



Validación de Formularios en PHP

Objetivo: Aprender a validar y procesar formularios de forma segura y eficiente.



Introducción a la Validación de Formularios

Importancia:

- Asegurar la integridad de los datos.
- Evitar ataques como inyección de SQL y XSS.

Paso 1: Crear el Formulario HTML

Formulario básico con campos para nombre y email:

```
<!-- Formulario HTML -->
<form method="POST" action="formulario.php">
  <label for="nombre">Nombre:</label>
  <input type="text" name="nombre" id="nombre" value="">
  <label for="email">Email:</label>
  <input type="email" name="email" id="email" value="">

  <input type="submit" name="submit" value="Enviar">
</form>
```

Paso 2: Inicializar Variables

Inicializar las variables antes de validar o usar los datos:

Esto evita errores por variables indefinidas al procesar o mostrar el formulario.

```
//inicializamos variables  
$nombre = '';  
$email = '';  
$errores = [];
```

Paso 3: Comprobar el Método y Submit

Comprobar si el formulario fue enviado con el método POST y si se ha pulsado el botón "submit":

```
//comprobar el método y si se ha pulsado el botón submit  
if ($_SERVER["REQUEST_METHOD"] == "POST" && isset($_POST['submit'])) {  
    // el formulario fue enviado  
}
```

Paso 4: Recoger los Datos de los Campos y mostrarlos en el formulario

Recoger los datos usando `$_POST` y escapar las variables cuando se muestren en el HTML con `htmlspecialchars()`:

```
// Recoger los datos
$nombre = $_POST['nombre'] ?? '';
$email = $_POST['email'] ?? '';
```

El operador `??` asigna un valor vacío (") si el campo no ha sido enviado.

Usamos `htmlspecialchars()` para escapar los valores al mostrar en el formulario y prevenimos la ejecución de código malicioso.

```
<!-- Formulario HTML -->
<form method="POST" action="formulario.php">
  <label for="nombre">Nombre:</label>
  <input type="text" name="nombre" id="nombre" value="<?php echo htmlspecialchars($nombre); ?>">
  <label for="email">Email:</label>
  <input type="email" name="email" id="email" value="<?php echo htmlspecialchars($email); ?>">

  <input type="submit" name="submit" value="Enviar">
</form>
```

P. Luyot-24

Paso 5: Validar los Campos

Validar el campo **nombre** e **email**

```
// Validar el campo nombre
if (empty($nombre)) {
    $errores['nombre'] = "El nombre es obligatorio.";
}

// Validar el campo email
if (empty($email)) {
    $errores['email'] = "El email es obligatorio.";
} elseif (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
    $errores['email'] = "Formato de email inválido.";
}
```

Cargamos el array asociativo `$errores` con el error particular

Paso 6: Sanitizar los Datos

Limpiar los datos recibidos antes de usarlos:

```
//Función que satiniza una cadena de caracteres
function test_input($data) {
    $data = trim($data);           // Elimina espacios al inicio y al final
    $data = stripslashes($data);  // Elimina barras invertidas
    $data = htmlspecialchars($data); // Convierte caracteres especiales
    return $data;
}
// Sanitizar los datos
$nombre = test_input($nombre);
$email = test_input($email);
```

Paso 7: Procesar los Datos si NO hay Errores

Si no hay errores, proceder con el procesamiento de los datos

```
// si no existen estos errores
if (empty($errores['nombre']) && empty($errores['email'])) {
    //procesamos la información
}

//podríamos poner también
if (empty($errores)){
    //procesamos la información
}
```

El procesar la información puede incluir:

- Guardar los datos en una base de datos,
- Enviar un correo electrónico de confirmación o notificación,
- Ejecutar lógica de negocio, como cálculos o interacciones con otros sistemas,
- Redireccionar al usuario a otra página (por ejemplo, una página de confirmación).
- etc..

Paso 8: Mostrar los mensajes de error en el formulario

Añadimos el error si existe junto al campo del formulario

```
<!-- Formulario HTML -->
<form method="POST" action="formulario.php">
  <label for="nombre">Nombre:</label>
  <input type="text" name="nombre" id="nombre" value="<?php echo htmlspecialchars($nombre); ?>">
  <!-- mensaje de error de validación del nombre -->
  <span><?php echo $errores['nombre'] ?? ''; ?></span>

  <label for="email">Email:</label>
  <input type="email" name="email" id="email" value="<?php echo htmlspecialchars($email); ?>">
  <!-- mensaje de error de validación del email -->
  <span><?php echo $errores['email'] ?? ''; ?></span>

  <input type="submit" name="submit" value="Enviar">
</form>
```

Usamos `htmlspecialchars()` para escapar los valores al mostrar en el formulario y prevenimos la ejecución de código malicioso.

P. Luyot-24

Código completo

```
1  <?php
2  //inicializamos variables
3  $nombre = '';
4  $email = '';
5  $errores = [];
6
7  //Función que satiniza una cadena de caracteres
8  function test_input($data)
9  {
10     $data = trim($data);          // Elimina espacios al inicio y al final
11     $data = stripslashes($data); // Elimina barras invertidas
12     $data = htmlspecialchars($data); // Convierte caracteres especiales
13     return $data;
14 }
15 //comprobar el método y si se ha pulsado el botón submit
16 if ($_SERVER["REQUEST_METHOD"] == "POST" && isset($_POST['submit'])) {
17     // Recoger los datos
18     $nombre = $_POST['nombre'] ?? '';
19     $email = $_POST['email'] ?? '';
20
21     // Validar el campo nombre
22     if (empty($nombre)) {
23         $errores['nombre'] = "El nombre es obligatorio.";
24     }
25     // Validar el campo email
26     if (empty($email)) {
27         $errores['email'] = "El email es obligatorio.";
28     } elseif (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
29         $errores['email'] = "Formato de email inválido.";
30     }
```

Código completo

```
31
32 // Sanitizar los datos
33 $nombre = test_input($nombre);
34 $email = test_input($email);
35
36 // si no existen estos errores
37 if (empty($errores['nombre']) && empty($errores['email'])) {
38     //procesamos la información
39 }
40 //podríamos poner también
41 if (empty($errores)) {
42     //procesamos la información
43 }
44 }
45 ?>
46
47 <!-- Formulario HTML -->
48 <form method="POST" action="formulario.php">
49     <label for="nombre">Nombre:</label>
50     <input type="text" name="nombre" id="nombre" value="<?php echo htmlspecialchars($nombre); ?>">
51     <!-- mensaje de error de validación del nombre -->
52     <span><?php echo $errores['nombre'] ?? ''; ?></span>
53
54     <label for="email">Email:</label>
55     <input type="email" name="email" id="email" value="<?php echo htmlspecialchars($email); ?>">
56     <!-- mensaje de error de validación del email -->
57     <span><?php echo $errores['email'] ?? ''; ?></span>
58
59     <input type="submit" name="submit" value="Enviar">
60 </form>
61
62
```