

# BEEVA - CURSO DE ANGULAR 2015

## TEORÍA (KATA NIVEL INTERMEDIO)

Fernando Castro, Juan Ferrer

### CONTROL DE ERRORES EN FORMULARIOS

Hablando de los formularios, Angular dispone de un sistema muy ágil para el manejo de errores. A través de directivas nativas, podremos controlar diferentes casuísticas según el tipo de campo a tratar.

### ESTADOS

Disponemos de **varios estados principales** para controlar el formulario (también asociados a cada campo), que nos darán información útil a la hora de realizar un determinado tratamiento de los datos. Veamos algunos de los más importantes:

- **\$valid (true/false)**: nos indica si el formulario o el campo es válido con respecto a las reglas internas de Angular.
- **\$invalid (true/false)**: nos indica si el formulario o el campo no es válido. En caso del formulario completo, un sólo campo erróneo genera un true en esta propiedad.
- **\$dirty (true/false)**: nos indica si el usuario ha interactuado con un campo o con el formulario. En este caso, un sólo campo tratado genera un true en la propiedad.
- **\$pristine (true/false)**: nos indica si el usuario no ha interactuado con un campo o con el formulario.

Un ejemplo utilizando alguno de estos estados:

```
<ng-form name='form'>
  <div class="">
    <label for='nombre'>Nombre</label>
    <input type='text' class="" name='nombre' ng-model='model' required>

    <span ng-show='!form.$pristine && form.nombre.$error.required'>
      El nombre de usuario es obligatorio
    </span>
  </div>
</ng-form>
```

Podemos observar que el error se mostrará cuando se cumplan dos condiciones:

- que el usuario haya interactuado con algún campo
- que el campo en cuestión sea erróneo (en este caso, cuando no se haya informado, puesto que se marca como obligatorio).

## TIPOS DE ERRORES

¿Qué se debe añadir al input y qué condición utilizaremos para controlar algunos de los errores más comunes con los que nos podemos encontrar en un formulario?

- **required**: indicamos que el campo es obligatorio

```
<input type="text" id="value" name="value" ng-model="model.value" required />
```

- \$error.required

```
<span ng-show="form.field.$error.required">Campo obligatorio</span>
```

- **type=email**: el campo es de tipo email

```
<input type="email" id="value" name="value" ng-model="model.value" required />
```

- \$error.email

```
<span ng-show="form.field.$error.email">Debe ser una dirección de email válida</span>
```

- **type=text, pattern=**: una expresión regular para validar el dato introducido

```
<input type="text" id="value" name="value" ng-model="model.value"
ng-pattern="/^[A-Za-zÑñÁáÉéÍíÚúÁÁÉÉÍÍÚúÇç ]+$/"/>
```

- \$error.pattern

```
<span ng-show="form.field.$error.pattern">Debe escribir un valor válido</span>
```

- **type=number, min=**: mínimo para un valor numérico

```
<input type="number" id="value" name="value" ng-model="model.value" min="1" />
```

- \$error.min

```
<span ng-show="form.field.$error.min">Número mínimo no alcanzado</span>
```

- **type=number, max=**: máximo para un valor numérico

```
<input type="number" id="value" name="value" ng-model="model.value" max="1000000000" />
```

- \$error.max

```
<span ng-show="form.field.$error.max">Número máximo superado</span>
```