

4. Formularios y eventos. Criterios de accesibilidad y usabilidad en el diseño de formularios.

El diseño de formularios debe considerar la accesibilidad y usabilidad para todos los usuarios. En esta sección, se abordan los eventos y criterios para lograr formularios accesibles y fáciles de usar.

4.1. Agrupación de datos.

Organizar los datos de manera lógica en un formulario mejora su usabilidad. Esta práctica facilita la entrada de información, haciendo que el formulario sea más intuitivo y fácil de completar. Para lograrlo, es recomendable agrupar campos relacionados dentro de secciones claramente definidas. Por ejemplo, se pueden utilizar elementos como `<fieldset>` y `<legend>` para crear grupos de campos que pertenezcan a la misma categoría, como información personal, detalles de contacto o preferencias. Además, los encabezados y las etiquetas descriptivas ayudan a guiar al usuario a través del formulario, reduciendo la posibilidad de errores y aumentando la eficiencia.

Recuerda

El elemento `<fieldset>` se utiliza para agrupar de manera lógica varios controles y etiquetas dentro de un formulario. Este agrupamiento visual y semántico ayuda a los usuarios a entender que los campos dentro de un `<fieldset>` están relacionados entre sí. El elemento `<legend>` proporciona una descripción para el contenido del `<fieldset>`, funcionando como un título para el grupo de campos.

Un ejemplo básico podría ser el siguiente:

```

<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Campos de texto</title>
</head>
<body>
  <form>
    <fieldset>
      <legend>Información Personal</legend>
      <label for="nombre">Nombre:</label>
      <input type="text" id="nombre" name="nombre"><br><br>
      <label for="apellido">Apellido:</label>
      <input type="text" id="apellido" name="apellido"><br><br>
    </fieldset>
    <fieldset>
      <legend>Detalles de Contacto</legend>
      <label for="email">Correo Electrónico:</label>
      <input type="email" id="email" name="email"><br><br>
      <label for="telefono">Teléfono:</label>
      <input type="tel" id="telefono" name="telefono"><br><br>
    </fieldset>
    <button type="submit">Enviar</button>
  </form>
</body>
</html>

```

Pie de página: Resultado del código anterior renderizado en un navegador web.

4.2. Adecuación del tamaño del formulario (división en distintas páginas).

Para formularios extensos, dividirlos en varias páginas puede mejorar significativamente la experiencia del usuario. Esta estrategia evita que los usuarios se sientan abrumados por la cantidad

de información que deben proporcionar de una vez. Al dividir un formulario en secciones más pequeñas y manejables, se facilita la navegación y se permite a los usuarios concentrarse en un conjunto de datos a la vez. Esta técnica también puede incluir indicadores de progreso para mostrar a los usuarios cuánto les falta por completar. Implementar esta división de manera efectiva implica asegurarse de que cada página del formulario tenga un propósito claro y esté enfocada en un tema específico.

Un ejemplo de división de formulario en varias páginas podría ser el siguiente:

```

<!-- Página 1 -->
<form id="form1">
  <fieldset>
    <legend>Información Personal</legend>
    <label for="nombre">Nombre:</label>
    <input type="text" id="nombre" name="nombre"><br><br>

    <label for="apellido">Apellido:</label>
    <input type="text" id="apellido" name="apellido"><br><br>
  </fieldset>
  <button type="button" onclick="mostrarPagina2()">Siguiete</button>
</form>
<!-- Página 2 -->
<form id="form2" style="display:none;">
  <fieldset>
    <legend>Detalles de Contacto</legend>
    <label for="email">Correo Electrónico:</label>
    <input type="email" id="email" name="email"><br><br>
    <label for="telefono">Teléfono:</label>
    <input type="tel" id="telefono" name="telefono"><br><br>
  </fieldset>
  <button type="button" onclick="mostrarPagina1()">Anterior</button>
  <button type="submit">Enviar</button>
</form>
<script>
function mostrarPagina1() {
  document.getElementById('form1').style.display = 'block';
  document.getElementById('form2').style.display = 'none';
}
function mostrarPagina2() {
  document.getElementById('form1').style.display = 'none';
  document.getElementById('form2').style.display = 'block';
}
</script>

```

The image displays two sequential screenshots of a web browser interface. The top screenshot shows a form titled "Información Personal" with two input fields: "Nombre:" and "Apellido:". Below these fields is a button labeled "Siguiete". The bottom screenshot shows a form titled "Detalles de Contacto" with two input fields: "Correo Electrónico:" and "Teléfono:". Below these fields are two buttons: "Anterior" and "Enviar". Both screenshots have a dark header bar with navigation icons (back, refresh, search) and the text "Buscar o escribir dirección web".

