

## VALIDACIÓN DE FORMULARIO CON FUNCIONES

### 1. Contexto de la actividad

En esta práctica vas a desarrollar la **lógica completa de validación de un formulario web** utilizando JavaScript y funciones.

### 2. Material proporcionado

Se te facilita una carpeta de proyecto con la siguiente estructura:

- index.html (estructura del formulario)
- style.css (diseño completo)

Los archivos HTML y CSS ya están terminados y no deben modificarse, salvo indicación expresa.

Tu trabajo consistirá exclusivamente en programar el archivo: js/app.js

### 3. Objetivo final

Al finalizar la práctica, el formulario deberá:

- Validar cada campo de forma individual.
- Mostrar errores claros junto a cada campo.
- Indicar el estado general del formulario.
- Generar un resumen de errores.
- Permitir el envío solo si todos los datos son correctos.
- Reiniciar correctamente los mensajes al usar el botón Reset.
- El formulario no debe enviarse si existen errores.

### 4. Campos que deben validarse

Debes validar correctamente los siguientes campos:

Datos personales

- Nombre
- Apellidos
- Email
- Teléfono
- Fecha de nacimiento
- Provincia
- Seguridad

- Contraseña
- Repetición de contraseña

Otros

- Observaciones (opcional)
- Aceptación de términos

## 5. Reglas de validación

Debes aplicar, como mínimo, las siguientes reglas:

Nombre

- Entre 2 y 30 caracteres
- Solo letras y espacios

Apellidos

- Entre 2 y 60 caracteres

Email

- Formato válido (usuario@dominio.ext)

Teléfono

- Exactamente 9 dígitos
- Solo números

Fecha de nacimiento

- El usuario debe ser mayor de 18 años

Provincia

- No puede estar vacía

Contraseña

- Mínimo 8 caracteres
- Al menos:
  - 1 mayúscula
  - 1 minúscula
  - 1 número
  - 1 símbolo

Repetir contraseña

- Debe coincidir con la contraseña

## Observaciones

- Máximo 200 caracteres

## Términos

- Deben estar aceptados obligatoriamente

## 6. Funciones obligatorias

Debes crear, como mínimo, las siguientes funciones en tu archivo app.js.

### 6.1 Funciones de lectura

leerCampoTexto(id): Devuelve el valor limpio (sin espacios) de un input de texto.

leerCheckbox(id): Devuelve true o false según esté marcado.

leerSelect(id): Devuelve el valor seleccionado.

### 6.2 Funciones de validación por campo

validarNombre()

validarApellidos()

validarEmail()

validarTelefono()

validarFecha()

validarProvincia()

validarPassword()

validarPassword2()

validarObservaciones()

validarTerminos()

Cada función debe:

- Comprobar su campo.
- Mostrar su propio mensaje de error.
- Devolver true o false.

### 6.3 Funciones de interfaz

mostrarError(idError, mensaje)

limpiarError(idError)

actualizarEstadoGeneral(texto, correcto)

Estas funciones controlan la presentación de errores y estado.

#### 6.4 Funciones generales

validarFormulario(): Llama a todas las validaciones. Devuelve true solo si todo es correcto.

generarResumenErrores(): Muestra una lista con los errores activos.

limpiarFormulario(): Limpia mensajes y estados.

### 7. Orden recomendado de programación

Debes seguir este orden para evitar errores:

- Captura de elementos del DOM
- Funciones de lectura
- Funciones mostrar/limpiar errores
- Validaciones individuales
- Validación general
- Resumen
- Eventos de botones
- No programes todas las funciones a la vez.
- Ve comprobando cada bloque antes de continuar.

### 8. Gestión de eventos

Debes programar, como mínimo:

- Evento submit del formulario
- Evento click del botón reset
- Eventos blur o input en los campos (opcional)
- El submit debe cancelarse si hay errores.

### 9. Ampliación (opcional)

Para subir nota:

- Validación en tiempo real
- Barra de progreso
- Indicador visual por campo