

# PRÁCTICAS API

## 1. Consumir API The Movie DB

Usando **The Movie Database API**, vamos a construir una página que muestre las **series de televisión** por género. Para usar la API tenemos el siguiente [endpoint](#).

Nos crearemos un menú lateral con, al menos, 5 géneros distintos de series. Al pinchar en un género haremos una petición Ajax a un fichero php que pida las series a la API que sean de ese género, las devuelva y se pinten en la web. Se puede hacer también directamente con Javascript.

Para cada serie **la información a mostrar** es: título, poster, media de los votos, sinopsis.

Cuando pinchamos en la serie aparecerá una **visualización ampliada** de la serie. En esta visualización ya decides tú lo que añadir (director, plataforma donde se ve, temporadas, los capítulos de cada temporada, etc.) Hay otro endpoint en la serie para mostrar información ampliada [link](#). Se deja a elección del alumno la cantidad de información a mostrar, pero cuanta más información mejor.

Hay que utilizar un diseño más atractivo que hasta la fecha.

## COMENTARIOS O CRÍTICAS

Debéis añadir una gestión de críticas o comentarios a las series, para que alguien que visite la web pueda poner comentarios y notas a las series. Esos comentarios se guardarán en BD (MongoDB). Tenéis que añadir botones/acciones que funcionen con Ajax:

- Ver comentarios
- Añadir comentario

La información a guardar para un comentario será: nick o nombre de la persona que lo hace, nota (1 a 10) y un texto. En la BD se debe guardar el id de la serie sobre la que se hace el comentario, puede ser el mismo de la API, de modo que queden relacionadas la serie y el comentario sobre esa serie.

## 2. Consumir API a elección

Desarrollar una pequeña app que consuma una API del siguiente listado (o de otro sitio):  
<https://github.com/public-apis/public-apis>

Se deja a elección del alumnado la API a usar, solo hay que usar sus verbos, no debemos grabar en BD nosotros nada. Construir una aplicación web sencilla sobre alguna de esas APIs

(tiempo en tu zona, anime, libros, recetas, videojuegos, covid-19, búsqueda empleo, música, noticias, Nasa, SpaceX, deportes, etc.)

### 3. Desarrollar una API REST sobre criptomonedas.

Desarrollar una API que permita consultar y crear criptomonedas. La información a guardar para una criptomoneda será: id, nombre, símbolo (3 letras), descripción breve, precio actual en euros, porcentaje de cambio de valor en 24h, capitalización (cuántas monedas hay en circulación). Verbos que admitirá nuestra API

- GET criptoc/ → devuelve las 50 primeras criptomonedas de la BD en formato Json.
- PUT criptoc/id/<id> → modifica una criptomoneda de la BD. Se añadirá en el Body en formato raw un JSON con todos los campos de la criptomoneda.
- PUT criptoc/up/id/<id> → sube en 0.1 el valor de una criptomoneda (id).
- PUT criptoc/down/id/<id> → baja en 0.1 el valor de una criptomoneda (id).
- POST criptoc/ → añade una nueva criptomoneda a la BD. Se incluirán inputs por cada atributo de la criptomoneda.
- GET criptoc/topvalue → muestra las primeras 10 criptomonedas ordenadas por mayor precio en euros.
- GET criptoc/id/<id> → muestra en formato Json la criptomoneda con el id dado.
- DELETE criptoc/id/<id> → borra la criptomoneda con ese id.

Cuando se inserte, elimine o modifique se devolverá un Json indicando si la acción ha podido ser realizada (success=true y status = 1) o no (success=false y status=2). Se pueden también devolver estados HTTP.

La Base de Datos será MongoDB y habrá que separar controlador y modelo (acceso a BD). Debéis crear un **índice** sobre el id de las criptomonedas (no sobre \_id, que ese es interno de MongoDB y ya se crea automáticamente).

No será necesario implementar sistema de autorización mediante api key (tipo OAuth o similares), pero si alguien quiere hacerlo se valorará muy **positivamente**.

Para comprobar que la API funciona correctamente podéis hacerlo de dos formas:

- Una página web sencilla que llame a la API. Se valorará más.
- Capturas de pantalla de POSTMAN probando todos los verbos.