Pie de página: Resultado del código anterior renderizado en un navegador web.

Actividad 8

Imagina que trabajas como diseñador/a de interfaces de usuario en una empresa que desarrolla aplicaciones web. Te han asignado la tarea de diseñar un formulario de inscripción en línea para un evento corporativo importante. Debes asegurarte de que el formulario sea accesible y fácil de usar para todos los participantes, siguiendo los criterios de accesibilidad y usabilidad.

El objetivo es diseñar un formulario de inscripción en línea que siga los principios de agrupación de datos y adecuación del tamaño del formulario, asegurando la accesibilidad y usabilidad para todos los usuarios.

Instrucciones:

Agrupar los campos de datos personales, detalles de contacto y preferencias en secciones claramente definidas usando `<fieldset>` y `<legend>`.

Asegúrate de que cada grupo tenga encabezados y etiquetas descriptivas.

Divide el formulario en varias páginas si es extenso.

Implementa indicadores de progreso para mostrar cuánto falta para completar el formulario.

Diseña un formulario de inscripción para el evento "Conferencia Anual de Innovación Tecnológica". Los campos requeridos son: nombre, apellido, correo electrónico, número de teléfono, empresa, cargo, preferencias de alimentación (vegano, vegetariano, omnívoro) y talleres a los que desean asistir.

4.3. Identificación de los campos obligatorios.

Es fundamental que los campos obligatorios de un formulario estén claramente identificados para evitar errores y asegurar que los usuarios proporcionen toda la información necesaria. La falta de claridad en este aspecto puede llevar a que los formularios se envíen incompletos, lo que resulta en frustración tanto para los usuarios como para los administradores de los datos.

Algunos de los métodos para identificar campos obligatorios son los siguientes:

Colocar un asterisco de algún color (habitualmente rojo) junto a la etiqueta del campo obligatorio. Este método es ampliamente reconocido y entendido por los usuarios:

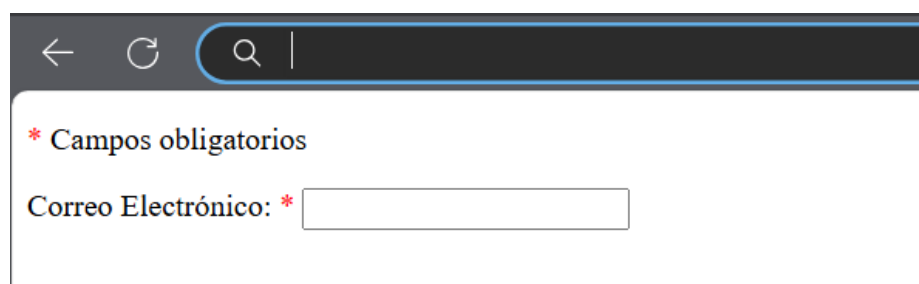
```
<label for="nombre">Nombre: <span style="color: red;">*</span></label>
<input type="text" id="nombre" name="nombre" required>
```



Pie de página: Resultado del código anterior renderizado en un navegador web.

Añadir una nota al inicio del formulario indicando que los campos marcados con un asterisco son obligatorios. Esto asegura que los usuarios comprendan el significado del asterisco:

```
<p><span style="color: red;">*</span> Campos obligatorios</p>
<label for="email">Correo Electrónico: <span style="color: red;">*</span></label>
<input type="email" id="email" name="email" required>
```



Pie de página: Resultado del código anterior renderizado en un navegador web.

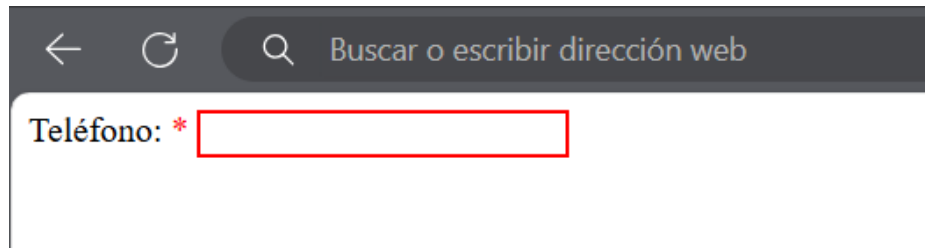
Utilizar estilos CSS para resaltar los campos obligatorios, por ejemplo, cambiando el color de fondo o el borde del campo:

```

<style>
  .obligatorio {
    border: 2px solid red;
  }
</style>

<label for="telefono">Teléfono: <span style="color:red;">*</span></label>
<input type="tel" id="telefono" name="telefono" class="obligatorio" required>

```



Pie de página: Resultado del código anterior renderizado en un navegador web.

Implementar mensajes de validación que se activen si el usuario intenta enviar el formulario sin completar los campos obligatorios. Estos mensajes deben ser claros y específicos:

```

<script>
  function validarFormulario() {
    var nombre = document.getElementById("nombre").value;
    var email = document.getElementById("email").value;
    if (nombre === "" || email === "") {
      alert("Por favor, complete todos los campos obligatorios.");
      return false;
    }
    return true;
  }
</script>

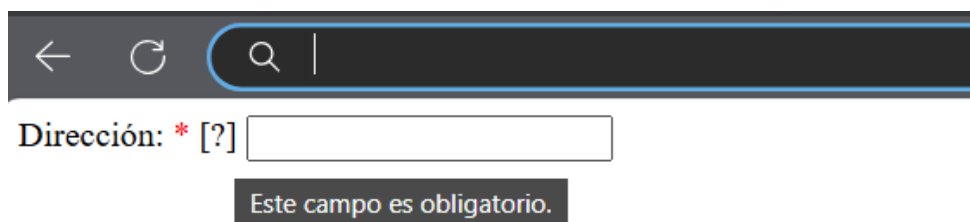
```

Añadir iconos informativos al lado de los campos obligatorios que, al pasar el cursor sobre ellos, muestren un mensaje explicativo:

```

<label for="direccion">Dirección: <span style="color:red;">*</span> <span title="Este campo es obligatorio.">[?]</span></label>
<input type="text" id="direccion" name="direccion" required>

```



Pie de página: Resultado del código anterior renderizado en un navegador web.

4.4. Orden lógico de la petición de datos.

Estructurar un formulario siguiendo un orden lógico mejora la experiencia del usuario. Un formulario bien organizado guía al usuario a través de un flujo natural de datos, lo que reduce la confusión y el tiempo necesario para completarlo.

Algunas estrategias populares para implementar un orden lógico en los formularios son las siguientes:

Ordenar los campos de acuerdo con la secuencia en que se espera que el usuario piense sobre la información. Por ejemplo, comenzar con información personal básica antes de pasar a detalles de contacto o preferencias:

```
<form>
  <!-- Información Personal -->
  <fieldset>
    <legend>Información Personal</legend>
    <label for="nombre">Nombre:</label>
    <input type="text" id="nombre" name="nombre"><br><br>

    <label for="apellido">Apellido:</label>
    <input type="text" id="apellido" name="apellido"><br><br>
  </fieldset>

  <!-- Detalles de Contacto -->
  <fieldset>
    <legend>Detalles de Contacto</legend>
    <label for="email">Correo Electrónico:</label>
    <input type="email" id="email" name="email"><br><br>

    <label for="telefono">Teléfono:</label>
    <input type="tel" id="telefono" name="telefono"><br><br>
  </fieldset>

  <button type="submit">Enviar</button>
</form>
```

Pie de página: Resultado del código anterior renderizado en un navegador web.

Agrupar los campos por tema o categoría, como información personal, detalles de contacto, preferencias, etc. Esto ayuda a los usuarios a concentrarse en un tipo de información a la vez.

EDITORIAL TUTOR FORMACIÓN

Colocar los campos más importantes o más frecuentemente utilizados al principio del formulario. Esto asegura que la información clave se recoja incluso si el usuario no completa el formulario en su totalidad.

Diseñar el formulario para que siga el flujo de trabajo del usuario. Por ejemplo, en un formulario de compra, comenzar con los datos de facturación, seguir con la información de envío y finalizar con el método de pago.

Para formularios largos, dividirlos en secciones claramente diferenciadas con encabezados que indiquen el contenido de cada sección. Esto mejora la legibilidad y hace que el formulario sea menos intimidante.

En formularios muy largos, utilizar indicadores de progreso para informar a los usuarios sobre su avance. Esto proporciona una sensación de logro y motiva a continuar hasta completar el formulario.

