

## Laboratorio 5: Formularios en PHP

### Objetivo de la práctica

Asentar los conocimientos sobre validación de formularios en servidor con PHP.

### Enunciado

La validación de datos de un formulario en el cliente utilizando JavaScript (que este curso veremos posteriormente) **no es en ningún caso suficiente** para asegurarnos que los datos recibidos en el servidor van a ser válidos. La razón es que cualquier usuario con los conocimientos suficientes podría modificar el código JavaScript o enviar directamente los datos al servidor sin necesidad de usar un navegador web. Por lo tanto, la validación de los datos provenientes de un formulario en el servidor es una tarea indispensable para garantizar que los datos que llegan son correctos y para poder evitar posibles ataques a nuestra máquina. En esta práctica vamos a retomar el formulario de prácticas anteriores y vamos a implementar su validación en el servidor usando PHP.

Antes de validar los datos del formulario en servidor, se recomienda repasar la práctica anterior en la que se mostraban el formulario de alta de usuario (Figura 1) y se pedía mostrar los datos recibidos por el servidor.

*Los campos obligatorios están marcados con \**

Datos personales

NIF\*12345678X

Nombre:\*

Apellidos:

Perfil: ☐ Alumno ☐ PDI ☐ PAS

Fecha de nacimiento: dd/mm/aaaa

Email:\*usuario@dominio.extens

Datos de usuario

Nickname:

Password:\*Mínimo 8 caracteres ent

Confirmar Password:Confirmación de contras

¿Cuáles son tus géneros literarios favoritos?Ciencia Ficción

Dirección

Calle/Avda.:\*

Provincia:\*

Enviar

Figura 1: Formulario de alta de usuario

## Laboratorio 5: Formularios en PHP

### 1. Validación en servidor

Valide los datos del formulario mostrado en la Figura 1 en el servidor. Utilice para implementar esta validación el esquema recomendado en la clase de teoría. Se deberán validar los siguientes aspectos en el archivo **accion\_alta\_usuario.php** que se facilita en la práctica:

- NIF, email y nombre no son vacíos.
- NIF debe constar de 8 números y una letra mayúscula.
- La dirección (Calle y Provincia) no deben ser vacíos.
- El email es una dirección de correo válida.
- El género literario escogido debe estar entre los predefinidos en la aplicación.
- La contraseña no debe ser vacía. Debe tener al menos 8 caracteres, con al menos un dígito y letras mayúsculas y minúsculas.
- La confirmación de la contraseña debe coincidir con la contraseña.

#### NOTAS:

- Para la validación del email en servidor podemos utilizar la función predefinida en PHP:  
`filter_var($email, FILTER_VALIDATE_EMAIL)`

Que devuelve cierto en caso de que el email tenga un format correcto, y falso en caso contrario.

### 2. Informar de los errores

Si una o varias de las condiciones antes mencionadas no se cumplen hay que informar al usuario, redirigiéndolo a una página (en nuestro caso, utilizamos el propio formulario **form\_alta\_usuario.php**) donde se detallen los errores detectados y se incluya un enlace de regreso al formulario de alta (ver Figura 2).

Gestión de biblioteca: Us X

localhost/Lab5/IISS-Biblioteca/form\_alta\_usuario.php

UNIVERSIDAD DE SEVILLA

## Biblioteca

### Gestión de usuarios y préstamos de libros

**Errores en el formulario**

- El NIF no puede estar vacío
- El nombre no puede estar vacío
- El email no puede estar vacío
- Contraseña no válida: debe tener al menos 8 caracteres
- La dirección no puede estar vacía
- La provincia no puede estar vacía

Figura 2: Página del formulario de alta con información sobre los errores

## Laboratorio 5: Formularios en PHP

Para ello utilizamos la variable de sesión para almacenar los datos que necesitemos (mensajes de error), y en la página del formulario, comprobar si en la sesión existe una variable de errores para mostrárselos al usuario en un `<div>` dedicado a ello.

### 3. Uso de la variable de sesión para recargar datos del formulario

Utilizar la variable de sesión para almacenar los datos del formulario y poder recuperarlos en caso de volver al formulario por algún error de validación. Procederemos de la siguiente manera:

- En el procesado del formulario (**accion\_alta\_usuario.php**) debemos comprobar que el formulario se ha enviado (mediante la función *isset*). En caso afirmativo, debemos almacenar los datos del mismo en una variable (en nuestro caso la hemos llamado `$_SESSION["formulario"]`) dentro de la sesión antes de proceder a la validación del mismo.
- En el formulario **form\_alta\_usuario.php**, comprobamos si existe una variable *"formulario"* en la sesión:
  - Si no existe la variable en la sesión, la creamos de forma que contenga valores por defecto para cada uno de los campos del formulario.
  - Si existe, se recogen sus valores en una variable local.
- Una vez hecho esto, se utilizan los valores de la variable *"formulario"* para inicializar los campos del formulario, utilizando el atributo *value* si son atributos atómicos.