

# Fase 1: crear app e implementar enrutamiento

## Enunciado.

Vamos a desarrollar una aplicación de Angular con **enrutamiento** (*routing*) que se llame **Despotify**. La aplicación permitirá consultar información sobre canciones de una base de datos. También permitirá que un usuario elabore una lista de reproducción.

## Hoja de estilos

De manera general, la aplicación usará **Bootstrap**, por lo que es necesario enlazar los archivos requeridos. Los estilos que sean exclusivos de un componente estarán en el CSS de dicho componente.

## Origen de datos

Los datos van a ser obtenidos de las siguientes fuentes por orden de prioridad, es decir, se intentará la fuente (1) y si hay algún obstáculo insalvable pasaremos a la fuente (2) y así sucesivamente:

1. La **API** de **My JSON Server** en <https://my-json-server.typicode.com/>.
2. Si fallara el acceso a este último servicio entonces el profesor os proporcionaría un archivo **JSON** para integrar en nuestra app como recurso local.

Las URL base para consultar datos son:

URL	<a href="https://my-json-server.typicode.com/songs">https://my-json-server.typicode.com/songs</a>
Parámetros	JSON con listado de canciones
URL	<a href="https://my-json-server.typicode.com/playlists">https://my-json-server.typicode.com/playlists</a>
Parámetros	JSON con listado de <i>playlists</i>

## Modelos de *templates* entregados por el profesor

Para ahorrar tiempo, se entregan algunas *templates* con código HTML genérico que ayudan mucho, pero deben ser adaptadas previamente para funcionar bien.

## Componente raíz

A continuación, se indica una lista de los 3 componentes que será necesario crear para dar contenido al componente raíz:

- **Componente para el encabezado.** Muestra la imagen logotipo de la aplicación y el nombre de la aplicación.
- **Componente para la barra de navegación.** Ofrecerá la posibilidad de clicar en las secciones listadas en el apartado **Navegación y enrutamiento** más abajo.
- **Componente para el pie.** Muestra el nombre del autor o autora de la aplicación, es decir, tu nombre. También cita la fuente de los datos.

## Navegación y enrutamiento

Justo debajo del encabezado aparecerá una barra de navegación horizontal con varias secciones:



## Some statistics about our database:

Number of songs: 30
Number of playlists: 4
Average number of songs in playlists: 0
Number of artists: 0
Top artist by songs:
Top song by length: "Un instante de luz", Robe (10:34)
Artists ordered by number of songs
Kings of Leon (7)
L.A. (4)
Pearl Jam (4)

**Sección *Home*:** muestra algunas estadísticas sobre la base de datos. Además de unas cifras, también muestra un listado ordenado de artistas de mayor a menor número de canciones.

**Sección *Songs*:** muestra un campo de texto para buscar por nombre de artista o título de canción. Listará resultados al presionar la tecla Intro. La vista se puede cambiar con el *switch*.

search by title or artist...			Search
Songs			Toggle card/rows views <input checked="" type="checkbox"/>
"Low" Lenny Kravitz 5:18 <a href="#">Add to playlist</a>	"Evil" Interpol 3:36 <a href="#">Add to playlist</a>	"Dublin in Ecstasy" Inhaler 5:11 <a href="#">Add to playlist</a>	"Fly To You" Caroline Polachek 4:05 <a href="#">Add to playlist</a>
"Learn to Fly" Foo Fighters 3:55 <a href="#">Add to playlist</a>	"Walk" Foo Fighters 4:16 <a href="#">Add to playlist</a>	"Wishlist" Pearl Jam 3:27 <a href="#">Add to playlist</a>	"High and Dry" Radiohead 4:17 <a href="#">Add to playlist</a>
search by title or artist...			Search
Songs			Toggle card/rows views <input type="checkbox"/>
"Low"	Lenny Kravitz	5:18	<a href="#">Add to playlist</a>
"Evil"	Interpol	3:36	<a href="#">Add to playlist</a>
"Dublin in Ecstasy"	Inhaler	5:11	<a href="#">Add to playlist</a>