A continuación, se presenta un código HTML que incorpora todas las mejores prácticas mencionadas. Este formulario está diseñado para un proceso de registro que incluye información personal, detalles de contacto y preferencias. Además, se ha dividido en secciones con encabezados claros y utiliza un indicador de progreso para guiar al usuario a través del formulario.

```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Formulario de Registro</title>
7   <style>
8     body {
9       font-family: Arial, sans-serif;
10      margin: 20px;
11    }
12    form {
13      max-width: 600px;
14      margin: auto;
15    }
16    fieldset {
17      margin-bottom: 20px;
18      padding: 20px;
19      border: 1px solid #ccc;
20      border-radius: 5px;
21    }
22    legend {
23      font-weight: bold;
24    }
25    label {
26      display: block;
27      margin: 10px 0 5px;
28    }
29    input, select {
30      width: 100%;
31      padding: 8px;
32      margin-bottom: 10px;
33      border: 1px solid #ccc;
34      border-radius: 4px;
35    }
```


EDITORIAL TUTOR FORMACIÓN

```
36     .progress {
37         display: flex;
38         justify-content: space-between;
39         margin-bottom: 20px;
40     }
41     .progress div {
42         flex-grow: 1;
43         height: 10px;
44         background-color: #ccc;
45         margin-right: 5px;
46         border-radius: 5px;
47     }
48     .progress div.active {
49         background-color: #4caf50;
50     }
51     .actions {
52         display: flex;
53         justify-content: space-between;
54     }
55 </style>
56 </head>
```

```
57 <body>
58     <h1>Formulario de Registro</h1>
59     <div class="progress">
60         <div id="progress1" class="active"></div>
61         <div id="progress2"></div>
62         <div id="progress3"></div>
63     </div>
64     <form id="registrationForm" onsubmit="return validarFormulario()">
65         <!-- Sección 1: Información Personal -->
66         <fieldset id="section1">
67             <legend>Información Personal</legend>
68             <label for="nombre">Nombre: <span style="color: red;">*</span></label>
69             <input type="text" id="nombre" name="nombre" required>
70
71             <label for="apellido">Apellido: <span style="color: red;">*</span></label>
72             <input type="text" id="apellido" name="apellido" required>
73         </fieldset>
74
75         <!-- Sección 2: Detalles de Contacto -->
76         <fieldset id="section2" style="display:none;">
77             <legend>Detalles de Contacto</legend>
78             <label for="email">Correo Electrónico: <span style="color: red;">*</span></label>
79             <input type="email" id="email" name="email" required>
80
81             <label for="telefono">Teléfono:</label>
82             <input type="tel" id="telefono" name="telefono">
83         </fieldset>
84     </form>
```

EDITORIAL TUTOR FORMACIÓN

```
84
85 <!-- Sección 3: Preferencias -->
86 <fieldset id="section3" style="display:none;">
87   <legend>Preferencias</legend>
88   <label for="boletin">Suscribirse al boletín:</label>
89   <input type="checkbox" id="boletin" name="boletin">
90
91   <label for="actualizaciones">Recibir actualizaciones:</label>
92   <input type="checkbox" id="actualizaciones" name="actualizaciones">
93 </fieldset>
94
95 <!-- Botones de Navegación -->
96 <div class="actions">
97   <button type="button" onclick="mostrarAnterior()" id="prevButton" style="display:none;">Anterior</button>
98   <button type="button" onclick="mostrarSiguiente()" id="nextButton">Siguiente</button>
99   <button type="submit" id="submitButton" style="display:none;">Enviar</button>
100 </div>
101 </form>
102
```

```
102
103 <script>
104   let currentSection = 1;
105
106   function mostrarSiguiente() {
107     document.getElementById('section' + currentSection).style.display = 'none';
108     document.getElementById('progress' + currentSection).classList.remove('active');
109     currentSection++;
110     document.getElementById('section' + currentSection).style.display = 'block';
111     document.getElementById('progress' + currentSection).classList.add('active');
112     actualizarBotones();
113   }
114
115   function mostrarAnterior() {
116     document.getElementById('section' + currentSection).style.display = 'none';
117     document.getElementById('progress' + currentSection).classList.remove('active');
118     currentSection--;
119     document.getElementById('section' + currentSection).style.display = 'block';
120     document.getElementById('progress' + currentSection).classList.add('active');
121     actualizarBotones();
122   }
123
124   function actualizarBotones() {
125     document.getElementById('prevButton').style.display = currentSection === 1 ? 'none' : 'inline';
126     document.getElementById('nextButton').style.display = currentSection === 3 ? 'none' : 'inline';
127     document.getElementById('submitButton').style.display = currentSection === 3 ? 'inline' : 'none';
128   }
129
```

```
129
130
131   function validarFormulario() {
132     var nombre = document.getElementById("nombre").value;
133     var email = document.getElementById("email").value;
134     if (nombre === "" || email === "") {
135       alert("Por favor, complete todos los campos obligatorios.");
136       return false;
137     }
138     return true;
139   }
140 </script>
141 </body>
</html>
```

