

# PEC 5

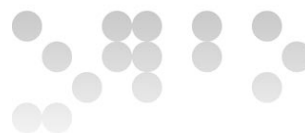
## Frameworks:

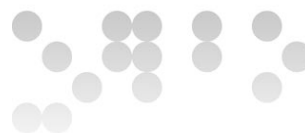
### Formularios en Angular

Desarrollo front-end con  
frameworks Javascript

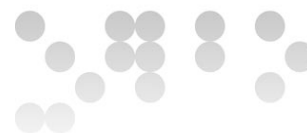
Máster Universitario en Desarrollo de sitios y  
aplicaciones web  
Semestre 20201

Estudios de Informática, Multimedia y Telecomunicación





# TEORÍA



## Formularios dirigidos por el *template* y Formularios Reactivos

En la anterior PEC se desarrolló una aplicación básica en Angular que estaba compuesta por componentes simples que respondían a las interacciones de los usuarios. En esta aplicación se han desarrollado componentes simples y se han realizado transformaciones de datos entre componentes, así como el “bindeo” de eventos entre el DOM y nuestros componentes sin necesidad de utilizar el DOM de JavaScript.

En esta PEC se trabajará el principal elemento que tienen todas las aplicaciones que recogen datos introducidos por los usuarios: **los formularios**. Los formularios son utilizados en la mayoría de aplicaciones web puesto que es el elemento que permite interactuar de manera más sencilla a los usuarios con nuestra aplicación. No obstante, la creación de formularios no es una tarea sencilla puesto que está compuesta de diferentes partes que deben ser controladas:

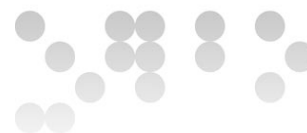
- Aspecto visual (*template*).
- Bindeo entre el formulario y el controlador.
- Seguimiento del estado de la aplicación (todos los objetos de la aplicación y estados de la GUI)
- Validaciones de las entradas de los usuarios.
- Gestión de errores.

A medida que los casos de uso se vuelven más complejos, la gestión de los formularios se transforma en una tarea mucho más compleja que debe ser gestionada de un modo organizado. Existen dos técnicas para trabajar con los formularios en Angular:

1. Formularios dirigidos por la plantilla (*Template Driven*).
2. Formularios reactivos.

### Dirigidos por la plantilla (*Template-Driven*)

Los formularios dirigidos por el *template* en Angular son una extensión de cómo se crean y trabajan con componentes. Por eso es la primera técnica que se suele utilizar para aprender a gestionar formularios. Esta técnica es parecida a la que se utilizaba en la gestión de formularios en el framework Angular JS (1.x).



En esta PEC se introducen los formularios dirigidos por el template por el hecho de que existe mucho código generado utilizando esta técnica, sobre todo en las primeras fases de un proyecto software. No obstante, se recomienda que se utilice la segunda técnica (i.e. los formularios reactivos).

Para la superación de esta práctica debes leer y realizar el paso a paso del código relativo al Capítulo 6 (<https://learning.oreilly.com/library/view/angular-up-and/9781491999820/ch06.html#chapter-6>).

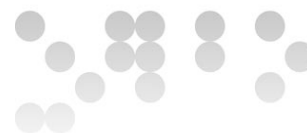
## Formularios Reactivos

Los formularios reactivos realizan la misma funcionalidad que los formularios dirigidos en *templates* pero con la diferencia de que están implementados utilizando el paradigma de programación reactivo. Angular, utiliza en toda su arquitectura de desarrollo la combinación de dos paradigmas de programación: funcional y reactivo. La programación funcional es la herencia del lenguaje JavaScript (tal y como se ha ido viendo a lo largo de las PECs), mientras que la programación reactiva es una excelente opción cuando se están construyendo interfaces de usuario (frontend), puesto que el sistema actuará sólo cuando se produzca algún tipo de evento, ya sea por parte del usuario o de otro componente del sistema. Por tanto, antes de introducirnos en la programación reactiva (asignatura avanzada de frontend), se va a hacer uso de los formularios reactivos, los cuales están desarrollados por el equipo de desarrollo de Angular y encajan perfectamente con todas las piezas de la arquitectura de Angular que se irán viendo a lo largo de esta asignatura y la asignatura de desarrollo frontend avanzado.

Para la superación de esta práctica debes leer y realizar el paso a paso del código relativo al Capítulo 7 (<https://learning.oreilly.com/library/view/angular-up-and/9781491999820/ch07.html>).

Uno de los puntos que no han sido tratados en el recurso anterior es la creación de validadores propios (*custom validators*). En ocasiones puede ser interesante poder crear validadores que cumplan reglas de negocio de nuestra aplicación. Esto se desarrolla de manera sencilla tal y como se puede leer en los siguientes recursos:

<https://codinglatte.com/posts/angular/angular-building-custom-validators/>  
<https://angular.io/guide/form-validation#custom-validators>



**NOTA:** Como miembros de la Comunidad UOC, tenéis disponible un gran catálogo de recursos de la editorial O'Reilly. Para acceder debéis seguir los siguientes pasos:

- \* Acceder a <https://learning.oreilly.com>
- \* Cuando te pida el *login* y *password*, escribir en el *campo Email Address or Username* vuestro correo electrónico de la UOC, p.ej. [dgarciaso@uoc.edu](mailto:dgarciaso@uoc.edu)
- \* Después pulsáis fuera con el ratón y desaparecerá el campo *password*. Además el botón de *Sign In* cambiará por un botón rojo con el texto *Sign In with Single Sign On*.
- \* A partir de aquí seguid los pasos que os vaya pidiendo, seguramente os llevará a una página de autenticación con el logo de la UOC. Además os llegará un e-mail a vuestra cuenta de correo electrónico de la UOC avisando de que os habéis suscrito a este catálogo.
- \* Una vez tengáis acceso al catálogo de O'Reilly, para ver el contenido de los enlaces que os proporcionamos, sólo tenéis que estar autenticados en O'Reilly y abrir una pestaña nueva en vuestro navegador y copiar el enlace que os facilitamos.