## Playlists:

Playlist: Rock americano Author: Luismi Songs: 5,6,7,8,10,11,12,18,19,20,21,22,27,28
Playlist: Grupos españoles Author: Luismi Songs: 13,14,15,16,17,23,24,25,26,29,30
Playlist: One hit wonders Author: Luismi Songs: 1,2,3,4
Playlist: Vacía Author: Desconocido Songs:

**Sección *Playlists*:** muestra las listas de reproducción. La última lista está vacía para poder añadir canciones en esta lista.

# Fase 2: desarrollar servicios y filtros

## Servicios

Tienes libertad para implementar servicios a tu criterio. Sin embargo, si te sirve de guía, aquí tienes una sugerencia:

- **Un servicio para obtener datos del origen de datos.** Es capaz de hacer llamadas *HTTP* para obtener los datos en formato *JSON* necesarios para alimentar a la aplicación.

Las canciones añadidas por el usuario a la última lista de reproducción no deben perderse, por lo que hay que gestionar la **persistencia** en el almacenamiento local del navegador (**local storage**).

## Filtros

Implementar los siguientes filtros:

- Filtro para que se muestren únicamente las canciones cuyo título o nombre de artista incluye una cadena patrón introducida por el usuario.
- Filtro para convertir duraciones de canciones de segundos al formato **mm:ss** (donde *mm* son minutos y *ss* son segundos).

## Objetivos

Objetivos obligatorios	OK
Los contenidos del componente raíz se ajustan a lo pedido. [0.25p]	
La barra de navegación funciona. [0.75p]	
Un servicio obtiene datos del servidor mediante HTTP y dichos datos son usados y mostrados correctamente en alguna parte de la aplicación. [1p]	
El usuario puede añadir canciones a la última lista de reproducción (inicialmente es una lista vacía y no puede contener canciones repetidas). [0.5p]	
Se muestran las estadísticas básicas (datos individuales) en la página de inicio de la aplicación. [0.75p]	
Los filtros han sido implementados: El buscador muestra resultados correctos al buscar por patrón sin importar mayúsculas o minúsculas. [0.75p] La duración se muestra en formato mm:ss. [0.5p]	
Las preferencias del usuario se conservan en <i>LocalStorage</i> . [0.5p]	
Objetivos opcionales	OK
No se realizan llamadas HTTP innecesarias y el aspecto está cuidado y los errores en la consola controlados. [1p]	
Se puede eliminar canciones de una lista de reproducción. [2p]	
Se muestran las estadísticas avanzadas (listado ordenado de artistas) en la página de inicio de la aplicación. [2p]	

## Conocimientos examinados

En este examen se **requiere** que el alumno o la alumna sepa:

- Crear aplicaciones web con Angular
- Insertar nuevos componentes en la aplicación
- Capturar eventos producidos por el usuario sobre ciertos elementos
- Definir enrutamiento para navegar dentro de la aplicación
- Establecer vías de comunicación entre componentes
- Añadir servicios e inyectarlos en los componentes que los requieran
- Realizar llamadas HTTP asíncronas para interactuar con una base de datos

En este examen se **valora** que el alumno o la alumna sepa:

- Detectar y manejar los posibles errores en tiempo de ejecución
- Ahorrar ancho de banda
- Implementar aspectos avanzados en aplicaciones web para aumentar la funcionalidad de la misma

## Calificación del examen

La correcta resolución de todos los objetivos obligatorios otorga una calificación de 5 sobre 10. Solamente si todos los objetivos obligatorios han sido resueltos adecuadamente se procederá a sumar puntuación adicional por cada uno de los objetivos opcionales solucionados con éxito, hasta un máximo de 5 puntos adicionales si se implementan 5 de los objetivos opcionales.