

Relación F

Formularios, validación y controles de interfaz

Nota

Esta relación trabaja el acceso y manejo de los distintos tipos de controles de formulario HTML, la lectura y escritura de sus valores, y la validación de datos del lado del cliente mediante JavaScript. Se introducen también patrones básicos de interfaz reactiva.

Ejercicio 1. Cree un formulario HTML completo que incluya al menos los siguientes controles: `input[text]`, `input[number]`, `input[email]`, `input[checkbox]` (varios), grupo de `input[radio]`, `textarea`, `select` simple, `select` múltiple y un botón de envío. Asígnele un `name` al formulario. Mediante JavaScript (sin enviar el formulario), al pulsar el botón, recoja y muestre por consola el valor de cada control, gestionando correctamente los checkboxes (array de valores marcados), los radios (valor seleccionado) y el select múltiple (array de opciones seleccionadas).

Ejercicio 2. Implemente una función de validación del formulario anterior que compruebe, al intentar enviar:

- Que el campo de texto no esté vacío.
- Que el número introducido esté entre 0 y 100.
- Que el email tenga formato válido (use una expresión regular básica).
- Que se haya marcado al menos un checkbox.
- Que se haya seleccionado una opción de radio.

Si hay errores, muéstrelos en el propio formulario (junto a cada campo, no en un `alert`), añadiendo un elemento de error que se crea o se elimina dinámicamente. Si todo es válido, muestre los datos en consola y un `alert` de confirmación.

Ejercicio 3. Cree un formulario de registro de estudiante con los campos: nombre, apellidos, DNI, fecha de nacimiento, grado (desplegable con varias opciones), semestre (radio: primero/segundo) y observaciones (textarea). Implemente validación en tiempo real (evento `input` o `blur`) que:

- Verifique que el nombre y apellidos no contengan números.
- Verifique el formato del DNI español (8 dígitos + letra).
- Calcule y muestre la edad a partir de la fecha de nacimiento.
- Limite las observaciones a 200 caracteres, mostrando un contador regresivo.

Ejercicio 4. Cree un control `select` desplegable vacío. Mediante JavaScript (no en el HTML), genere dinámicamente las opciones a partir de un array de objetos. Por ejemplo, un array de distribuciones estadísticas con campos `id`, `nombre` y `descripcion`. Al cambiar la selección, muestre en un `<div>` adyacente la descripción de la distribución seleccionada.

Ejercicio 5. Implemente un control de autocompletado básico:

- Un campo de texto y, debajo de él, un `` inicialmente oculto.

- Un array de al menos 15 términos estadísticos o matemáticos como fuente de datos.
- Al escribir en el campo (evento `input`), filtre los términos que contienen el texto escrito e inserte los resultados como `` en la lista desplegable.
- Al hacer clic en un ``, complete el campo de texto con ese valor y oculte la lista.
- Si no hay coincidencias, muestre un `` con el texto “Sin resultados”.

Ejercicio 6. Cree un formulario con tres campos numéricos: media (μ), desviación típica (σ) y un valor x . Al pulsar un botón, calcule y muestre en la página:

- La puntuación típica (z-score): $z = (x - \mu) / \sigma$.
- Una interpretación textual del z-score (muy por debajo de la media, por debajo, en la media, por encima, muy por encima).

Valide que $\sigma > 0$ antes de calcular. Muestre los resultados en un `<div>` de resultados debajo del formulario.

Ejercicio 7. Cree un formulario para introducir una lista de datos (campo de texto con valores separados por comas) y un botón “Calcular”. Al pulsar el botón:

- Parsee los datos introducidos, validando que son números.
- Muestre debajo los estadísticos básicos: n , mínimo, máximo, media, mediana y desviación típica.
- Genere una representación visual simple del histograma usando barras HTML (un `<div>` con `width` proporcional a la frecuencia).

Ejercicio 8. Cree una calculadora estadística con un `select` que permita elegir entre varias operaciones (media, varianza, mediana, percentil p , etc.) y un `textarea` para introducir los datos. Al seleccionar una operación, muestre debajo del `select` los parámetros adicionales necesarios (por ejemplo, para el percentil, un campo numérico para introducir p). Valide la entrada y muestre el resultado. Use `style.display` para mostrar u ocultar secciones del formulario según la operación elegida.

Ejercicio 9. Implemente un formulario de encuesta de satisfacción con al menos 5 preguntas, cada una respondida con un `input[range]` de 1 a 5. Muestre en tiempo real, según el usuario mueve los sliders:

- El valor numérico actual de cada slider junto a él.
- La puntuación media actual de todas las respuestas.
- Una barra de progreso visual (un `<div>` con ancho proporcional) que represente la media.

Ejercicio 10. Cree un formulario de alta de asignatura con los campos: nombre, código (formato: 2 letras + 4 dígitos), créditos (entero entre 1 y 12), tipo (obligatoria / optativa / TFG) y departamento (desplegable). Implemente validación completa y, si el formulario es válido, añada la asignatura a un array en memoria y actualice una tabla HTML que muestre todas las asignaturas añadidas. Incluya un botón para eliminar filas de la tabla (y del array) y otro para limpiar el formulario.

Ejercicio 11. Cree un control de paginación genérico. Dado un array de al menos 30 elementos (puede ser una lista de términos estadísticos), cree una interfaz que:

- Muestre solo n elementos por página (con n configurable mediante un `select` con opciones 5, 10, 20).
- Tenga botones “Anterior” y “Siguiente” que naveguen entre páginas.
- Muestre el número de página actual y el total.
- Deshabilite el botón “Anterior” en la primera página y “Siguiente” en la última.

Ejercicio 12. Cree un formulario con un campo de contraseña y un botón “Mostrar/ocultar contraseña”. Además, implemente un indicador de fortaleza que se actualice en tiempo real y evalúe: longitud mínima de 8 caracteres, presencia de mayúsculas, minúsculas, dígitos y caracteres especiales. Muestre la fortaleza como una barra de color (rojo / amarillo / verde) y un texto descriptivo (débil / media / fuerte).

Ejercicio 13. Cree una página con un área de texto en la que el usuario puede escribir libremente. Implemente las siguientes funcionalidades mediante botones:

- Contar palabras únicas en el texto (ignorando mayúsculas y signos de puntuación).
- Resaltar en un `<div>` de previsualización (con `innerHTML`) todas las apariciones de una palabra buscada.
- Reemplazar todas las apariciones de una cadena por otra.
- Exportar el texto como archivo descargable (`.txt`) usando `Blob` y `URL.createObjectURL`