AMS Bootcamp Fullstack Developer.

**Angular**



Índice

[1. Personal Notebook 4](#_Toc155540130)

[1.1 Datos alumno 4](#_Toc155540131)

[1.2 Personal Notebook 4](#_Toc155540132)

[2. Angular 5](#_Toc155540133)

[2.1 Objetivos de aprendizaje 5](#_Toc155540134)

[2.2 Angular: Instalación, estructura y flujo 6](#_Toc155540135)

[2.2.1 Objetivos 6](#_Toc155540136)

[2.2.2 Recursos formativos 6](#_Toc155540137)

[2.2.3 Posibles preguntas 7](#_Toc155540138)

[2.2.4 Tareas 7](#_Toc155540139)

[2.2.5 Dudas 7](#_Toc155540140)

[2.3 Componentes, Interpolación y Binding 8](#_Toc155540141)

[2.3.1 Objetivos 8](#_Toc155540142)

[2.3.2 Recursos formativos 8](#_Toc155540143)

[2.3.3 Posibles preguntas 8](#_Toc155540144)

[2.3.4 Tareas 8](#_Toc155540145)

[2.3.5 Dudas 8](#_Toc155540146)

[2.4 Directivas, Comunicación entre componentes y Servicios 9](#_Toc155540147)

[2.4.1 Objetivos 9](#_Toc155540148)

[2.4.2 Recursos formativos 9](#_Toc155540149)

[2.4.3 Posibles preguntas 9](#_Toc155540150)

[2.4.4 Tareas 9](#_Toc155540151)

[2.4.5 Dudas 9](#_Toc155540152)

[2.5 Routing y Login 10](#_Toc155540153)

[2.5.1 Objetivos 10](#_Toc155540154)

[2.5.2 Recursos formativos 10](#_Toc155540155)

[2.5.3 Posibles preguntas 10](#_Toc155540156)

[2.5.4 Tareas 10](#_Toc155540157)

[2.5.5 Dudas 10](#_Toc155540158)

[2.6 Actualización Angular 17 11](#_Toc155540159)

[2.6.1 Objetivos 11](#_Toc155540160)

[2.6.2 Recursos formativos 11](#_Toc155540161)

[2.6.3 Posibles preguntas 11](#_Toc155540162)

[2.6.4 Tareas 11](#_Toc155540163)

[2.6.5 Dudas 11](#_Toc155540164)

[2.7 Ejercicio práctico final 12](#_Toc155540165)

[2.7.1 Objetivos 12](#_Toc155540166)

[2.7.2 Recursos formativos 12](#_Toc155540167)

[2.7.3 Posibles preguntas 12](#_Toc155540168)

[2.7.4 Tareas 12](#_Toc155540169)

[2.7.5 Dudas 12](#_Toc155540170)

# Personal Notebook

## Datos alumno

* DAS
* Nombre Apellidos
* email
* URL de GitHub corporativo

## Personal Notebook

Este es tu cuaderno de trabajo. En él tendrás que registrar tu aprendizaje, tus avances y tus dudas. ¡Vas a generar tu propio material!

Puede ser una gran herramienta si la utilizas correctamente.

El cuaderno estará compartido con los tutores y el resto de los colaboradores para ver tu progreso y ayudarte.

Al final del Bootcamp tendrás que entregarlo como evidencia de tu aprendizaje.



Instrucciones de uso:

El fichero está ubicado en tu carpeta del espacio que tenemos en Teams.

Siempre deberá ser editado en modo online y/o con guardado automático en la nube.

Responde a las preguntas con la información que necesites para dar una respuesta cualificada al formador o a tus compañeros. No es necesario copiar texto solo por “rellenar”. Cita las Fuentes en caso de utilizar otros recursos.

# Angular

## Objetivos de aprendizaje

* Objetivo: **Adquirir los conocimientos básicos sobre el framework Angular. Comprender como se crean aplicaciones web basadas en Angular y su integración con sistemas de Backend.**
* Conceptos principales a adquirir:
  + Estructura
  + Componentes
  + Binding
  + Directivas
  + Forms
  + Servicios
  + Routing
  + Http y autenticación

Lecciones:

1. Angular: componentes, directivas, observables, validaciones, forms
2. Ejercicio práctico

## Angular: Instalación, estructura y flujo

### Objetivos

**Instalar todas las herramientas necesarias para poder usar angular, entender que es angular, conocer su estructura y su flujo de funcionamiento.**

