**UNIR - Universidad Internacional de La Rioja**

**Maestría en Dirección e Ingeniería de Sitios Web**

Curso:

Computación Cliente y Servidor

Actividad individual:

Realización de una Página Web Utilizando PHP

Alumno: Luis de la Garza González

Matrícula: 5589438-499539

24 de junio de 2024

**ÍNDICE**

[**Introducción** 3](#_Toc169894443)

[**Objetivos** 4](#_Toc169894444)

[**Descripción** 4](#_Toc169894445)

[**Utilización de etiquetas HTML5.** 5](#_Toc169894446)

[**Campos de por lo menos 5 tipos.** 5](#_Toc169894447)

[**Funcionalidad, comprobaciones en PHP** 5](#_Toc169894448)

[**Manejo de errores** 6](#_Toc169894449)

[**Explicación del código** 8](#_Toc169894450)

[**Conclusiones** 10](#_Toc169894451)

Actividad 1.- Realización de una página web utilizando PHP

**Introducción**

El sitio Web implementado para la presente práctica se puede acceder en:

<https://yellow-mink-580191.hostingersite.com/index.html>

La codificación se anexó a la tarea como el archivo zip:

formularioPHP-main.zip

El cual incluye la siguiente estructura y archivos:

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

**Objetivos**

Practicar habilidades de programación en PHP con los conocimientos aprendidos durante la asignatura.

**Descripción**

Se Realiza una página web utilizando PHP con un formulario que se debe validar.

Los requisitos son los siguientes:

* El formulario utiliza etiquetas HTML5.
* El formulario contiene campos de por lo menos 5 tipos diferentes (por ejemplo, texto, fecha, contraseña, etc.).
* El formulario tiene la funcionalidad necesaria para que existan al menos dos comprobaciones en PHP, para que no sean posibles en HTML5. Por ejemplo, se pueden validar correos electrónicos, direcciones de sitios web, formato del DNI español con su letra, etc.

Cuando se entregue el formulario se deberá:

* Si había algún campo erróneo, informar de los errores detectados y redirigir al usuario al formulario para que lo pueda cumplimentar debidamente.
* Si todos los campos eran correctos, generar un informe con las validaciones realizadas y mostrarlo por pantalla.
* Al final del código se incluirá un comentario de no más de 200 palabras que explique el funcionamiento del código implementado.

**Utilización de etiquetas HTML5.**

Ver archivo formato.php (incluido en el archivo formularioPHP-main.zip)

**Campos de por lo menos 5 tipos.**

Se utilizaron campos de: texto, email, teléfono, fecha y url.

**Funcionalidad, comprobaciones en PHP**

Todos los campos fueron validados (texto, email, teléfono, fecha y url)

Ver archivo validación.php (incluido en el archivo formularioPHP-main.zip)

**Manejo de errores**

• Si hubiera algún campo erróneo, informar de los errores detectados y redirigir al usuario al formulario para que lo pueda cumplimentar debidamente.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamenteLa siguiente imagen muestra los errores detectados.

• Si todos los campos son correctos, generar un informe con las validaciones realizadas y mostrarlo por pantalla.

Interfaz de usuario gráfica, Texto, Aplicación, Word, Correo electrónico

Descripción generada automáticamenteLa siguiente imagen muestra el informe generado al final del formulario.

**Explicación del código**

Explicación del funcionamiento del código implementado para la validación.

El siguiente texto se encuentra comentado al final del archivo validación.php (incluido en el archivo formularioPHP-main.zip).

Inicialmente definimos e inicializamos las variables tanto para mostrar errores como para los datos del libro que recibimos para validación.

Para mostrar errores de validación

$autorErr = "";

$emailErr = "";

$telefonoErr = "";

$libroErr = "";

$fechaErr = "";

$resumenErr = "";

$urlErr = "";

Para los datos del libro

$autor = "";

$email = "";

$telefono = "";

$libro = "";

$fecha = "";

$resumen = "";

$url = "";

Validación de los campos de entrada recibidos con el método POST

Únicamente explicaré la validación de un campo, en este caso el nombre del autor, ya que todas las validaciones son similares

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

// Validación del nombre del autor

// Primero validamos que el campo no sea vacío

if (empty($\_POST["autor"])) {

$autorErr = "El nombre del autor es requerido";

} else {

$autor = input\_data($\_POST["autor"]);

// Checamos que el nombre del autor contenga únicamente letras del alfabeto, espacios, números y guiones bajos.

if (!preg\_match("/^[a-zA-Z0-9\_\sñáéíóúÁÉÍÓÚ]\*$/", $autor)) {

$autorErr = "Únicamente letras del alfabeto, números y guión bajo son permitidos para el nombre del autor";

}

}

...

}

// Para manejar espacios en blanco y caracteres especiales en los datos utilizamos la siguiente función:

function input\_data($data)

{

$data = trim($data); // para remover espacios al inicio de los datos

// Para evitar ataques Cross-Site Scripting (XSS) debemos convertir caracteres especiales en sus correspondientes entidades HTML, ejemplo - "&" -> "&"

$data = htmlspecialchars($data);

return $data;

}

**Conclusiones**

PHP es un lenguaje de scripting del lado del servidor de código abierto que es muy utilizado para el desarrollo web. Ha sido referencia para los servidores web durante más de 15 años.

Algunas ventajas de PHP son:

* Es multiplataforma, se ejecuta en todas las plataformas, ya sea Mac, Windows o Linux.
* Es de código abierto.
* Es fácil de aprender.
* Puede conectar fácilmente a MySQL, Postgress, MongoDB o cualquier otra base de datos.
* Tiene una comunidad en línea en la que apoyarse.

Por todo lo anterior, considero que PHP sigue siendo una buena opción para la programación del lado del servidor.