

EJEMPLO DE UN FORMULARIO HTML QUE INCLUYE MEJORAS EN ACCESIBILIDAD Y USABILIDAD:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Formulario de Ejemplo</title>
  <style>
    .error {
      color: red;
    }
  </style>
</head>
<body>
  <h1>Formulario de Ejemplo</h1>
  <form action="procesar.php" method="post" enctype="multipart/form-data"
aria-labelledby="formTitle">
    <h2 id="formTitle">Por favor, complete el siguiente formulario</h2>

    <!-- Campo de Texto -->
    <div>
      <label for="nombre">Nombre:</label>
      <input type="text" id="nombre" name="nombre" aria-
required="true">
      <span class="error" id="nombreError" aria-live="polite"></span>
    </div>
    <br>

    <!-- Área de Texto Multilínea -->
    <div>
      <label for="comentarios">Comentarios:</label><br>
      <textarea id="comentarios" name="comentarios" rows="4" cols="50"
aria-required="true"></textarea>
```

```

        <span class="error" id="comentariosError" aria-
live="polite"></span>
    </div>
    <br>

    <!-- Campo de archivo -->
    <div>
        <label for="archivo">Subir archivo:</label>
        <input type="file" id="archivo" name="archivo" aria-
required="true">
        <span class="error" id="archivoError" aria-live="polite"></span>
    </div>
    <br>

    <!-- Botón de Envío -->
    <input type="submit" value="Enviar">
</form>

<script>
    // Añadir un event listener al formulario para validar los campos
antes de enviar
    document.querySelector('form').addEventListener('submit',
function(event) {
        let isValid = true; // Bandera para verificar si todos los campos
son válidos

        // Validación del campo nombre
        const nombre = document.getElementById('nombre');
        const nombreError = document.getElementById('nombreError');
        if (nombre.value.trim() === '') { // Si el campo está vacío
            nombreError.textContent = 'El nombre es obligatorio.'; //
Mostrar mensaje de error
            isValid = false; // Marcar el formulario como inválido
        } else {
            nombreError.textContent = ''; // Limpiar el mensaje de error
si el campo es válido
        }

        // Validación del campo comentarios
        const comentarios = document.getElementById('comentarios');

```

```

        const comentariosError =
document.getElementById('comentariosError');

        if (comentarios.value.trim() === '') { // Si el campo está vacío
            comentariosError.textContent = 'Los comentarios son
obligatorios.'; // Mostrar mensaje de error

            isValid = false; // Marcar el formulario como inválido
        } else {
            comentariosError.textContent = ''; // Limpiar el mensaje de
error si el campo es válido
        }

        // Validación del campo archivo
        const archivo = document.getElementById('archivo');
        const archivoError = document.getElementById('archivoError');
        if (archivo.value.trim() === '') { // Si el campo está vacío
            archivoError.textContent = 'Debe subir un archivo.'; //
Mostrar mensaje de error

            isValid = false; // Marcar el formulario como inválido
        } else {
            archivoError.textContent = ''; // Limpiar el mensaje de error
si el campo es válido
        }

        // Si algún campo es inválido, prevenir el envío del formulario
        if (!isValid) {
            event.preventDefault();
        }
    });
</script>
</body>
</html>

```

En este ejemplo:

Accesibilidad

1. Uso de etiquetas <label>: Cada campo de entrada tiene una etiqueta asociada para mejorar la accesibilidad.
2. Atributos ARIA: Se utilizan atributos ARIA como aria-required y aria-labelledby para proporcionar información adicional a las tecnologías de asistencia.

Usabilidad

1. Campos obligatorios: Se indica claramente cuáles son los campos obligatorios.
2. Retroalimentación de errores: Se incluye un script de JavaScript para validar los campos obligatorios y proporcionar retroalimentación en tiempo real si hay errores. Los mensajes de error se muestran en rojo y se utilizan atributos ARIA para anunciar los errores a las tecnologías de asistencia.

Este formulario incluye:

1. Un campo de texto para ingresar el nombre.
2. Un área de texto multilínea para comentarios.
3. Un campo de archivo para subir un archivo.
4. Un botón de envío.
5. Validación y retroalimentación de errores en tiempo real.