### Recursos formativos

* <https://youtube.com/playlist?list=PLU8oAlHdN5BnNAe8zXnuBNzKID39DUwcO> (1-4)

**Nota importante**: Al seguir el segundo video de la serie, es crucial tener en cuenta que la instalación de Node.js debe realizarse de acuerdo con las instrucciones detalladas en el siguiente enlace:

[Instalación nodejs con nvm](https://www.youtube.com/watch?v=Z-Ofqd2yBCc&pp=ygUdaW5zdGFsYXIgbm9kZSB2ZXJzaW9uIG1hbmFnZXI%3D)

Asegúrate de instalar la versión de Node.js que corresponde a la versión LTS (Long Term Support) para garantizar la compatibilidad con el desarrollo en Angular 17.

Es importante destacar que, en la sección del video dedicada a la instalación de Angular, se recomienda instalar la última versión disponible, que en este caso es la versión 17. Esto se logra ejecutando el siguiente comando en la terminal sin indicar una versión específica:

*npm install @angular/cli*

**Nota Importante para la Creación de Proyectos en Angular:**

Para llevar a cabo el curso de Angular con la versión 17, es esencial generar los proyectos utilizando el siguiente comando:

*ng new nombre-proyecto --no-standalone*

*A la pregunta ‘Do you want to enable server-side rendering and static site generation?’ responderemos con ‘N’.*

Esto se debe a que esta versión utiliza por defecto componentes standalone, que no requiere la configuración tradicional en el archivo app.module.ts. El comando asegura la compatibilidad con los cursos diseñados para versiones anteriores, donde sí se utiliza.

* [Angular - What is Angular?](https://angular.io/guide/what-is-angular)
* [Angular - Understanding Angular](https://angular.io/guide/understanding-angular-overview)

VIP: recuerda que tienes que responder a estas preguntas de forma que te ayude a explicárselas al formador o a los compañeros!!

* [Estructura de Carpetas en Angular](https://codigoencasa.com/estructura-de-carpetas-en-angular/)

### Posibles preguntas

* ¿Qué es Angular? ¿Qué características tiene? ¿Por qué utilizarlo?
* ¿Cuáles diferencias hay entre una web SPA y una tradicional?
* ¿Para qué se usa el archivo app.module.ts?
* ¿Qué es Typescript? ¿Cómo funciona? ¿Qué diferencias tiene con Javascript?
* ¿Qué diferencias hay entre React y Angular?

### Tareas

* NA

### Dudas

¿?

VIP: registra las dudas que tengas para la sesión con el formador!!

## Componentes, Interpolación y Binding

### Objetivos

* **Conocer y comprender los componentes en angular y por qué se usan, la interpolación y el binding**

### Recursos formativos

* <https://youtube.com/playlist?list=PLU8oAlHdN5BnNAe8zXnuBNzKID39DUwcO> (5-13)
* [Angular - Angular components overview](https://angular.io/guide/component-overview)
* [Componentes - Aprendiendo Angular (gitbook.io)](https://ngchallenges.gitbook.io/project/componentes)
* [Interpolación {{}} - Aprendiendo Angular (gitbook.io)](https://ngchallenges.gitbook.io/project/interpolacion)

### Posibles preguntas

* ¿Qué es un componente en Angular? ¿Por qué es mejor usar componentes? ¿De qué formas se pueden crear?
* ¿Qué es la interpolación en Angular? ¿Cómo funciona eso mostrar datos de forma dinámica?
* ¿Cuál es la estructura del operador ternario en una plantilla html?
* ¿Qué es un enlace unidireccional en Angular? ¿Y un enlace bidireccional?

### Tareas

* Crear una aplicación como la de los vídeos 12-13

### Dudas

¿?

## Directivas, Comunicación entre componentes y Servicios

### Objetivos

* **Conocer y comprender el uso de las directivas en Angular, la comunicación entre componentes de Angular y el uso de los Servicios.**

### Recursos formativos

* <https://youtube.com/playlist?list=PLU8oAlHdN5BnNAe8zXnuBNzKID39DUwcO> (14-23)
* [Angular - Built-in directives](https://angular.io/guide/built-in-directives)
* [Directivas - Aprendiendo Angular (gitbook.io)](https://ngchallenges.gitbook.io/project/directivas)

### Posibles preguntas

* ¿Qué es una directiva en Angular? ¿Cómo se crea? ¿Para qué se usan?
* ¿Qué tipos de directivas hay?
* Nombra algunas de las directivas estructurales que podemos encontrar en angular y explica su uso
* ¿Qué es un servicio en Angular? ¿Cómo se crea? ¿Para qué se usan?
* ¿Cómo se utilizan los servicios para compartir lógica entre componentes?
* ¿Qué es una comunicación entre componentes en Angular? ¿Y una comunicación entre servicios?

