



# Angular S8-9-10-11: Proyecto B

Bienvenidos al final de Angular, si has decidido optar por trabajar con una API externa o usar una API fake con Json-Server aquí tendrás una pequeña guía de todo lo que debemos tener en nuestro proyecto. Definiremos diferentes pantallas con la funcionalidad asociada a cada una de ellas, esto nos servirá para afianzar los conocimientos adquiridos a lo largo del bloque de Angular. Comenzamos!

## ITERATION - 1

Lo primero será definir qué pantallas tiene nuestro proyecto, en cada una de ellas daremos una breve descripción de lo que ha de contener, vamos a ello:

**HOME** → En la pantalla principal explicaremos en qué consiste nuestra aplicación. Por ejemplo si nos hemos decidido por la temática de super-heroes explicaremos:

1. Qué es un Super-heroe.
2. Requisitos para ser un Super-heroe.
3. Funcionamiento de nuestra Web.

En la Home tendremos un componente padre y tres hijos con cada uno de los bloques definidos.

**LIST** → En la pantalla de listado tendremos una lista de todos nuestros elementos, ya sean productos, jugadores de la NBA o en el caso de este ejemplo Super-heroes. Para ello tenemos que tener en cuenta lo siguiente:

1. Definir nuestro modelo de datos. Interfaz del listado.
2. Servicio que realiza la petición a la API → Si es global irá en la carpeta Services.
3. Componente hijo → Card-list → tarjeta que pinta cada uno de los elementos del listado.
4. Pipe para la paginación → se pintarán un máximo de 25 elementos por página.
5. Pipe de búsqueda por nombre → buscará en el listado el nombre que has introducido a tiempo real.

En el List tendremos un componente padre, un componente hijo (card) y dos Pipes. El servicio depende de si es global o único. En cada Card se mostrarán dos atributos - Nombre - Imagen.

**DETAIL** → En la pantalla de detalle mostrará una información ampliada del elemento seleccionado, en este caso del Super-heroe. Vamos a desgranarlo poco a poco:

1. Definir nuestro modelo de datos. Interfaz del detalle.
2. Servicio que realiza la petición a la API → Si es global irá en la carpeta Services.
3. Componentes hijos (mínimo 3 hijos):
  1. Biography → Descripción biográfica del elemento.
  2. Powerstats → Poderes del elemento.
  3. Appearance → Apariencia o datos relativos a su físico.
  4. Works → Trabajo que desempeña.

5. Connections → Conexiones con otros personajes o grupos.
6. Image → Componente Imagen → cuando haces Hover/Click debe hacer una transición mostrando otro elemento o ampliando la img.

**NAVBAR** → Barra de navegación con al menos las secciones comentadas arriba, HOME y LIST ya que al DETAIL se accede desde el listado de elementos.

Distribución de la aplicación en módulos. La aplicación deberá contar con al menos dos módulos, tal y como os hemos indicado con anterioridad uno será el shared.module, lo correcto es que uses además de este, un módulo por página también.

Router: Deberás hacer uso del router y aplicar carga dinámica de módulos a través del lazy loading.

## ITERATION - 2

**MIX** → Posibilidad de combinar peticiones a dos o más APIS diferentes para mostrar un listado de elementos. En el caso de Super-heroes se puede combinar las APIS de Marvel y Super-Hero.

**FORMS** → Usando una API propia o API fake propia, crear formulario que permita añadir elementos. Si has añadido nuevos elementos deberás tener la opción de visualizarlo en una nueva pantalla bajo el nombre de My Heroes o My NBA Players... Además daremos la opción de editar y eliminar elementos.

**MY CREATES** → Listado de elementos que hemos generado en nuestros formularios. En cada uno de estos elementos deben tener dos botones, uno para UPDATE y otro para DELETE.

**ABOUT** → Explicaremos qué nos ha llevado hacer este proyecto, si es en colaboración una descripción de cada uno de sus integrantes, los Gitlabs de cada uno de los miembros del equipo, agradecimientos si los hubiese y no estaría de más algún aporte divertido.

**NABVAR** → Ampliar el Nabvar con las pantalla que hemos implementado.

**FOOTER** → Implementar un Footer común para todas las pantallas o vistas.

## BONUS

**RESPONSIVE** → Que nuestra aplicación sea responsive. Recordad el mobile first, usar BEM y el buen uso de SCSS o SASS.

**IDIOMAS** → Implementar opciones para visualizar Castellano, Inglés u otros. Para ello puedes hacer uso de ng-translate o i18n.

**TESTING** → Añadir testing en alguna de las funcionalidades implementadas. Test unitarios.