

# PRÁCTICA 5.05 Asincronismo libre

## Normas de entrega

- En cuanto al **código**:
  - en la **presentación interna**, importan los **comentarios**, la claridad del código, la significación de los nombres elegidos; todo esto debe permitir considerar al programa como **autodocumentado**. No será necesario explicar que es un **if** un **for** pero sí su funcionalidad. Hay que comentar las cosas destacadas y, sobre todo, las **funciones** y **clases** empleadas. La ausencia de comentarios será penalizada,
  - en la **presentación externa**, importan las leyendas aclaratorias, información en pantalla y avisos de ejecución que favorezcan el uso de la aplicación,
  - todo el código debe estar situado dentro del evento **window.onload = () => {};** o a través del evento **document.addEventListener("DOMContentLoaded", () => {});**,
  - si no se especifica lo contrario, la información resultante del programa debe aparecer en la consola del navegador **console.log(información)**,
  - los ejercicios deben realizarse usando **JavaScript ES6** y usar el modo estricto (**use strict**) No se podrá utilizar **jQuery** ni cualquier otra biblioteca (si no se especifica lo contrario en el enunciado),
  - para el nombre de **variables**, **constantes** y **funciones** se utilizará *lowerCamelCase*,
  - para la asignación de eventos se utilizará **addEventListener()** indicando sus tres parámetros en su definición,
  - debes dividir tu código en **bibliotecas temáticas** de funciones y/o clases a partir de este ejercicio,
  - se usarán las funcionalidades **import** y **export** para crear bibliotecas de funciones temáticas a partir de esta práctica,
  - todo el código que sea susceptible de retrasar su ejecución debe escribirse de forma asíncrona utilizando **async/await**,
  - todas las funciones asíncronas deben tener control de errores.
- En cuanto a la **entrega** de los archivos que componen los ejercicios:
  - todos los ejercicios en **una carpeta** (creando las **subcarpetas** necesarias para documentación anexa como imágenes o estilos) cuyo nombre queda a discreción del discente,
  - el nombre de los ficheros necesarios para resolver el ejercicio será el número de ejercicio que contenga,
  - el código contendrá ejemplos de ejecución, si procede, y
  - la carpeta será **comprimida** en formato **ZIP** y será subida a **Aules** de forma puntual.

**NOTA:** Para la realización de los siguientes ejercicios se utilizarán API REST reales ya que este comportamiento es el que se utiliza en aplicaciones web. Si el ejercicio no especifica ninguna, puedes encontrar una API pública aquí: <https://github.com/public-apis/public-apis/> (tratar de no elegir la misma).

### Ejercicio 1 - Explotando una API

Realiza una pequeña aplicación web que consuma los datos de una API de tu elección y modifique dinámicamente el **DOM**.

La aplicación deberá, en función de la API seleccionada, mostrar los datos y permitir hacer selecciones o ampliaciones de información (del mismo modo que en la práctica anterior). El funcionamiento de tu aplicación dependerá de la API que elijas, así que deberás estudiarla con detalle.

Algunas ideas:

- Juego tipo trivial que cada vez que carga nos ofrece preguntas distintas sobre un tema concreto. El usuario rellena el formulario y obtiene su puntuación. Se utiliza la API **OpenTrivialDB** que encontrarás en [https://opentdb.com/api\\_config.php](https://opentdb.com/api_config.php).
- Enciclopedia *Pokemon* (no me hagáis esto), en la que se escribe el nombre de un pokemon y la web devuelve resultados con sus características y alguna imagen del mismo. Se usa la API **PokéAPI** que encontrarás en <https://pokeapi.co>.
- *Lyrics Search* que se utiliza para obtener la letra de una canción. Se busca por artista o por canción y cuando elije una canción se muestra la letra. Se utiliza la API **Lyrics.ovh** que encontrarás en <https://lyricsovh.docs.apiary.io/#reference/0/lyrics-of-a-song/search>.

Por supuesto puedes improvisar alguna otra aplicación con una de las API que encontrarás en el listado de *GitHub* on el cualquier otro repositorio que encuentres interesante.