### Tareas

* NA

### Dudas

¿?

## Routing y Login

### Objetivos

* **Conocer, comprender y aprender a utilizar el Routing en angular, la forma de hacer un Login en angular, los usuarios y roles y la protección de páginas mediante Guardians**

### Recursos formativos

* <https://youtube.com/playlist?list=PLU8oAlHdN5BnNAe8zXnuBNzKID39DUwcO> (24-46)

**Nota importante para la gestión de rutas en Angular:**

En el video dedicado a la implementación de rutas en Angular, el archivo a modificar es el *app.routing.module*, el cual se encuentra en el directorio raíz de la aplicación. A diferencia de otras configuraciones, no se recomienda modificar directamente el archivo *app.module*. La gestión de rutas en Angular se centraliza en el módulo de enrutamiento (*app.routing.module*), donde se definen y configuran las rutas específicas de la aplicación.

* Al seguir este enfoque, se garantiza una organización más clara y modular de las rutas, lo que facilita la comprensión y el mantenimiento del código.

JWT Angular: JWT Angular: [Angular. ¿Cómo crear el Login?. components + routing + forms + services + observables + http. (youtube.com)](https://www.youtube.com/watch?v=jkdLRbrJj9M)

* [Angular - Angular Routing](https://angular.io/guide/routing-overview)
* [Introducción al sistema de Routing en Angular (desarrolloweb.com)](https://desarrolloweb.com/articulos/introduccion-sistema-routing-angular.html)

### Posibles preguntas

* ¿Qué es el enrutamiento en Angular? ¿Cómo se configura?
* ¿Qué son los Guards de enrutamiento en Angular? ¿Cómo se usan?
* ¿Qué son los Resolvers de enrutamiento en Angular? ¿Cómo se usan?

### Tareas

* NA

### Dudas

¿?

## Actualización Angular 17

### Objetivos

* **Explorar y familiarizarse con las nuevas características y mejoras introducidas en Angular 17 en comparación con versiones anteriores.**

### Recursos formativos

* + [Aprende ANGULAR 17](https://youtu.be/RK_ZPKn_I3I?si=3EVegPbO5M8DLhgY)
  + [Angular 17: Cosas que debes saber de la nueva versión](https://medium.com/@dgongoragamboa/angular-17-cosas-que-debes-de-saber-de-la-nueva-versi%C3%B3n-ad22d0cd53de)

### Posibles preguntas

* ¿Qué significa que Angular 17 utiliza standalone por defecto en sus componentes?
* ¿Qué debemos hacer para utilizar un componente dentro de otro?
* ¿Qué son y en qué casos se utilizan los ‘defer blocks’?
* ¿En qué archivo se importan las rutas por defecto?

### Tareas

* N/A

### Dudas

¿?

## Ejercicio práctico final

### Objetivos

* **Hacer uso de todos los conocimientos adquiridos para crear una aplicación usando Angular en la parte de front, Spring en la parte de back y comunicándolas mediante comunicación rest**

### Recursos formativos

* https://www.udemy.com/angular-spring/ ([Angular & Spring 5 Creando web app full stack (Angular 8+).zip](https://eviden.sharepoint.com/:u:/s/FormadoresCanteraSistemasAbiertos/ETanxXsw5fJCi1VnJEKdjhwBtRSEmTxStp_5wn2wxscWkg?e=jZdL62))
* [Angular - Communicating with backend services using HTTP](https://angular.io/guide/http)

### Posibles preguntas

* ¿Qué es la programación reactiva y qué es RxJS??
* ¿Qué es el patrón Observable? ¿Por qué se usa en Angular?
* ¿Qué es la programación orientada a eventos?
* ¿Cómo se gestiona la seguridad en las aplicaciones Angular?
* ¿Dónde se almacena la sesión? (¿en front o back) ¿Cómo?

### Tareas

* El ejercicio es la creación del aplicativo como el que se está realizando en el recurso formativo de “Angular & Spring 5 Creando web app full stack” (hasta la sección 10 aprox)
* El ejercicio tiene que estar plenamente operativo.
* Se valorará positivamente la realización de un caso de uso personal, definido por el alumno (en vez del existente en el curso de “Angular & Spring 5 Creando web app full stack”), así como la utilización del framework en su versión 17.

### Dudas

¿?