



# **INTRODUCCIÓN A LOS FORMULARIOS EN PÁGINAS WEB**

# CONCEPTO DE FORMULARIOS EN PÁGINAS WEB

- Conjunto de campos que permiten a los usuarios ingresar y enviar información.
- Esenciales para interacciones como registros, búsquedas y encuestas.
- Facilitan el procesamiento y almacenamiento de datos del usuario.

# CARACTERÍSTICAS DE LOS FORMULARIOS WEB

- **Interactividad:** Permite al usuario enviar información de manera activa.
- **Envío de Información:** Uso de métodos GET y POST.
  - **GET:** Envía datos a través de la URL (búsquedas).
  - **POST:** Envía datos de manera segura (formularios de registro).
- **Variabilidad de Datos:** Admite texto, opciones múltiples, archivos, etc.

# ELEMENTOS Y CONTROLES DEL FORMULARIO

- **Campos de Texto:** `<input type="text">`
- **Área de Texto Multilínea:** `<textarea>`
- **Botón de Envío:** `<input type="submit">`
- **Botones de Radio:** `<input type="radio">`
- **Casillas de Verificación:** `<input type="checkbox">`
- **Listas Desplegables:** `<select`
- Ejemplo.

# ATRIBUTOS DE LOS FORMULARIOS

- **action:** URL donde se envía el formulario (<form action="procesar.php">).
- **method:** Define el método de envío (GET o POST).
- **enctype:** Codificación de datos (multipart/form-data para envío de archivos).
- Ejemplo.

# IMPORTANCIA DE USABILIDAD Y ACCESIBILIDAD EN FORMULARIOS

- **Accesibilidad:** Uso de etiquetas <label> y atributos ARIA.
- **Usabilidad:** Claridad en campos obligatorios y retroalimentación de errores.
- Ejemplo.

# VALIDACIÓN DE FORMULARIOS

- **Lado del Cliente:** Validación con HTML5 (required, pattern) y JavaScript.
- **Lado del Servidor:** Validación posterior al envío para mayor seguridad.
- [Ejemplo.](#)

# FORMULARIOS Y EVENTOS

- **onsubmit:** Ejecuta código al enviar el formulario.
- **onchange:** Detecta cambios en campos específicos.
- **onfocus:** Resalta campos cuando el usuario interactúa.
- Ejemplo.





# **CONTROLES DE FORMULARIOS – CAMPOS Y BOTONES**

# TIPOS DE CAMPOS EN FORMULARIOS

- **Texto Simple:** `<input type="text">`
- **Contraseña:** `<input type="password">`
- **Números:** `<input type="number">`
- **Fecha:** `<input type="date">`
- **Correo Electrónico:** `<input type="email">`
- **Área de Texto Multilínea:** `<textarea>`
- Ejemplo.

# CAMPOS DE ENTRADA DE TEXTO

- **Texto:** Entrada simple para nombres y direcciones.
  - Atributos: placeholder, maxlength.
- **Contraseña:** Oculta los caracteres introducidos.
  - Atributos: maxlength, required.
- Ejemplo.

# CAMPOS NUMÉRICOS Y DE FECHA

- **Números:** Control con flechas de incremento/decremento.
  - Atributos: min, max, step.
- **Fecha:** Calendario emergente para selección.
- Atributos: min, max.
- Ejemplo.

# BOTONES DE ENVÍO Y RESTABLECIMIENTO

- **Botón de Envío:** `<input type="submit">`
- **Botón de Restablecimiento:** `<input type="reset">`
- **Botón Genérico:** `<button>` (Para validaciones o acciones personalizadas).
- Ejemplo.

# MEJORES PRÁCTICAS EN FORMULARIOS

- **Placeholder:** Guía visual en campos vacíos.
- **Validación HTML5:** required, pattern, min, max.
- **Accesibilidad:** Etiquetas <label>, atributos ARIA.
- **Agrupación Lógica:** Uso de <fieldset> y <legend>.
- [Ejemplo.](#)

# CAMPOS OBLIGATORIOS Y OPCIONALES

- **Campos Obligatorios:** Indicados con required o \*.
- **Campos Opcionales:** Claramente diferenciados visualmente.
- **Errores y Mensajes:** Retroalimentación en tiempo real.
- Ejemplo.