EDITORIAL TUTOR FORMACIÓN

- Descripción del código
 - Agrupación por categoría: Los campos están agrupados en secciones lógicas utilizando <fieldset> y <legend> para Información Personal, Detalles de Contacto y Preferencias.
 - Campos importantes al principio: Los campos de nombre y apellido, que son críticos, están ubicados en la primera sección.
 - Flujo de Trabajo natural: El formulario está diseñado para seguir un flujo natural: primero la información personal, luego los detalles de contacto y finalmente las preferencias.
 - División en secciones: El formulario está dividido en secciones claramente diferenciadas con encabezados (<legend>) que indican el contenido de cada sección.
 - Indicadores de progreso: Se utiliza un indicador de progreso visual para mostrar en qué parte del formulario se encuentra el usuario.
- Funcionalidad del código
 - Navegación entre secciones: Botones "Siguiente" y "Anterior" permiten moverse entre las secciones del formulario.
 - Validación: La función validarFormulario() se asegura de que los campos obligatorios estén completos antes de permitir el envío del formulario.
 - Actualización de botones: La función actualizarBotones() controla la visibilidad de los botones de navegación y envío según la sección actual.

The image displays three sequential screenshots of a web browser showing a registration form titled "Formulario de Registro". Each screenshot shows a different section of the form, indicated by a green progress bar at the top of the form area.

- First Screenshot:** Shows the "Información Personal" section. It contains two text input fields labeled "Nombre:" and "Apellido:", both with red asterisks indicating they are required. A "Siguiente" button is located below the fields.
- Second Screenshot:** Shows the "Detalles de Contacto" section. It contains two text input fields labeled "Correo Electrónico:" and "Teléfono:", both with red asterisks. "Anterior" and "Siguiente" buttons are located below the fields.
- Third Screenshot:** Shows the "Preferencias" section. It contains two checkboxes: "Suscribirse al boletín:" (checked) and "Recibir actualizaciones:" (unchecked). "Anterior" and "Enviar" buttons are located below the checkboxes.

Pie de página: Resultado del código anterior renderizado en un navegador web.

Nota

- Las tres ventajas principales de mantener un orden lógico en los formularios son las siguientes:
- Reducción de la confusión: Un flujo de datos lógico y natural ayuda a los usuarios a entender qué información se solicita y en qué orden, reduciendo la posibilidad de errores y omisiones.
- Mejora de la experiencia del Usuario: Los usuarios pueden completar el formulario de manera más rápida y eficiente cuando los campos están organizados de forma coherente.
- Mayor tasa de finalización: Un formulario que es fácil de seguir y no abruma a los usuarios tiene más probabilidades de ser completado en su totalidad.

Actividad 9

- ¿Por qué es importante estructurar un formulario siguiendo un orden lógico?
- Menciona dos estrategias populares para implementar un orden lógico en los formularios.
- ¿Cuál es la ventaja de colocar los campos más importantes o más frecuentemente utilizados al principio del formulario?
- En un formulario de compra, ¿cuál sería un ejemplo de diseño que sigue el flujo de trabajo del usuario?
- ¿Cómo se puede mejorar la legibilidad y reducir la intimidación en formularios largos?
- ¿Qué función tienen los indicadores de progreso en formularios muy largos?

4.5. Información correcta al usuario.

Proporcionar información clara y precisa es esencial para garantizar que los usuarios comprendan los procesos y requisitos del formulario. Cuando un usuario comete un error al completar un formulario, es fundamental informarle de manera efectiva para que pueda corregirlo. Del mismo modo, es importante confirmar que las acciones han sido completadas correctamente. Esta subsección aborda las mejores prácticas para comunicar errores y confirmaciones a los usuarios.

