Logotipo

El contenido generado por IA puede ser incorrecto.

Santiago Yosa González

**Ingeniero Sistemas de Información**



Logotipo

El contenido generado por IA puede ser incorrecto.}

**Etiquetas de formulario**

• form, input, label, button, select, textarea

**Atributos de las etiquetas de formulario**

• type, name, value, min, max entre otros.

**Unidad I**

**Maquetación de páginas Web**

**Situación - Analogía Ventanilla del Banco**

Imagina que vas a un banco.

1. **El Formulario (<form>):** Es el sobre o la carpeta donde metes todos tus documentos. Tiene una etiqueta que dice a qué departamento va (action) y cómo se debe entregar (correo normal o certificado - method).
2. **Los Campos (<input>, <textarea>):** Son los espacios en blanco en el papel. Algunos son pequeños (nombre), otros grandes (comentarios).
3. **Las Etiquetas (<label>):** Son las instrucciones impresas al lado del espacio en blanco que dicen "Nombre aquí". Sin ellas, verías un espacio en blanco y no sabrías qué escribir.

**El Contenedor Padre: <form>**

Es la etiqueta que envuelve todos los controles. Sin ella, los inputs son solo dibujos en la pantalla sin funcionalidad de envío.

* **Atributo action:** Especifica la URL a la que se enviarán los datos. (En este curso introductorio, usualmente será # o una página de gracias, ya que no tenemos Backend aún).
* **Atributo method:** Define cómo viajan los datos HTTP.
  + **GET:** Los datos viajan visibles en la URL (ej: busqueda.php?q=gatos). Rápido, cacheable, **inseguro** para contraseñas.
  + **POST:** Los datos viajan en el cuerpo de la petición HTTP, invisibles al usuario. **Seguro** y capaz de enviar grandes cantidades de datos.

<form action="/procesar-datos" method="POST">

**La Etiqueta Vital: <label>**

Jamás se debe crear un input sin su etiqueta. Existen dos formas de vincularlas:

1. **Vinculación Explícita (Estándar de Oro):** Usando el atributo **for** en el label y el atributo **id** en el input. Deben ser idénticos.
   * *Ventaja:* Aumenta el área de clic (clic en el texto activa el input) y es fundamental para lectores de pantalla.
2. **Vinculación Implícita:** El input va dentro del label.

**El Camaleón: <input> y sus Tipos**

La etiqueta <input> es autocerrada y cambia drásticamente según su atributo type.

|  |  |
| --- | --- |
| Tipo (type) | Uso y Comportamiento |
| text | Texto de una sola línea. Valor por defecto. |
| password | Oculta los caracteres con puntos o asteriscos. Ojo: No cifra los datos, solo los oculta visualmente. |
| email | Valida que el texto tenga formato de correo y cambia el teclado en móviles (añade @). |
| number | Solo permite números. Habilita flechas de incremento/decremento. |
| date | Despliega un calendario nativo del sistema operativo. |
| radio | Selección única entre varias opciones. Deben compartir el mismo name para ser excluyentes. |
| checkbox | Selección múltiple. Funcionan independientemente. |
| file | Permite subir archivos del ordenador. |
| hidden | Invisible para el usuario, pero envía datos al servidor (ej: ID de usuario, tokens de seguridad). |

**<laber> type**

<label for="nombre">Nombre de Usuario:</label>

<input type="text" id="nombre" name="username" placeholder="Ej: JuanPerez">

**type="radio" (Selección Única)**

<p>Selecciona tu membresía:</p>

<input type="radio" id="plan\_gratis" name="plan" value="gratis">

<label for="plan\_gratis">Plan Gratuito</label>

<input type="radio" id="plan\_pro" name="plan" value="pro">

<label for="plan\_pro">Plan Pro</label>

**type="checkbox" (Selección Múltiple)**

<p>Intereses (Marca los que apliquen):</p>

<div>

<input type="checkbox" id="interes\_html" name="intereses" value="html">

<label for="interes\_html">Aprender HTML</label>

</div>

<div>

<input type="checkbox" id="interes\_css" name="intereses" value="css">

<label for="interes\_css">Aprender CSS</label>

</div>

**type="file" (Subida de Archivos)**

<label for="hoja\_vida">Sube tu CV (PDF):</label>

<input type="file" id="hoja\_vida" name="documento\_cv" accept=".pdf">

**type="hidden" (Campo Oculto)**

**Uso típico:** Enviar un ID de usuario, un token de seguridad (CSRF) o un código de referido sin que el usuario lo vea o lo modifique.