# EJEMPLOS PRÁCTICOS DE FORMULARIOS

- Creación de un formulario con diferentes tipos de campos y botones.
- Validación en HTML5 y JavaScript.
- Inclusión de etiquetas <fieldset>, <label>, y botones interactivos.
- [Ejemplo.](#)



The background features abstract, flowing waves in shades of red, orange, and yellow, creating a dynamic and modern aesthetic. The waves are layered, with some appearing more prominent than others, giving a sense of depth and movement.

# **FORMULARIOS Y USABILIDAD**

# CONCEPTO DE USABILIDAD APLICADO A FORMULARIOS

- **Contenido:**

- Capacidad del usuario para completar formularios de forma eficiente y satisfactoria.
- Reducción de la carga cognitiva mediante diseño claro y lógico.
- Principios:
  - **Claridad:** Elementos directos y comprensibles.
  - **Consistencia:** Estilo uniforme en etiquetas y botones.
  - **Feedback Visual:** Mensajes de error y éxito inmediatos.
  - **Minimización de Esfuerzo:** Formularios cortos y agrupados.
  - **Validación en Tiempo Real:** Corrección de errores instantánea.

# ELEMENTOS CLAVE PARA FORMULARIOS USABLES

- **Etiquetas Claras:** Uso de <label> para cada campo.
- **Placeholder:** Sugerencias adicionales (no sustituye etiquetas).
- **Mensajes de Error Útiles:** Específicos y orientados a soluciones.
- **Orden Lógico:** Flujo natural en la disposición de campos.
- **Indicadores de Campos Obligatorios:** Asterisco \* o required.

# VALIDACIÓN Y RETROALIMENTACIÓN DEL USUARIO

- **Validación Cliente:** HTML5 y JavaScript.
- **Retroalimentación Positiva:** Iconos o bordes verdes.
- **Mensajes de Ayuda:** Atributos title o mensajes emergentes.

# AGRUPACIÓN Y ORDENACIÓN DE CAMPOS

- **Agrupación con <fieldset> y <legend>.**
- **Orden Natural:** De arriba a abajo, de izquierda a derecha.
- **Secciones Claras:** Información personal, contacto, etc.



# MINIMIZACIÓN DE CARGA COGNITIVA

- **Listas Desplegables y Botones:** Minimizar texto libre.
- **Autocompletado:** Uso de autocomplete para sugerencias.
- **Formularios Cortos:** Dividir formularios largos en pasos.



# MEJORAS DE ACCESIBILIDAD

- **Navegación por Teclado:** Campos accesibles sin ratón.
- **Etiquetas ARIA:** Uso de aria-label y aria-describedby.
- **Foco de Entrada:** Resaltar campo activo con CSS.



# EJEMPLO DE FORMULARIO USABLE Y ACCESIBLE

- Formulario con validación, agrupación y accesibilidad.
- Inclusión de mensajes de error, etiquetas ARIA y autocompletado.
- Ejemplo práctico paso a paso.
- [Ejemplo.](#)





# **EVENTOS Y VALIDACIÓN DE FORMULARIOS**

# CONCEPTO DE EVENTOS EN FORMULARIOS

- Los eventos permiten interactuar con los elementos del formulario.
- Se activan por acciones del usuario (clic, entrada de datos, envío).
- Tipos comunes:
  - **onfocus:** Al obtener foco.
  - **onblur:** Al perder foco.
  - **oninput:** Durante la entrada de datos.
  - **onsubmit:** Al enviar formulario.
  - **onclick:** Al hacer clic.

# MANEJO DE EVENTOS CON JAVASCRIPT

- **addEventListener():** Añade eventos a campos y botones.

