

PRÁCTICA 4.11 Formularios controlados 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**.
 - todos los formularios en **React** deben ser del tipo **controlados** (a través del **estado**). En caso contrario debe ser debidamente justificado.
- 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 - Lista de la compra con **React**

Crea una aplicación que lleve el registro de una lista de la compra con las siguientes características:

- los datos estarán almacenados en un objeto **JSON** (situado en una variable). Diseña su estructura para que permita almacenar un identificador de producto, su nombre, su descripción, su precio, las unidades que hay que comprar y el subtotal (total por línea de producto),
- crea un formulario que permita añadir productos a la lista de la compra. Dispondrá de un botón que, antes de realizar la inserción en el objeto **JSON**, comprobará si los campos poseen un valor válido,
- el listado de productos debe mostrarse en tiempo real. Reserva una parte de la aplicación para mostrar este listado de productos como ya sabes: debidamente formateado. Si la lista está vacía mostrará un mensaje indicando que no hay productos en la lista todavía,
- se debe mostrar el total de carrito (debidamente formateado) y actualizarse en cada cambio en la lista de la compra,
- al hacer clic sobre un producto de la lista se eliminará de ella previa confirmación del usuario,
- añade un botón que permita eliminar todo el listado.