En este contexto, existen varios métodos para informar al usuario sobre errores:

- **Mensajes de error:** Los mensajes de error deben aparecer cerca del campo donde se produjo el error. Esto facilita al usuario identificar y corregir el problema rápidamente. Por ejemplo, si el campo de correo electrónico no se ha completado correctamente, el mensaje de error debería aparecer junto a ese campo:

```
<form onsubmit="return validarFormulario()">
  <label for="email">Correo Electrónico:</label>
  <input type="email" id="email" name="email">
  <span id="emailError" style="color:■red; display:none;">Por favor, ingrese un correo electrónico válido.</span><br><br>

  <button type="submit">Enviar</button>
</form>

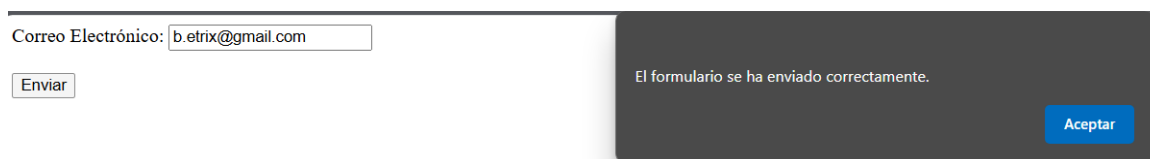
<script>
function validarFormulario() {
  var email = document.getElementById("email").value;
  var emailError = document.getElementById("emailError");
  if (email === "") {
    | emailError.style.display = 'inline';
    | return false;
  } else {
    | emailError.style.display = 'none';
  }
  return true;
}
</script>
```

- **Indicaciones claras:** Los mensajes de error deben ser claros y específicos, indicando exactamente qué es lo que está mal y, si es posible, cómo corregirlo. Evitar mensajes ambiguos que puedan confundir al usuario.
- **Uso de estilos:** Utilizar colores y estilos que resalten los mensajes de error. El rojo es un color comúnmente usado para indicar errores. Además, se pueden utilizar iconos visuales (como un signo de exclamación) para hacer que el mensaje sea más notorio.

Por otra parte, después de que un usuario completa una acción con éxito, es importante que reciba un mensaje de confirmación inmediato, el cual puede ser una ventana emergente, un mensaje de texto en la misma página o una redirección a una página de confirmación:

```
<script>
  function mostrarConfirmacion() {
    | alert("El formulario se ha enviado correctamente.");
  }
</script>

<button type="button" onclick="mostrarConfirmacion()">Enviar</button>
```



Correo Electrónico:

El formulario se ha enviado correctamente.

Pie de página: Resultado del código anterior renderizado en un navegador web.

Los mensajes deben incluir detalles sobre la acción realizada; por ejemplo, si el formulario se ha enviado correctamente, el mensaje de confirmación puede contener un resumen de los datos enviados o el siguiente paso que debe seguir el usuario.

Además de los mensajes de texto, se pueden utilizar animaciones o cambios de estilo, como cambiar el color del botón de envío, para proporcionar retroalimentación visual y señalar que la acción se ha completado con éxito.

4.6. Utilización de páginas de error y de confirmación.

Las páginas de error y confirmación juegan un papel fundamental en la comunicación con el usuario durante la interacción con formularios y otros procesos. Implementarlas correctamente mejora significativamente la experiencia del usuario al proporcionar información clara y directa sobre el estado de sus acciones.

Las páginas de error deben proporcionar un mensaje claro y útil que explique lo que ha salido mal. Deben evitarse los mensajes técnicos o confusos que no aportan ninguna solución al usuario. Por ejemplo:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Error</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      text-align: center;
      padding: 50px;
    }
    .error-message {
      color: red;
      font-size: 24px;
    }
  </style>
</head>
<body>
  <h1 class="error-message">Lo sentimos, ha ocurrido un error.</h1>
  <p>Por favor, intente nuevamente o contacte al soporte técnico si el problema persiste.</p>
  <button onclick="window.history.back()">Volver</button>
</body>
</html>
```



Pie de página: Resultado del código anterior renderizado en un navegador web.

Además de explicar el error, la página debe ofrecer pasos claros que el usuario puede seguir para intentar solucionar el problema. Esto puede incluir instrucciones para revisar ciertos campos, volver a intentarlo, o contactar con soporte técnico.

Además, incluir enlaces o botones que permitan al usuario regresar a la página anterior, intentar nuevamente, o ir a la página principal del sitio evita que el usuario se quede atrapado en una página de error sin saber cómo proceder.

Por otro lado, una página de confirmación debe indicar claramente que la acción del usuario se ha completado con éxito. El mensaje debe ser positivo y agradecer al usuario por su interacción. Por ejemplo:

