Ejercicio 1

Usando el código desarrollado durante este tema (aplicación de gestión de usuarios), vamos a añadir la opción de editar usuarios. Es decir, además del enlace “Delete”, añadiremos “Edit”.

Lo que esta opción hará es lo siguiente:

1. Una vez pulsado, carga en el formulario los valores del cliente en cuestión (nombre, apellido y email). Además, el texto del botón de *submit* cambiará a “Edit”. Ejemplo al pulsar “Edit” para John Doe:

2. Tras hacer los cambios pertinentes en los datos (p.e. cambiamos “John” por “Jane”), si pulsamos el botón “Edit” se actualizará el usuario en la base de datos y habrá una redirección para recargar la página y mostrar la lista de usuarios actualizada. Además, la petición para buscar al usuario se hará mediante la API Fetch.

Ejercicio 2

Partiendo del ejercicio de la semana anterior (Drag&Drop), donde desarrollamos un formulario con su correspondiente verificación de campos, vamos a desarrollar un servidor en Express para la gestión de los archivos subidos. Podéis partir de vuestra implementación o de la solución de la semana pasada.

En concreto, si la verificación de campos es correcta, en lugar de enviar un mensaje por consola, vamos a subir estos ficheros al servidor (podemos convertir nuestros datos del formulario en un objeto que se puede enviar usando FormData). Para ello, usaremos una llamada Fetch API al servidor a /upload/files (revisad esto para saber cómo). En caso de que todo vaya bien, mostrará en el apartado de “Mensajes de estado” los siguientes datos:

Mensajes de estado

Datos del fichero: **3A0.jpg** Tipo: **image/jpeg** Tamaño: **1553742** bytes

Datos del fichero: **17010.jpg** Tipo: **image/jpeg** Tamaño: **979433** bytes

Datos del fichero: **BE.png** Tipo: **image/png** Tamaño: **367231** bytes

Resultados del formulario:

- Nombre: adrian
- Teléfono: 123456789
- Email: adrian@gmail.com
- Libro: Libro1
- Cantidad: 1
- Imágenes:

Además, si hacemos click en la imagen, esta se abrirá (hay que meter la imagen dentro de un enlace).

En el servidor, atenderemos la subida de archivos en el enrutamiento /upload/files. Usaremos multer para gestionar la subida de ficheros y guardarlos en /public/imgs/ (usad la opción *destination*). En la subida pondremos algunas condiciones:

- En total, la subida máxima será de 2MB. Para ello tendremos que usar la opción *limits*.

- Aceptaremos únicamente jpgs y pngs. Para ello se usa la opción *fileFilter*. En caso de no tener el formato correcto, mostrar un error por consola (en el servidor).

La respuesta de vuelta contendrá un JSON con una respuesta booleana de éxito y con los datos del formulario y las rutas de los ficheros. La respuesta booleana la imprimiremos por consola en el cliente y, si la consulta fue un éxito, imprimimos los datos como hemos mencionado previamente.