

## PRÁCTICA 4.07 useEffect en React

### 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,
  - 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**. No se podrá utilizar ninguna biblioteca (si no se especifica lo contrario en el enunciado),
  - para el nombre de **variables**, **constantes** y **funciones** se utilizará *lowerCamelCase*,
  - el nombre de los componentes debe comenzar con letra **mayúscula**.
- En cuanto a la **entrega** de los archivos que componen los ejercicios de **React**:
  - entrega **la práctica** en un sólo proyecto (el nombre a tu discreción),
  - los componentes creados deben estar separados en carpetas (los creados en el Ejercicio1 dentro de una carpeta denominada **Ejercicio1**),
  - el código contendrá ejemplos de ejecución, si procede,
  - comprime la carpeta **src** junto con los ficheros **package.json** y **package-lock.json** en un fichero **ZIP** , y
  - sube a **Aules** el fichero comprimido.

### Ejercicio 1 - Cronómetro en **React**

Crea un cronómetro en un componente con las siguientes características:

- debe mostrar **segundos** y **centésimas** (o milésimas si te parece mejor),
- contiene tres botones: `empezar` , `pausar` y `reiniciar` que no es necesario aclarar para qué es cada uno,
- como siempre, la salida debe estar debidamente formateada.

Recuerda que si se usa `setInterval`, `setTimeout` o `addEventListener` en el objeto `window` siempre se debe utilizar una función de *clean up* en `useEffect`.