```

<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Confirmación</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      text-align: center;
      padding: 50px;
    }
    .success-message {
      color: green;
      font-size: 24px;
    }
  </style>
</head>
<body>
  <h1 class="success-message">¡Gracias! Su formulario ha sido enviado con éxito.</h1>
  <p>Hemos recibido su información y nos pondremos en contacto con usted pronto.</p>
  <a href="index.html">Volver a la página principal</a>
</body>
</html>

```



Pie de página: Resultado del código anterior renderizado en un navegador web.

La página de confirmación puede incluir un resumen de la información enviada o la acción realizada. Esto proporciona al usuario una revisión de lo que ha hecho y confirma que todo se ha procesado correctamente.

Además, es importante incluir información sobre los próximos pasos que el usuario debe seguir. Por ejemplo, si el formulario era para una inscripción, la página de confirmación puede indicar cuándo y cómo el usuario recibirá más información.

Actividad 10

A continuación, se presentan varias descripciones y términos relacionados con la comunicación de errores y confirmaciones en formularios. Relaciona cada descripción con el término correcto.

Descripciones:

Mensajes de error deben aparecer cerca del campo donde se produjo el error para facilitar su identificación y corrección.

Deben utilizarse colores y estilos que resalten los mensajes de error, como el uso del color rojo y iconos visuales.

Mensajes de error claros y específicos que indiquen exactamente qué está mal y cómo corregirlo.

EDITORIAL TUTOR FORMACIÓN

Proporcionar retroalimentación visual y señalización de que la acción se ha completado con éxito, como cambiar el color del botón de envío.

Un mensaje que aparece después de que una acción se ha completado correctamente, puede ser una ventana emergente o un mensaje de texto.

Una página que explica lo que ha salido mal con pasos claros para solucionar el problema y enlaces para volver a intentar o regresar a la página principal.

Una página que indica claramente que la acción del usuario se ha completado con éxito, incluyendo un resumen de la información enviada o la acción realizada.

Términos:

A. Indicaciones claras

B. Mensajes de error

C. Uso de estilos

D. Página de confirmación

E. Mensaje de confirmación

F. Retroalimentación visual

G. Página de error

Para cerrar la primera unidad se expone un ejemplo de cómo estructurar un cuestionario en HTML. En este caso, se trata de un cuestionario relacionado con Chiara Oliver, que invita a los participantes a compartir sus opiniones y preferencias sobre su música y próximos conciertos. Este ejemplo abarca elementos clave como campos de texto, correos electrónicos, menús desplegables y botones, además de mostrar cómo aplicar estilos CSS para mejorar la presentación visual del formulario:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Cuestionario sobre Chiara Oliver</title>
5   <style>
6     body {
7       background-color: #FFB6C1;
8       font-family: 'Courier New', Courier, monospace;
9     }
10    .container {
11      width: 80%;
12      margin: auto;
13      background-color: white;
14      padding: 20px;
15      border-radius: 15px;
16      box-shadow: 0px 0px 10px 2px #000000;
17    }
18    h1, h2 {
19      text-align: center;
20    }
21    label {
22      display: block;
23      margin: 10px 0px;
24    }
25    input, textarea, select {
26      width: 100%;
27      padding: 10px;
28      margin-bottom: 10px;
29      border-radius: 5px;
30      border: 1px solid #000000;
31    }
```

```

32         button {
33             display: block;
34             width: 100px;
35             margin: auto;
36             padding: 10px;
37             border: none;
38             color: □white;
39             background-color: ■#000000;
40             border-radius: 5px;
41             cursor: pointer;
42         }
43         button:hover {
44             background-color: ■#FF69B4;
45         }
46     </style>
47 </head>

