

Desarrollo de Software VII

Laboratorio # 18—Prof. Regis Rivera

Objetivo: Introducción a expresiones regulares en php (back-end) y HTML5 (front-end)

Parte I – RegEx desde Back-end

1. Crear un formulario web en PHP para registrar un nuevo usuario donde se solicite lo siguiente:

- Inputs tipo text:
 - Nombre
 - Apellido
 - Email
 - Contraseña
 - Repetir Contraseña
 - IP actual del equipo
- Input tipo submit:
 - Boton de “Registrar Usuario”

2. Validar los datos del formulario (con las funciones dadas en el punto 3), todos los campos son requeridos:

- Si los datos están correctos, enviar una alerta que diga “Usuario registrado exitosamente” (en este caso, simularemos quedaron registrados en una base de datos, no es necesario elabore procedimientos almacenados e integraciones con MySQL ya que solo es un ejemplo del uso de expresiones regulares)
- Si los datos no están correctos, enviar mensaje al usuario indicando que el dato esta incorrecto (a un costado del campo input o debajo, como los que usted ha visto en formularios de registro que ha utilizado en su experiencia)

3. Validaciones (copiar/pegar funciones):

A) Que la contraseña sea segura, utilizando esta función:

```
function verificar_email($email)
{
    if(preg_match("/^([a-zA-Z0-9])+([a-zA-Z0-9\._-])*@([a-zA-Z0-9_-])+([a-zA-Z0-9\._-]+)$/",$email))
    {
        return true;
    }
    return false;
}
```

B) Que la dirección de email sea válida, utilizando esta función:

```
function verificar_password_strength($password)
{
    if (preg_match("/^.*(?=.{8,})(?=.*\d)(?=.*[a-z])(?=.*[A-Z]).*$/", $password))
        echo "Su password es seguro.";
    else
        echo "Su password no es seguro.";
}
```

C) Que la dirección de IP sea válida, utilizando esta función:

```
function verificar_ip($ip)
{
    return preg_match("/^([1-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])" .
    "(\.([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])){3}$/", $ip );
}
```

Parte II – RegEx desde Front-end

4. Utilice el atributo pattern dentro del input. El atributo pattern contendrá la expresión regular (misma utilizada en back-end, por ejemplo).

Además, recordar HTML5 incorporó nuevos inputs (además del text, button, submit) como email, number, password entre otros.

Ejemplo:

```
<!DOCTYPE html>
<html>
<body>

<h1>El atributo Pattern en el tipo Input</h1>

<p>Un formulario con un campo de correo electrónico que debe estar en el siguiente orden:
caracteres@caracteres.dominio (caracteres seguidos de un signo @, seguidos de más caracteres y luego un ".".
Después del signo ".", agregue al menos 2 letras de la a a la z.</p>

<form action="/action_page.php">
  <label for="email">Correo electrónico:</label>
  <input type="email" id="email" name="email" pattern="[a-z0-9_\.-]+@[a-z0-9-]+\.[a-z]{2,}$">
  <input type="submit">
</form>

</body>
</html>
```

El atributo Pattern en el tipo Input

Un formulario con un campo de correo electrónico que debe estar en el siguiente orden: caracteres@caracteres.dominio (caracteres seguidos de un signo @, seguidos de más caracteres y luego un ".". Después del signo ".", agregue al menos 2 letras de la a a la z.

Correo electrónico:

 Incluye "@" en la dirección de correo electrónico. En "no valido" falta un símbolo "@".