.marca }}, {{ car.modelo }}</h1>

  <h2

    [ngClass]="{

      verde: car.estado == true,

      rojo: car.estado == false

    }"

  >

    {{ mensaje }}

  </h2>

  <h2>

    Velocidad actual:

    <span class="azul"> {{ car.velocidad }}</span>

  </h2>

  <button (click)="encenderCoche()">

    Encender/Apagar Coche

  </button>

  <button (click)="acelerarCoche(car.aceleracion)">

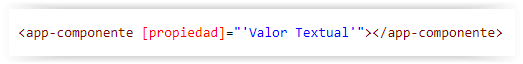
    Acelerar coche

  </button>

</div>

Para enviar datos entre componentes, se utiliza la siguiente sintaxis:

Estamos enviando un dato estático, es decir, un String a una propiedad del Parent.

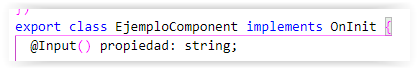


Dentro del padre, se recuperará con la palabra clave **@Input()**. Dicha declaración indica que estamos recibiendo un dato desde otro componente.

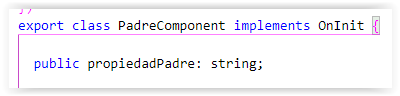
Primero debemos hacer referencia a **Input** dentro de los import.



Posteriormente, declaramos las variables que vayamos a recibir como propiedades:



También podemos enviar propiedades del Padre hasta el hijo. Para ello, debemos declarar una propiedad dentro del componente Padre:



Y para enviar la propiedad, lo haremos desde el código del padre, utilizando el nombre de la propiedad a enviar.



A continuación, enviaremos los datos de un componente a otro.

Tendremos un parent component que contendrá los coches.

Y enviaremos los coches al hijo (Coche) para visualizar su funcionalidad.

Creamos un nuevo **component** llamado **Concesionario**

**CONCESIONARIO.COMPONENT.TS**

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

import { Coche } from "./../../models/coche";

@Component({

  selector: "app-concesionario",

  templateUrl: "./concesionario.component.html",

  styleUrls: ["./concesionario.component.css"]

})

export class ConcesionarioComponent implements OnInit {

  public coches: Array<Coche>;

  constructor() {

    this.coches = [

      new Coche("FORD", "MUSTANG", 0, 320, 30, false),

      new Coche("KIA", "PICANTO", 0, 120, 5, false),

      new Coche("LAMBORGUINI", "DIABLO", 0, 40, 340, false)

    ];

  }

  ngOnInit(): void {}

}

**CONCESIONARIO.COMPONENT.HTML**

<h1>Concesionario de coches</h1>

<app-coche [car]="coches[0]"></app-coche>

<app-coche [car]="coches[1]"></app-coche>

<app-coche [car]="coches[2]"></app-coche>

A continuación, debemos modificar la clase hijo **Coche** para que reciba un coche dentro de la propiedad **car**.

Las claves están en que ya no inicializaremos el coche dentro del constructor, sino que nos será enviado.

Comprobaremos el estado del coche dentro del método **ngOnInit()**

**COCHE.COMPONENT.TS**

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

import { Coche } from "./../../models/coche";

@Component({

  selector: "app-coche",

  templateUrl: "./coche.component.html",

  styleUrls: ["./coche.component.css"]

})

export class CocheComponent implements OnInit {

  public mensaje: string;

  @Input() car: Coche;

  acelerarCoche(aceleracion) {

    if (this.comprobarEstado() == false) {

      this.car.velocidad = 0;

      alert("El coche está apagado!!!");

    } else {

      this.car.velocidad = this.car.velocidad + aceleracion;

      if (this.car.velocidad >= this.car.velocidadmaxima) {

        this.car.velocidad = this.car.velocidadmaxima;

      }

    }

  }

  encenderCoche() {

    this.car.estado = !this.car.estado;

    this.comprobarEstado();

  }

  private comprobarEstado() {

    if (this.car.estado == false) {

      this.mensaje = "El coche está apagado";

      this.car.velocidad = 0;

      return false;

    } else {

      this.mensaje = "El coche está encendido";

      return true;

    }

  }

  constructor() { }

  ngOnInit(): void {

    this.comprobarEstado();

  }

}

Y visualizaremos el resultado en nuestra página:

