

# Angular Formularios

---

Para validar formularios en AngularJS lo haremos de la siguiente manera.

Empezaremos con éste formulario y a partir de aquí le iremos añadiendo funcionalidades para la validación.

```
<form role="form">
  <div class="form-group">
    <input class="form-control input-lg" placeholder="Correo electrónico">
  </div>
</form>
```

Primero de todo pondremos nombre al formulario y al input. De manera que después los podamos utilizar. Al hacer esto podemos acceder a cualquier característica del forms usando `myForm.myEmail`. Más adelante lo veremos.

```
<form role="form" name="myForm" >
  <div class="form-group">
    <input class="form-control input-lg" placeholder="Correo electrónico" name
    ="myEmail">
  </div>
</form>
```

En segundo lugar vamos a definir el tipo de input. Será un input de tipo `email`, es decir que se deberá entrar un texto tipo `example@example.com`. También pondremos que éste campo sea requerido con `required`

```
<form role="form" name="myForm" >
  <div class="form-group">
    <input class="form-control input-lg" placeholder="Correo electrónico" name
    ="myEmail" type="email" required>
  </div>
</form>
```

Añadiremos un `ng-model` al input de manera que podamos obtener sus valores en el javascript con el `$scope`.

```
<form role="form" name="myForm" >
  <div class="form-group">
    <input class="form-control input-lg" placeholder="Correo electrónico" name
```

```

    = "myEmail" type="email" ng-model="email" required>
    </div>
</form>
`

```

## Comprobación de los campos

Ahora ya podemos actuar en función de lo que el usuario haya introducido. En este caso con la directiva `ng-class` (que sirve para añadir css si se producen ciertas condiciones) podremos el input de color rojo si el texto ha sido modificado (`myForm.myEmail.$dirty`) y el formato es inválido (`myForm.myEmail.$invalid`). De manera que nos queda de la siguiente manera: `ng-class="{ 'has-error': myForm.myEmail.$dirty && myForm.myEmail.$invalid }"`.

Hay más propiedades que `$dirty` y `$invalid`. Estas son las siguientes:

- `$pristine` No se ha modificado nada
- `$dirty` Se ha modificado algo
- `$invalid` El contenido no es válido
- `$valid` El contenido es válido
- `$submitted` El formulario ha sido enviado

```

<form role="form" name="myForm" >
  <div class="form-group" ng-class="{ 'has-error': myForm.myEmail.$dirty && myForm.myEmail.$invalid }">
    <input class="form-control input-lg" placeholder="Correo electrónico" name="myEmail" type="email" ng-model="email" required>
  </div>
</form>

```

## Mostrar mensaje del error

Ahora simplemente nos queda enviar un mensaje indicando el error. Con un `ng-show` conseguiremos ocultar o mostrar el mensaje: `ng-show="myForm.myEmail.$dirty && myForm.myEmail.$invalid"`. Con esto conseguiremos que se muestre un error si el texto ha sido modificado (`myForm.myEmail.$dirty`) y el formato es inválido (`myForm.myEmail.$invalid`).

Tendremos dos mensajes a mostrar, uno para indicar que el mail es obligatorio y otro para indicar que el formato del mail no es correcto. Para saber que tipo de error tenemos, usaremos la propiedad `$error`. Para indicar que el campo es obligatorio usaremos `$error.required` y para indicar que formato del mail no es correcto, usaremos `$error.email`.

```

<span class="text-danger" ng-show="myForm.myEmail.$dirty && myForm.myEmail.$invalid">
  <h5 ng-show="myForm.myEmail.$error.required">El mail es obligatorio</h5>
  <h5 ng-show="myForm.myEmail.$error.email">El formato del mail no es correcto</h5>
</span>

```

Ya tenemos listo nuestro forms y queda de la siguiente manera.

```
<form role="form" name="myForm" >
  <div class="form-group" ng-class="{ 'has-error': myForm.myEmail.$dirty && myForm.myEmail.$invalid}">
    <input class="form-control input-lg" placeholder="Correo electrónico" name="myEmail" type="email" ng-model="email" required>
  </div>
</form>

<span class="text-danger" ng-show="myForm.myEmail.$dirty && myForm.myEmail.$invalid">
  <h5 ng-show="myForm.myEmail.$error.required">El mail es obligatorio</h5>
  <h5 ng-show="myForm.myEmail.$error.email">El formato del mail no es correcto</h5>
</span>
```

Si tenemos un boton de submit para enviar el formulario y queremos desabilitarlo si todos los campos no son correctos, usaremos ng-disabled con tal de desabilitarlo. De tal manera que tendremos ng-disabled="myForm.\$invalid", es decir que se desabilitará el botón si nuestro formulario es inválido.

```
<input ng-disabled="myForm.$invalid" type="submit" class="btn btn-lg btn-success btn-block" value="Submit">
```