<input type="hidden" name="user\_id" value="54892">

<input type="hidden" name="token\_seguridad" value="abc-123-xyz">

**Lo que sucede:** **Nada visual.** El usuario no ve nada en la página, pero cuando le da clic al botón "Enviar", esos datos (user\_id y token) viajan al servidor junto con el resto del formulario

**Controles Avanzados: <textarea>, <select>, <button>**

**<textarea>:** Para texto multilínea. **No usa el atributo value**. El texto por defecto va *entre* las etiquetas de apertura y cierre.

<textarea id="alergias" name="lista\_alergias" rows="4" cols="50"></textarea>

**<select> y <option>:** Crea listas desplegables.

* <select>: El contenedor.
* <option>: Cada ítem. Tiene un atributo value (lo que se envía al servidor) y el texto visible (lo que ve el usuario).

<select id="sangre" name="tipo\_sangre" required>

<option value="" disabled selected>Seleccione una opción</option>

<option value="A+">A+</option>

<option value="A-">A-</option>

<option value="O+">O+</option>

</select>

**<button>:** Siempre prefiera <button type="submit"> sobre <input type="submit">. Permite tener iconos o HTML dentro del botón.

<button type="submit">Registrar Paciente</button>

Texto

El contenido generado por IA puede ser incorrecto.

**Atributos de Control y Validación**

Estos atributos reducen la necesidad de JavaScript para validaciones básicas.

* **name**: **EL MÁS IMPORTANTE**. Es el nombre de la variable que recibirá el servidor. Sin name, el dato no viaja.
* **id**: Identificador único para CSS y vinculación con <label>.
* **placeholder**: Texto fantasma de ayuda (se borra al escribir). **No es un sustituto del label**.
* **required**: Impide enviar el formulario si el campo está vacío.
* **readonly**: El usuario ve el dato pero no puede modificarlo (sí se envía).
* **disabled**: El usuario no puede interactuar y **el dato NO se envía**.
* **min / max**: Para números y fechas.
* **step**: Intervalos de números (ej: step="0.5").
* **pattern**: Acepta Expresiones Regulares (Regex) para validaciones complejas.

**Preguntas**

**¿Cuál es la diferencia técnica entre name e id?**

El id es para el **Navegador** (CSS, JavaScript y Labels). El name es para el **Servidor** (Backend). Si envías un formulario sin name, el servidor recibe un sobre vacío. Si usas el mismo id dos veces, rompes el HTML. Si usas el mismo name (en radios), agrupas opciones.

**¿Por qué mi <textarea> tiene espacios en blanco al cargar la página?**

A diferencia de los inputs, el <textarea> guarda todo lo que esté entre la etiqueta de apertura y cierre, **incluyendo saltos de línea y tabulaciones del código**.

<textarea>Texto</textarea> (todo en una línea).

**Tienes un input de tipo number. ¿El servidor recibe un número matemático o un texto que parece número?**

En HTML/HTTP, **todo viaja como texto (string)**. Aunque restrinjas la entrada a números visualmente, el backend recibirá un string "50" que deberá convertir

**Ejemplos Demostrativos**

1. **Demo Visual de Tipos:** Crear un archivo simple con todos los tipos de input (color, range, date) ¿Qué cambia si lo abrimos en un celular?
2. **El "Hack" del Label:** Si vinculamos un texto, por ejemplo "Acepto términos y condiciones" (que suele ser pequeño) con un checkbox, ¿Se marca el checkbox automáticamente si damos clic en el texto?

**Práctica Acompañada**

**Práctica 1: Formulario de Registro de Paciente**

Vamos a crear el formulario de ingreso de una clínica. Sigue las instrucciones para asegurar que cada dato tenga el tipo correcto.

