**Práctica 3: Sistema de Registro, Validación de usuarios y manipulación del DOM avanzada**

**Objetivo:**

Implementar un sistema de registro y validación de usuarios utilizando JavaScript. El objetivo principal es diseñar un formulario de registro de usuarios que permita a los usuarios ingresar sus datos personales y posteriormente validar la información introducida. Además, se busca proporcionar retroalimentación visual mediante iconos y mensajes de estado.

Aquí tienes una demo sobre como debería verse tu página web, atento a los detalles:

<https://www.loom.com/share/88f2fc5cecd24f7e9b8a0b8b9cd6d739?sid=d9acc16a-05f2-4574-b878-ad053d774fb8>

Tienes que conseguir lo siguiente utilizando solamente HTML (para el primer apartado) y JavaScript:

* Diseñar los campos del formulario faltantes en el código HTML
* Comprobar que todos los campos están rellenados
* Validar si los datos insertados pertenecen a uno de los registrados en el JSON
* Tiene que aparecer un ícono de éxito en caso en que el inicio de sesión se realice correctamente, y un icono de error en caso contrario
* Cuando el usuario no está registrado, espera 2 segundos y borra todos los campos
* En caso en que sean válidos, tiene que redirigirte a otra página web (BUSCAR COMO HACERLO EN EL TEMA DE BOM EN W3SCHOOLS)
  + Efectos
    - El color del botón Iniciar Sesión se pone de color verde cuando el mouse pasa por encima, y recupera el color original cuando quitamos el mouse
    - Se puede validar el formulario pulsando al botón o pulsando la tecla ***Enter***
    - Puedes navegar entre los ***inputs*** del formulario con las teclas ***ArrowDown*** y ***ArrowUp*. (Este es el más difícil jeje)**
    - Si al enviar los datos del formulario falta un campo, se tiene que enfocar dicho campo directamente.
    - Cada vez que un campo es enfocado, el color de los bordes se pone de color blanco y desaparece el placeholder. Cuando cambiamos de campo, recupera el color del borde anterior y el placeholder (en caso de no haber introducido texto)

**Pasos a seguir:**

**Diseño del Formulario en el HTML:**

Reutilizando el código HTML ya proporcionado, se tiene que crear un formulario de registro con campos para el nombre, apellidos, edad, sexo, email y contraseña.

**NO se puede modificar, o eliminar ninguna etiqueta ya existente**

**Implementación del procesamiento de fichero JSON:**

En el código JavaScript, se proporciona la variable de texto jsonRegistrados y se tiene que procesar para obtener una colección (o un array de objetos literales). Representando cada elemento de la colección a un usuario ya “registrado en nuestro programa”.

**Clase Registrado:**

Crear una clase llamada "Registrado" que represente a un usuario, con propiedades para el nombre, apellidos, edad, sexo, email y contraseña.

Crear la variable ***coleccion\_registrados*** como un array de instancias de la clase 'Registrado', cuyos valores de nombre, apellidos, edad, sexo, email y contraseña, se asignan en base al jsonRegistrados ya procesado

**Filtrado de Usuarios:**

Implementar la función ***filter\_user\_from\_collection***, que filtre los usuarios registrados según los valores ingresados en el formulario.

Esta función tiene que devolver ***coleccion\_registrados*** filtrado según el formulario.

(La función tendría que devolver una colección vacía en caso de que los datos introducidos no correspondan con ningún registrado)

**Inicialización de la etiqueta imagen:**

Crea la función ***initialize\_img*** que tiene que realizar lo siguiente al ser ejecutada:

* Crear etiqueta de imagen HTML, como un nodo hijo de la etiqueta ***<span>***
* Asignarle las siguientes propiedades
  + Ruta de la imagen: ./imgs/error.png
  + Texto alternativo: *check-icon*
  + Id: img
* Asignarle las siguientes propiedades de estilo:
  + Ancho: 20 píxeles
  + Alto: 20 píxeles
  + Ocultación de la etiqueta

Una vez se defina esta función, la ejecutamos para crearla en el HTML nada más abramos la página web

**Comprobación del rellenado de los campos**

Implementa la función ***comprobarCamposRellenos***. Esta función comprueba en cada campo si se han metido datos correctamente.

La función deberá realizar lo siguiente:

* Tiene que devolver ***true***, en caso de que todos los campos estén rellenados. En caso contrario, tiene que devolver ***false***
* Para cada campo, en el caso en que esté vacío:
  + Cambiar el contenido del párrafo ***errorcase*** del HTML con el texto: “El campo \*\*{campo}\*\* no puede estar vacío”
  + El campo vacío se tiene que enfocar (se tiene que seleccionar y poner el cursor directamente sobre el campo vacio)

**Borrar los campos del formulario**

Implementar la función ***borrarCampos()*** que borra el contenido que se encuentra en cada input del formulario

**Validación de Campos:**

Función principal que se ejecuta cuando se pulsa el botón *Iniciar Sesión* o se pulsa la tecla *Enter* con el atributo ***onclick***de la etiqueta ***<button>***

Los pasos a realizar son:

1. Comprobar si ***comprobarCamposRellenos()*** devuelve ***true*** o devuelve ***false***.
2. En caso de true, ejecutar el resto
3. Obtener colección filtrada con ***filter\_user\_from\_collection()***
4. Si los datos del formulario no corresponden con ningún registrado
   1. Mostrar el icono de la imagen ./imgs/error.png
   2. Cambiar el contenido HTML del párrafo ***errorcase*** por: “Usuario no registrado”
   3. Esperar 2 segundos y borrar los campos del formulario y volver a ocultar la imagen
5. Si los datos del formulario corresponen con un registrado:
   1. Mostrar el icono de la imagen ./imgs/success.png
   2. Crear un párrafo como último nodo HTML hijo de ***<form>*** cuyo contenido sea: “Usuario registrado correctamente”
   3. Redirigir al HTML “src/page\_successful.html” tras esperar 3 segundos (buscar métodos en BOM que te permitan realizar esta acción)

**Eventos adicionales de validación**

Crea el evento necesario para que, cuando pulsemos la tecla Enter, automáticamente hagamos un ***submit*** de los datos y también se ejecute el método ***validar()***

Si eres valiente, en este mismo evento añade una funcionalidad que te permita moverte entre inputs de texto del formulario con las teclas ArrowDown (moverse al input de abajo) y ArrowUp (moverse al input de arriba)

**Eventos adicionales del botón**

Añade los eventos necesarios al botón HTML de *Iniciar Sesión* para:

* Cambiar el color al pasar el ratón por encima a: “#056d28”
* Cambiar el color al original cuando quitamos el ratón: “#083eb2”

**Eventos adicionales entradas formulario**

Añade los eventos necesarios en cada input del formulario de modo que:

* Cuando enfocamos una entrada de texto, el color de los bordes de la entrada sea “#083eb2” y desaparezca el placeholder
* Cuando salimos de esa entrada de texto, se recupera el color original “#1f53c5” y regresa el placeholder anterior, “Ingrese su []”