

# Ejercicio 1

## Enunciado.

Vamos a desarrollar una aplicación de Angular que se llame CineDB. Intenta crear la aplicación base desde una consola que pregunte si quieres **enrutamiento**, para decirle que **sí** y ahorrar tiempo.

## Origen de datos de películas

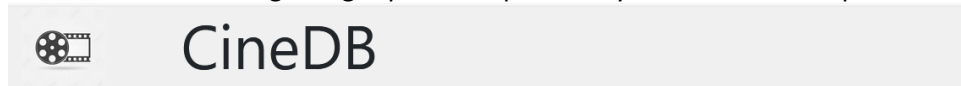
Los datos van a ser obtenidos de las siguientes fuentes por orden de prioridad, es decir, se intentará la fuente (1) y si hay algún obstáculo insalvable pasaremos a la fuente (2) y así sucesivamente:

1. API pública "OMDb API" en [www.omdbapi.com](http://www.omdbapi.com). Requiere una API key que se obtiene fácil e inmediatamente.
2. Si falla la obtención de la API Key os proporcionaré una.
3. Si falla el servicio entonces usaremos datos "caseros": una API REST de "My JSON Server" en [my-json-server.typicode.com](http://my-json-server.typicode.com) y que se alimentará de un fichero de datos alojado en un repositorio de *GitHub* del profesor.
4. Si también fallara el acceso a este último servicio entonces el profesor os proporcionará los datos para integrar directamente (*hard-coded*) en el código fuente.

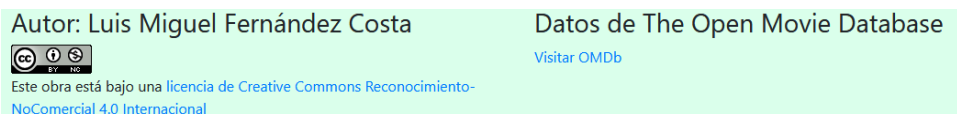
## Componente raíz

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

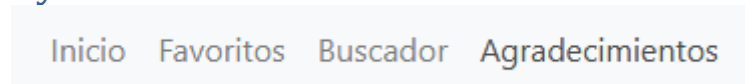
- **Encabezado.** Muestra la imagen logotipo de la aplicación y el nombre de la aplicación.



- **MenuNavegacion.** Ofrecerá la posibilidad de clicar en las secciones listadas en el apartado **Navegación y enrutamiento** más abajo.
- **Pie.** Muestra el nombre del autor o autora de la aplicación y cita a *OMDb* como fuente de los datos. Si finalmente no pudiera utilizarse *OMDb* como fuente de datos entonces se citará la fuente utilizada.



## Navegación y enrutamiento



Justo debajo del encabezado aparecerá una barra de navegación horizontal con los siguientes elementos:

- La sección **Inicio** mostrará el componente **Inicio**. Muestra un mensaje de bienvenida y explica qué se puede hacer con esta aplicación.
- La sección **Favoritos** mostrará el componente **ListadoFavoritos**. Muestra un listado de los elementos favoritos del usuario en formato cuadrícula. Para conseguir este listado nos apoyaremos en el siguiente componente:

- **Favorito.** Muestra un elemento favorito, con su carátula y su nombre y año. También 2 botones “Detalles” y “Borrar”, que sirven para ver detalles y borrar la película de favoritos, respectivamente.
- La sección **Buscador** mostrará el componente **BuscadorConsultas**. Permite escribir en un campo de texto para buscar por nombre y, de los resultados obtenidos, añadir a favoritos con un clic las películas deseadas.
- La sección **Agradecimientos** mostrará el componente **Agradecimientos**. Muestra los agradecimientos que consideremos oportunos.

Si a la URL base de nuestra aplicación se le añade alguno de los siguientes sufijos, se mostrará lo que se indica a continuación:

- “/home”: lleva a la sección **Inicio**.
- “/fav”: lleva a la sección **Favoritos**.
- “/query”: lleva a la sección **Buscador**.
- “/thank”: lleva a la sección **Agradecimientos**.
- “/detail:” y un ID válido: muestra el componente **Detalles** con los detalles de la película con el ID indicado.
- “/”: lleva a la sección **Inicio**.
- Cualquier otro sufijo: muestra el componente **Pagina404**.

## Para implementar los detalles completos de una película

Cuando el usuario haga clic en la carátula de una película favorita o presione su botón “Detalles”, se mostrará el componente **Detalles**. Este componente lista información detallada de una película, como carátula, título, año, duración en minutos, géneros, dirección, reparto, sinopsis, país, un enlace a dicha película en la web IMDB y su puntuación en dicha web.

## Para implementar una página 404

- **Pagina404.** Este componente muestra un mensaje de error al usuario indicando que ha intentado acceder a una página que no existe.

## Servicios

Será necesario crear los siguientes servicios:

- **BaseDatosCine.** Aporta las llamadas HTTP que se consideren necesarias para que el usuario se comunique con la API externa y pueda:
  - Buscar una lista de películas cuyo título incluya la palabra indicada.
  - Consultar por ID los detalles completos de una película.
- **Favoritos.** Es el lugar centralizado donde se almacena la lista de favoritos del usuario y contiene métodos para todas las operaciones que puedan ser necesarias sobre dicha lista de favoritos:
  - Obtener la lista de favoritos.
  - Añadir un favorito a la lista.
  - Borrar un favorito de la lista.
  - Vaciar la lista.
  - Comprobar si un elemento ya existe en la lista.

## Modelos de *templates* entregados por el profesor

Para ahorrar tiempo, algunas *templates* son entregadas con código HTML genérico que ayuda mucho, pero deben ser adaptadas previamente para funcionar bien:

- Encabezado
- Pie
- Inicio
- Agradecimientos
- Favorito
- Detalles

## Objetivos

Objetivo	OK
Los contenidos del componente raíz se ajustan a lo pedido	
La barra de navegación funcional	
La sección Favoritos lista las películas favoritas	
Las películas favoritas pueden ser borradas individualmente	
El servicio BaseDatosCine obtiene datos de la API externa mediante HTTP	
El buscador muestra resultados de búsqueda correctos	
Los resultados de búsqueda pueden ser añadidos a Favoritos	
Se pueden consultar los detalles completos de una película favorita	
El aspecto está cuidado y los errores controlados	

## 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
- 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:

- Definir una ruta parametrizada
- Detectar y manejar los posibles errores en tiempo de ejecución

## 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 todos los objetivos opcionales son satisfechos.