**Paso a Paso:**

1. Crea la estructura <form> con método POST.
2. Añade un campo para **Nombre Completo** (text, requerido).
3. Añade un campo para **Fecha de Nacimiento** (usa el tipo date y pon un max para que no puedan elegir fechas futuras).
4. Añade un selector (select) para el **Tipo de Sangre** (A+, A-, O+, etc.).
5. Añade un área de texto (textarea) para **Alergias** (opcional).
6. Añade un botón de envío.

**Solución sugerida Práctica 1:**

<form action="#" method="POST">

<h2>Registro de Paciente</h2>

<div>

<label for="nombre">Nombre Completo:</label>

<input type="text" id="nombre" name="nombre\_paciente" required placeholder="Ej: Pepito Pérez">

</div>

<div>

<label for="nacimiento">Fecha de Nacimiento:</label>

<input type="date" id="nacimiento" name="fecha\_nacimiento" max="2025-12-03">

</div>

<div>

<label for="sangre">Tipo de Sangre:</label>

<select id="sangre" name="tipo\_sangre" required>

<option value="" disabled selected>Seleccione una opción</option>

<option value="A+">A+</option>

<option value="A-">A-</option>

<option value="O+">O+</option>

<option value="O-">O-</option>

</select>

</div>

<div>

<label for="alergias">Alergias (Describa si tiene):</label>

<br>

<textarea id="alergias" name="lista\_alergias" rows="4" cols="50"></textarea>

</div>

<button type="submit">Registrar Paciente</button>

</form>

**Práctica 2: Refuerzo Independiente**

Ahora crea un formulario para comprar una camiseta personalizada.

**Paso a Paso a Seguir:**

1. Crea un input number para la **Cantidad**. No puedes comprar menos de 1 ni más de 10.
2. Crea un grupo de radio buttons para la **Talla** (S, M, L). Recuerda: para que sean excluyentes (solo puedas elegir una), deben tener el mismo atributo name.
3. Crea un input color para elegir el **Color** de la tela.
4. Crea un input email para enviar el recibo.
5. **Desafío:** Añade un checkbox que diga "Es para regalo" (valor opcional).

**Ejercicio Independiente**

**Formulario de Solicitud de Empleo**

Debemos crear un archivo empleo.html que integre todo lo visto. El formulario debe permitir a un usuario postularse a una vacante.

**Requisitos:**

1. **Datos Personales:** Nombre, Teléfono (tel), Email.
2. **Perfil:** Un campo URL para su perfil de LinkedIn (type="url").
3. **Experiencia:** Un selector (select) que pregunte años de experiencia (0-1, 2-5, +5).
4. **Hoja de vida:** Un campo para subir archivo (file) que solo acepte PDFs (investigar atributo accept).
5. **Validación:** El nombre y el archivo deben ser obligatorios.

**Criterios de Logro:**

* Uso correcto del atributo accept=".pdf" en el input file.
* Validación correcta del campo URL (si el usuario escribe texto normal, el navegador debe mostrar error).
* Correcta asociación Label-Input en todos los campos.

**¿Debería saber algo más?**

**Agrupación Semántica (<fieldset> y <legend>):**

* No se deben usar <div> o <p> para agrupar radios y checkboxes. La forma correcta y accesible es usar <fieldset> (el recuadro) y <legend> (el título del recuadro). Esto ayuda a los usuarios ciegos a saber que esas opciones pertenecen a la misma pregunta.

**Seguridad Básica:**

* Nunca confíes en la validación de HTML (required, min, max). Un usuario con conocimientos técnicos puede borrar esos atributos desde el inspector del navegador y enviar datos corruptos. **La validación real siempre se hace en el Servidor (Backend)**. HTML es solo para mejorar la experiencia del usuario (UX).

**Autocompletado (autocomplete):**

* Es buena práctica usar el atributo autocomplete (ej: autocomplete="email", autocomplete="new-password") para ayudar a los gestores de contraseñas y navegadores a rellenar el formulario por el usuario.

**Sugerencia.**

Investiguen sobre "Estilos por defecto del navegador (User Agent Stylesheet)". Notarán que los botones y inputs se ven diferentes en Chrome, Safari y Firefox. Con CSS pueden normalizarlos.

Logotipo

El contenido generado por IA puede ser incorrecto.