Ejemplo:

```
const emailField = document.getElementById('email');
emailField.addEventListener('blur', function() {
    if (!validateEmail(emailField.value)) {
        alert('Por favor ingrese un correo electrónico válido');
    }
});
```

**event.preventDefault():** Evita el envío de formularios no válidos.

# VALIDACIÓN DE FORMULARIOS

- Asegura que los datos ingresados sean correctos y completos.
- Se realiza en dos niveles:
  - **Cliente (Navegador):** Inmediata, reduce errores y tráfico.
  - **Servidor:** Seguridad adicional, evita manipulación de datos.

# VALIDACIÓN CON HTML5

- **required:** Campo obligatorio.
- **pattern:** Expresión regular para formatos específicos.
- **min, max, maxlength:** Rango de valores o caracteres.

Ejemplo:

```
<input type="email" required pattern=".+@.+\.com">
```

# VALIDACIÓN CON JAVASCRIPT

- Validación en tiempo real durante la entrada de datos.
- Función de validación personalizada:

```
function validateEmail(email) {  
    const regex = /^[^\\s@]+@[^\\s@]+\\.\\.[^\\s@]+$/;  
    return regex.test(email);  
}
```

Aplicación:

```
document.querySelector('form').onsubmit = function(event) {  
    if (!validateEmail(document.getElementById('email').value)) {  
        event.preventDefault();  
        alert('Correo electrónico no válido.');    }  
}
```

# MEJORES PRÁCTICAS EN VALIDACIÓN

- Validar durante la escritura (oninput) y al perder foco (onblur).
- Mostrar errores progresivos, uno a la vez.
- Retroalimentación visual con colores (rojo: error, verde: éxito).
- Mensajes claros y orientados a soluciones.



# FORMULARIO COMPLETO CON VALIDACIÓN

- Formulario con validación HTML5 y JavaScript.
- Validación de campos de correo, contraseña y número.
- Inclusión de mensajes personalizados y prevención de envío.
- [Ejemplo.](#)





# **ADAPTACIÓN DE PLANTILLAS Y FORMULARIOS**

# CONCEPTO DE ADAPTACIÓN DE PLANTILLAS

- Estructura estandarizada y reutilizable para páginas web.
- Mantienen coherencia visual y optimizan el desarrollo.
- Beneficios:
  - **Consistencia:** Apariencia uniforme en todo el sitio.
  - **Reutilización de Código:** Evita duplicidad y facilita el mantenimiento.
  - **Facilidad de Actualización:** Modificaciones globales desde una plantilla.

# ELEMENTOS ADAPTABLES EN PLANTILLAS

- **Regiones Editables:** Contenido que varía (texto, imágenes).
- **Regiones No Editables:** Encabezados, pies de página.
- **Campos Dinámicos:** Variables para personalizar títulos y descripciones.

**Ejemplo:**

```
<h1>{{titulo}}</h1>
```

```
<p>{{descripcion}}</p>
```

- [Ejemplo.](#)

# ADAPTACIÓN DE FORMULARIOS

- Formularios adaptados para diferentes contextos.
- **Formularios Dinámicos:**
  - **Campos Condicionales:** Mostrar campos según respuestas previas.
  - **Multi-Pasos:** Dividir formularios largos en secciones.
- Ejemplo.



# HERRAMIENTAS PARA LA ADAPTACIÓN

- **HTML y CSS Dinámicos:** Modificación visual y estructural.
- **JavaScript:** Mostrar/ocultar campos dinámicamente.
- **CMS (WordPress, Joomla):** Uso de plantillas preexistentes.



# MEJORES PRÁCTICAS DE ADAPTACIÓN

- **Simplificación:** Mantener plantillas sencillas y funcionales.
- **Consistencia Estética:** Alineación con el diseño general del sitio.
- **Validación Personalizada:** Validar datos según el propósito del formulario.



# EJEMPLO DE ADAPTACIÓN DE PLANTILLA

- Demostración de plantilla con regiones editables y dinámicas.
- Ejemplo práctico con cambios en títulos, imágenes y contenido.