

UT5 Práctica 4

EJERCICIO 1

JS

EJERCICIO 2

JS

UT5_Práctica_4

EJERCICIO 1

```
1 window.onload = function() {
2   // Selección del formulario y botón
3   const formulario = document.getElementById("formulario");
4   const submitBtn = document.getElementById("submitBtn");
5   const salida = document.getElementById("salida");
6
7   // Función para validar campos
8   function validarFormulario() {
9     let errores = [];
10
11     // Validar la fecha
12     const fecha = document.getElementById("date");
13     if (!fecha.value) {
14       errores.push("La fecha es obligatoria.");
15     }
16
17     // Validar el rango
18     const rango = document.getElementById("range");
19     if (!rango.value || rango.value < 1 || rango.value > 10) {
20       errores.push("El nivel de satisfacción debe estar entre 1 y 10.");
21     }
22
23     // Validar la contraseña
24     const password = document.getElementById("password");
25     if (!password.value || password.value.length < 8) {
26       errores.push("La contraseña debe tener al menos 8 caracteres.");
27     }
28
29     // Validar el número de teléfono
30     const telefono = document.getElementById("tel");
31     const telefonoRegex = /^(\d{3})\s\d{2}\s\d{2}\s\d{2}$/;
32     if (!telefono.value || !telefonoRegex.test(telefono.value)) {
33       errores.push("El número de teléfono debe tener el formato XXX XX XX XX.");
34     }
35
36     // Validar la URL
37     const url = document.getElementById("url");
38     try {
39       new URL(url.value); // Verifica si la URL es válida
40     } catch (e) {
41       errores.push("El sitio web debe tener un formato válido (https://example.com).");
42     }
43
44     // Mostrar errores si existen
45     if (errores.length > 0) {
46       alert("Errores en el formulario:\n" + errores.join("\n"));
47       return false;
48     }
49
50     return true;
51   }
52
53   // Manejar el evento click del botón de enviar
54   submitBtn.addEventListener("click", () => {
55     if (validarFormulario()) {
56       // Si pasa las validaciones, muestra los datos en el pre
57       const datos = {
58         Fecha: document.getElementById("date").value,
59         "Nivel de satisfacción": document.getElementById("range").value,
60         Contraseña: document.getElementById("password").value,
61         "Número de teléfono": document.getElementById("tel").value,
62         "sitio web": document.getElementById("url").value,
63       };
64
65       salida.textContent = JSON.stringify(datos, null, 2);
66       alert("Formulario enviado con éxito.");
67     }
68   });
69 }
70
71 // Pruebas de escritorio
72 // 1. Prueba de campo vacío
73 // 2. Prueba de campo incorrecto
74 // 3. Prueba de campo correcto
75 // 4. Prueba de campo obligatorio
76 // 5. Prueba de campo con formato incorrecto
77 // 6. Prueba de campo con formato correcto
78 // 7. Prueba de campo con formato incorrecto
79 // 8. Prueba de campo con formato correcto
80 // 9. Prueba de campo con formato incorrecto
81 // 10. Prueba de campo con formato correcto
```

```
1 window.onload = function() {
2   // Selección del formulario y botón
3   const formulario = document.getElementById("formulario");
4   const submitBtn = document.getElementById("submitBtn");
5   const salida = document.getElementById("salida");
6
7   // Función para validar campos
8   function validarFormulario() {
9     let errores = [];
10
11     // Validar la fecha
12     const fecha = document.getElementById("date");
13     if (!fecha.value) {
14       errores.push("La fecha es obligatoria.");
15     }
16
17     // Validar el rango
18     const rango = document.getElementById("range");
19     if (!rango.value || rango.value < 1 || rango.value > 10) {
20       errores.push("El nivel de satisfacción debe estar entre 1 y 10.");
21     }
22
23     // Validar la contraseña
24     const password = document.getElementById("password");
25     if (!password.value || password.value.length < 8) {
26       errores.push("La contraseña debe tener al menos 8 caracteres.");
27     }
28
29     // Validar el número de teléfono
30     const telefono = document.getElementById("tel");
31     const telefonoRegex = /^(\d{3})\s\d{2}\s\d{2}\s\d{2}$/;
32     if (!telefono.value || !telefonoRegex.test(telefono.value)) {
33       errores.push("El número de teléfono debe tener el formato XXX XX XX XX.");
34     }
35
36     // Validar la URL
37     const url = document.getElementById("url");
38     try {
39       new URL(url.value); // Verifica si la URL es válida
40     } catch (e) {
41       errores.push("El sitio web debe tener un formato válido (https://example.com).");
42     }
43
44     // Mostrar errores si existen
45     if (errores.length > 0) {
46       alert("Errores en el formulario:\n" + errores.join("\n"));
47       return false;
48     }
49
50     return true;
51   }
52
53   // Manejar el evento click del botón de enviar
54   submitBtn.addEventListener("click", () => {
55     if (validarFormulario()) {
56       // Si pasa las validaciones, muestra los datos en el pre
57       const datos = {
58         Fecha: document.getElementById("date").value,
59         "Nivel de satisfacción": document.getElementById("range").value,
60         Contraseña: document.getElementById("password").value,
61         "Número de teléfono": document.getElementById("tel").value,
62         "sitio web": document.getElementById("url").value,
63       };
64
65       salida.textContent = JSON.stringify(datos, null, 2);
66       alert("Formulario enviado con éxito.");
67     }
68   });
69 }
70
71 // Pruebas de escritorio
72 // 1. Prueba de campo vacío
73 // 2. Prueba de campo incorrecto
74 // 3. Prueba de campo correcto
75 // 4. Prueba de campo obligatorio
76 // 5. Prueba de campo con formato incorrecto
77 // 6. Prueba de campo con formato correcto
78 // 7. Prueba de campo con formato incorrecto
79 // 8. Prueba de campo con formato correcto
80 // 9. Prueba de campo con formato incorrecto
81 // 10. Prueba de campo con formato correcto
```

Configuración inicial:

- Al cargar la página (`window.onload`), se obtienen referencias al formulario, al botón de envío (`submitBtn`) y al elemento donde se mostrarán los datos ingresados (`salida`).

