formularios

template-driven forms

- la plantilla es la responsable de crear los controles, formularios y reglas de validación
- sincronización entre la vista y el modelo con el doble enlace (two-way databinding)
  - si la vista cambia, se actualiza el modelo, y viceversa
  - se enlaza con el atributo **name** de html y el atributo **ngModel** de Angular

```
<input id="nombre" name="nombre" class="form-control"
    type="text"
    placeholder="Nombre" ngModel>
```

  - **[(propiedad)]**[1], excepto si se usa **ngModel**
- importar el **FormsModule** en **app.module.ts**

caso

- doble binding con **ngModel** y **required**:

```
<input required id="nombre"
       class="form-control"
       type="text"
       placeholder="Nombre" name="nombre" ngModel>
```

  - el framewrok añadirá una serie de clases para indicar diferentews estados (ng-pristne, ng-dirty, ng-untoched, ng-touched, ng-valid, ng-invalid) que nos puede servir para darle nuestro estilo (css)
- añadir **#espia** y espia.classname para opciones de «debug» (y ver que clases tiene nuestro control):

```
<div class="form-group row">
   <label for="nombre" class="col-2 col-form-label">Nombre</label>
   <div class="col-8">
     <input #espia ngModel id="nombre" name="nombre" required class="form-
control" type="text" placeholder="Nombre">
  </div>
  {{espia.className}}
</div>
```

- además, para evitar la validación propia de HTML5 (y el navegador), tenemos:

```
<form [noValidate]="false">
```

  - si no la queremos, quitamos la etiqueta
- por contra, le añadimos una función y la vinculación a ella:

```
 guardar() {
  console.log("Formulario enviado");
}
```

```
<form (ngSubmit)="guardar()">
```

PPT 8.2~2

reactive forms (aka data-driven forms)

- la clase del componente es responsable de craer el formulario, sus controles y la configuración de validación

[1)]

banana in a box

From:
https://matebcn.eu/ - **miguel angel torres egea**

Permanent link:
https://matebcn.eu/development:angular:formularios

Last update: **10/02/2020 18:45**