

EJERCICIO 5: FORMULARIO DE ENTRADA DE DATOS

Alumnos: Laura Alonso, David Navarro

Profesor: Jairo García

Asignatura: Desarrollo Web Servidor

Fecha: 22 de noviembre de 2021

Contenido

EJERCICIO N°5	2
Enunciado	2
Descripción del proceso	2

EJERCICIO N°5

Enunciado

Desarrolla en parejas un formulario de entrada de datos personales de una empresa de la temática queelijáis (no se puede repetir) que incluya los campos: nombre y apellidos, dirección, ciudad, provincia (un select con todas las provincias de España junto con sus códigos de provincia que se cargan de un fichero), código postal, teléfono, email, password y web. Si para las más complicadas se utilizan recursos web perfecto, siempre que se comenten adecuadamente.

1. Escribe una función que devuelva un mensaje en caso de que el usuario escriba números en el campo de entrada de nombre y apellidos o en el de ciudad.
2. Escribe una función que devuelva un mensaje en caso de que el usuario escriba letras en el campo de entrada de teléfono o en el de código postal.
3. Modifica la anterior para que devuelva un mensaje en caso de que el usuario escriba más de 9 números en el teléfono.
4. Modifica la anterior para que devuelva un mensaje en caso de que el usuario escriba más de 5 números en el código postal.
5. Modifica la anterior para que devuelva un mensaje si los dos primeros dígitos del código de la provincia seleccionada no se corresponden con los dos primeros del código postal.
6. Escribe una función que compruebe que la dirección de email es correcta.
7. Escribe una función que compruebe si un input de tipo password cumple los siguientes requisitos:
 - Entre 8 y 16 caracteres
 - Al menos un número
 - Al menos una mayúscula
 - Al menos una minúscula
 - Al menos un carácter extraño
8. Escribe una función que compruebe que la dirección web es correcta.

Descripción del proceso

1. En nuestro trabajo tenemos presente varios archivos, el html principal llamado index.php, el archivo de texto llamado Provincias.txt, el php complementario llamado logica.php y el CSS llamado CSSFormulario.css. El formulario principal lo hemos organizado en una tabla, y en la parte de arriba hemos enlazado el archivo Provincias.txt con las provincias a su input con un function compuesto de un foreach explicado a continuación:

```
function provincias()
{
    //Para ello nos creamos la variable a utilizar
    $provincias = [];
    //Y utilizamos la funcion file_get_contents para extraer los datos
    $string = file_get_contents("Provincias.txt");

    $array = explode("\n", $string);
    foreach ($array as $fila) {
        $item = explode(" ", $fila);
        $provincias[] = [
            'numero' => $item[1],
            'nombre' => str_replace("_", " ", $item[0])
        ];
    }
    //Devolvemos la variable que almacena los datos
```

```

        return $provincias;
    }

```

2. Para que esa función se muestre en el option, se combinan de la siguiente manera:

```

<td><label>Provincia: </label></td>
<td><select name="Provincia">
    <?php
        //utilizamos un foreach para evitar crearnos muchas etiquetas
    option
        foreach (provincias() as $provincias){
            ?> <option> <?php echo $provincias['nombre']; ?></option>
        }
    </select>
</td>

```

3. Para comprobar los elementos que nos piden los ejercicios, creamos una variable booleana llamada "correcto" y le ponemos de valor true por defecto. Con esto se hace un if que nos especifiquen los mensajes que han de poner si cumplen o no las indicaciones de los ejercicios.

```

$correcto = true;
if($correcto == true ){
    ?>
    <div class ="final"> <h1 class ="b">Formulario enviado correctamente <br><br>
    <small>Se te redireccionará a la pagina del formulario en unos
    instantes</small></h1> <br> <?php
        header("refresh:3;url= ../index.php"); ?>

    </div>
    <?php
        //Si ha habido fallos se mostraran y se tendrá la opción de volver ha
        introducir valores
    }else{
        ?> <h1>Ha ocurrido un error</h1>
        <a href=" ../index.php"> Pulsa aqui para rellenar el formulario de nuevo</a>
    <?php
    }
}

```

4. Para el primer ejercicio se crea la función busca_numeros que compruebe si hay números en los campos. Se implementa dicha función mediante un isset y varios ifs dentro de este mismo.

```

function busca_numeros($enter) {
    $regex = '/[0-9]+/';
    return preg_match($regex,$enter);
}

if (isset($_POST['enviar'])) {
    //Comprueba que el nombre no tenga números
    if(busca_numeros($_POST['nombre']) != 0){
        $correcto = false;
        echo "<p class ='err'>El nombre no puede contener números. </p>";
    }

    //Comprueba que el apellido no tenga números
    if(busca_numeros($_POST['apellidos']) != 0){
        $correcto = false;
        echo "<p class ='err'> Los apellidos no pueden contener números.
    </p>";
    }

    //Comprueba que la ciudad no tenga números
    if(busca_numeros($_POST['ciudad']) != 0){
        $correcto = false;
        echo "<p class ='err'> La ciudad no puede contener números. </p>";
    }
}