Validación del formulario (`validarFormulario`):

- **Fecha:** Se verifica que el campo de fecha no esté vacío.
- **Rango:** Se asegura que el valor esté entre 1 y 10.
- **Contraseña:** Debe tener al menos 8 caracteres.
- **Teléfono:** Se valida el formato específico (`XXX XX XX XX`) mediante una expresión regular.
- **URL:** Se verifica si es una URL válida usando el objeto `URL`.
- Si hay errores, se muestra un mensaje de alerta con una lista de los problemas encontrados.

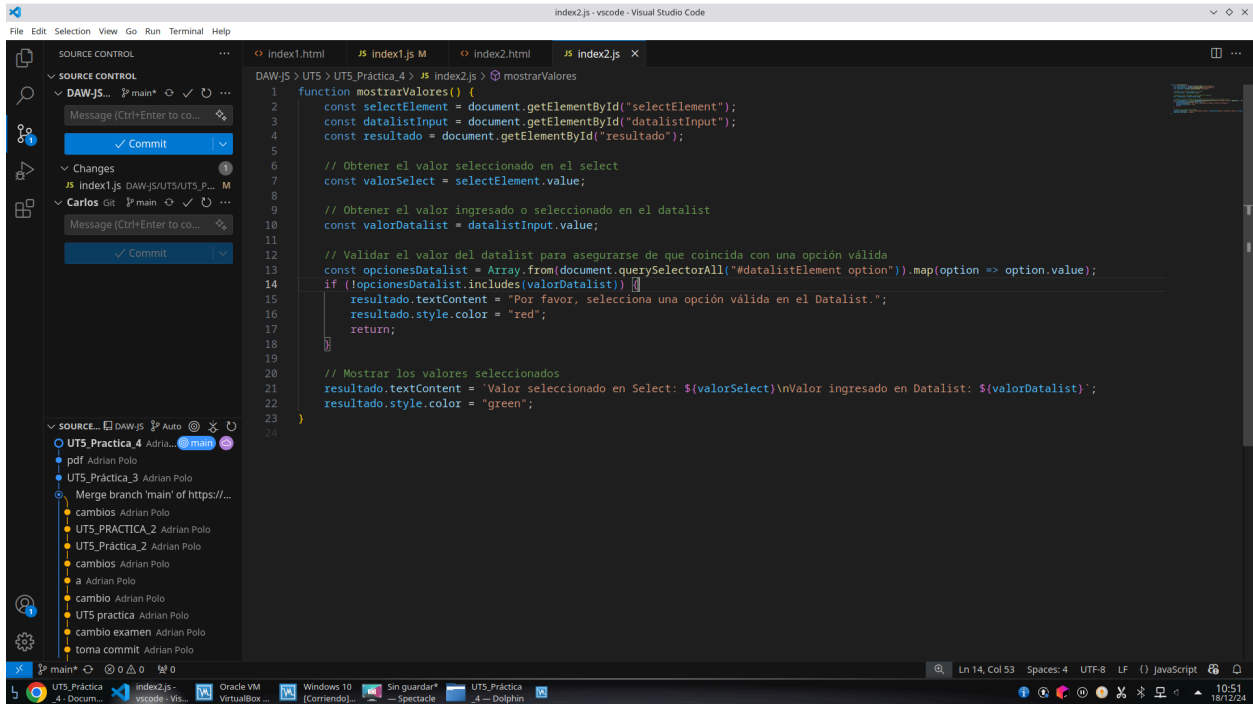
Gestión del envío:

- Se asocia un evento al botón de envío. Al hacer clic:
 - Se llama a `validarFormulario`.
 - Si las validaciones pasan, se recogen los valores de los campos del formulario y se muestran en formato JSON en el elemento `salida`.
 - Se muestra un mensaje de éxito con una alerta.

Manejo de errores:

- Si las validaciones fallan, el envío del formulario se detiene y se alerta al usuario con los errores específicos.

EJERCICIO 2



Obtención de referencias:

- Se obtienen referencias a:
 - El elemento `<select>` (`selectElement`) para capturar la opción seleccionada.
 - El campo de entrada está asociado al `<datalist>` (`datalistInput`).
 - El elemento donde se mostrarán los resultados (`resultado`).

Captura de valores:

- Se obtiene el valor seleccionado del `<select>` mediante su propiedad `value`.
- Se captura el valor ingresado o seleccionado en el campo `<input>` asociado al `<datalist>`.

Validación del `<datalist>`:

- Se crea un array con las opciones válidas del `<datalist>` usando `querySelectorAll` y `map`.
- Se verifica si el valor ingresado en el `<datalist>` coincide con alguna de las opciones disponibles.
- Si el valor no es válido, se muestra un mensaje de error en el elemento `resultado` con un color rojo.

Visualización de los valores:

- Si las validaciones son exitosas, se muestra en el elemento `resultado`:
 - El valor seleccionado en el `<select>`.
 - El valor ingresado o seleccionado en el `<datalist>`.
- El mensaje se estiliza con un color verde.