```

```

</body>
<div class="container">
  <h1>Cuestionario sobre Chiara Oliver</h1>
  <h2>¡Nos encantaría saber lo que piensas!</h2>
  <form>
    <label for="name">Nombre:</label>
    <input type="text" id="name" name="name" required>
    <label for="email">Correo electrónico:</label>
    <input type="email" id="email" name="email" required>
    <label for="concert">¿A qué concierto planeas asistir?</label>
    <select id="concert" name="concert" required>
      <option value="">Selecciona un concierto</option>
      <option value="menorca">Menorca - 21 y 22 SEP</option>
      <option value="murcia">Murcia - 4 OCT</option>
      <option value="valencia">Valencia - 5 OCT</option>
      <!-- Añade aquí las demás opciones -->
    </select>
    <label for="favorite-song">¿Cuál es tu canción favorita de 'La Libreta Rosa'?</label>
    <select id="favorite-song" name="favorite-song" required>
      <option value="">Selecciona una canción</option>
      <option value="3 de febrero">3 de febrero</option>
      <option value="Ronda de Más">Ronda de Más</option>
      <option value="Cada Vez">Cada Vez</option>
      <option value="La Invitada">La Invitada</option>
      <option value="El parque (Ft. Violeta Hódar)">El parque (Ft. Violeta Hódar)</option>
      <option value="Todas Las Versiones De Mí">Todas Las Versiones De Mí</option>
    </select>
    <button type="submit">Enviar</button>
  </form>
</div>
</body>
</html>

```

EDITORIAL TUTOR FORMACIÓN

Buscar o escribir dirección web

Cuestionario sobre Chiara Oliver

¡Nos encantaría saber lo que piensas!

Nombre:

Correo electrónico:

¿A qué concierto planeas asistir?

Selecciona un concierto

¿Cuál es tu canción favorita de 'La Libreta Rosa'?

Selecciona una canción

Enviar

¿A qué concierto planeas asistir?

Selecciona un concierto

Selecciona un concierto

Menorca - 21 y 22 SEP

Murcia - 4 OCT

Valencia - 5 OCT

Enviar

¿Cuál es tu canción favorita de 'La Libreta Rosa'?

Selecciona una canción

Selecciona una canción

3 de febrero

Ronda de Más

Cada Vez

La Invitada

El parque (Fl. Violeta Hódar)

Todas Las Versiones De Mí

Pie de página: Resultado del código anterior renderizado en un navegador web.

5. Prueba de autoevaluación.

¿Cuál es una función principal de los formularios en una página web?

- A) Permitir que los usuarios descarguen archivos
- B) Facilitar la interacción y recopilación de datos
- C) Proporcionar enlaces a otras páginas

¿Qué método se puede utilizar para asociar una etiqueta (<label>) a un campo de entrada en un formulario?

- A) Utilizando el atributo name
- B) Utilizando el atributo value
- C) Utilizando el atributo for

¿Cuál de los siguientes no es un tipo de campo de entrada en un formulario web?

- A) <input type="email">
- B) <input type="image">
- C) <input type="print">

¿Qué tecnología se recomienda para asegurar la transmisión de datos en formularios web?

- A) HTTP
- B) HTTPS
- C) FTP

¿Cuál es el propósito de usar tokens CSRF en los formularios?

- A) Permitir la carga de archivos grandes
- B) Proteger contra ataques de falsificación de solicitudes entre sitios
- C) Mejorar la estética del formulario

Los formularios son herramientas fundamentales para la ___ entre los usuarios y las páginas web.

Un formulario de contacto permite a los visitantes de un sitio web enviar un mensaje directamente al ___ del sitio.

Los campos de texto permiten al usuario introducir texto libre, y se utilizan para recoger información como nombres y ___ electrónicos.

Al enviar el formulario, se desencadena una solicitud HTTP al ___.

Las etiquetas deben colocarse en una posición consistente en relación con los campos de ___.

Plantillas en la construcción de páginas web

The image displays a screenshot of the 'Electro' website template, which is a responsive e-commerce site for electronics. The main banner features a woman wearing headphones and holding a smartphone, with a 'GET 15% SALE ON UNDER FAVORABLE SMART GADGETS' promotion. Below the banner, there are sections for 'Smart Mobiles', 'Smart Headset', and 'Portable Bluetooth Speaker'. The 'Trending Products' section includes various electronic items like smart speakers, headphones, and smartwatches. A 'Special Products' section highlights a 'Big Billion Day 2021' sale with up to 85% off on smartphones. The 'Our Brands' section features logos for D&G, Sony, Rolex, Microsoft, and Levi's.

100% Responsive

Best Electronic Store
Electro.

- Elementor
- Revolution Slider
- Mailchimp
- WPML Support
- 1-Click Installation

WPML Support Documentation

Wordpress WooCommerce

PSD Files Revolution Slider

Elementor Mailchimp