```

5. En el segundo ejercicio se crea un define RELET para meter el rango de caracteres que se quieren poner a un campo. Con esto hecho se crean las funciones checkTelefono y checkCP para especificar qué tiene que pasar cuando metamos letras en esos campos. Al final se llaman a las funciones por sus respectivos campos.

```
function checkTelefono($telf) {
    //llamamos a la variable global correcto
    global $correcto;
    //Y comprobamos si el valor introducido cumple los parámetros si no
    mandamos un mensaje de error y aplicamos correcto a falso
    if(preg_match(RELET,$telf)!=0) {
        echo " <p class = 'err'> El teléfono no puede contener
letras.</p>";
        $correcto = false;
    }
    function checkCP($cp) {
        //llamamos a la variable global correcto
        global $correcto;
        // creamos una variable que almacene el valor de $_POST['cp'] con un
valor string
        $cpf = strval($_POST['cp']);

        //Y comprobamos si el valor introducido cumple los parámetros si no mandamos
un mensaje de error y aplicamos correcto a falso
        if(preg_match(RELET,$cp)!=0) {
            echo " <p class = 'err'> El código postal no puede contener letras </p>";
            $correcto = false;
        }

        checkTelefono($_POST['telefono']);
        checkCP($_POST['cp']);
    }
}
```

6. Para el tercer y cuarto ejercicios, se modifican las funciones del teléfono y del código postal para que en el caso de exceder el límite que nos piden los ejercicios nos ponga un mensaje que nos indique el error y, en el caso del código postal, que compruebe si los 2 primeros valores del número introducido coincide con el código de provincia en el documento Provincias.txt.

```
function checkTelefono($telf) {
    //llamamos a la variable global correcto
    global $correcto;
    //Y comprobamos si el valor introducido cumple los parámetros si no
    mandamos un mensaje de error y aplicamos correcto a falso
    if(preg_match(RELET,$telf)!=0) {
        echo " <p class = 'err'> El teléfono no puede contener
letras.</p>";
        $correcto = false;
    }
    else{
        //Ahora comprobamos que la longitud del valor sea exactamente igual
a nueve de no ser así mandamos un mensaje de error y aplicamos correcto a
falso
        if(strlen($telf)!=9) {
            echo " <p class = 'err'> Tienes que introducir 9 números en el
teléfono. </p>";
            $correcto = false;
        }
    }

    function checkCP($cp) {
        //llamamos a la variable global correcto
        global $correcto;
        // creamos una variable que almacene el valor de $_POST['cp'] con un
valor string
    }
```

```

$cpf = strval($_POST['cp']);

//Y comprobamos si el valor introducido cumple los parámetros si no mandamos un
mensaje de error y aplicamos correcto a falso
if(preg_match(RELET,$cp) !=0) {
    echo " <p class = 'err'> El código postal no puede contener letras
</p>";
    $correcto = false;
} else {
    //Ahora comprobamos que la longitud del valor sea exactamente igual a
nueve de no ser así mandamos un mensaje de error y aplicamos correcto a falso
    if (strlen($cp) != 5) {
        echo " <p class = 'err'> Tienes que introducir 5 números en el código
postal. </p>";
        $correcto = false;
    } else {
        //Ahora comprobaremos que el código de provincia corresponde con el
principio del código postal
        foreach (provincias() as $provincias) {
            //primero debemos saber que provincia se ha marcado
            if ($provincias['nombre'] == $_POST['Provincia']) {
                //Y después comprobar si se cumple o no las condiciones

                if ($provincias['numero'] != substr($cpf, 0, 2)) {
                    echo " <p class = 'err'> el código postal no corresponde con la
Provincia seleccionada </p> ";
                    $correcto = false;
                }
            }
        }
    }
}

```

7. Para el sexto punto nos pide que se compruebe si el email introducido cumple con el formato email para ello utilizamos una función ya creada en PHP llamada filter que nos permite filtrarla directamente.

```

function email($enter){
    global $correcto;
    if( filter_var($enter, FILTER_VALIDATE_EMAIL) == 0){
        $correcto = false;
        echo '<p class = "err"> Email no válido </p>';
    }
}

```

8. Para el séptimo punto se nos pide que al introducir una contraseña se cumplan las siguientes condiciones:

Al menos un número	Al menos una mayúscula
Al menos una minúscula	Al menos un carácter especial
Mínimo 8 caracteres	máximo 16 caracteres

Para ello creamos el patrón que haga cumplir estos requisitos

```
$regex = "/((?=.*?[A-Z])(?=.*?[a-z])(?=.*?[0-9])(?=.*?[^\A-Za-z0-9])).{8,16}/";
```

Y hacemos las comprobaciones y alertamos al usuario si ha habido algún problema

```
function contrasena($enter){
    global $correcto;
    $regex = "/((?=.*?[A-Z])(?=.*?[a-z])(?=.*?[0-9])(?=.*?[^A-Za-z0-9])){8,16}/";
    if (preg_match($regex,$enter) != 1) {
        $correcto = false;
        echo "<p class='err'> Contraseña no válida. </p>";
    }
}
```

9. Y por ultimo creamos la función para comprobar el formato de una web y alerte al usuario si ha habido algún fallo

//Esta función comprueba si el valor introducido cumple con las reglas de formato de una página web

```
function web($enter){
    global $correcto;
    $regex = '#((https?|ftp)://(\S*?\.\S*))([\s\[\]\{\},;"\':<|\\.\s|$)#i';
    if (preg_match($regex,$enter) == 0) {
        $correcto = false;
        echo '<p class="err"> Web no válida. </p>';
    }
}
```