

REGISTRO DE USUARIOS EN PHP Y MYSQL

En que consiste? y ¿para que sirve? muy sencillo, en algunas webs como por ejemplo esta queremos interactuar con los usuarios, deseamos que estos puedan enviar scripts al sistema para ayudar a crecer la coleccion de scripts que disponemos, pero deseamos llevar un control sobre los autores de los scripts asi que no cualquiera puede enviar scripts, para hacerlo tienes que registrarte y hacer el respectivo "ingreso" para que el sistema te reconosca y te deje enviar scripts, archivos y articulos como este, de otra manera no puedes hacer ninguna de estas acciones.

¿Pero como se logra esto?, se hace teniendo en una base de datos una tabla de usuarios(Podria hacerse con archivos, aunque seria un poco inseguro) y por medio de archivos PHP y manejo de sesiones(session) o por "cukis"(cookies).

En este articulo explicare como crear un sistema de registro y reconocimiento de usuarios teniendo una tabla en una base de datos MySql y por medio de "cukis"(COOKIES) para nuestra web de una manera sencilla.

Para aprender un poco sobre las cookies(algo basico) deberas de leerte primero que todo el articulo [que he enviado al respecto\(http://www.miscodigos.com/JLMCodigos/articulo.php?id=5\)](http://www.miscodigos.com/JLMCodigos/articulo.php?id=5), de esta manera entenderas mucho mejor el funcionamiento de este sistema, si ya lo has leido o tienes conocimiento previo sobre cookies continua leyendo, de lo contrario leelo ahora ya que no explicare nada que tenga que ver con dicho tema(cookies) en este articulo.

A la hora de crear un sistema de registro de usuarios siempre pensamos en lo mismo:

registro + login + reconocimiento

Queremos que los usuarios se registren dejando sus datos y puedan acceder a algunas zonas de la pagina y sea reconocido por nuestra pagina(por ejemplo "bienvenido jlmm" o "bienvenido visitante" en caso de no estar ingresado(o registrado) en el sistema).

Muchos piensan en crear un sistema de usuarios por que les parece "bonito", pero hay pocos que realmente lo necesitan(lo digo por experiencia con dichas personas), y cuando les haces las dos preguntas:

- 1- ¿que beneficios tienen los usuarios registrados con respecto a los demas?
- 2- ¿que zonas de tu web quieres que sean de acceso restringido y PORQUE?

La mayoría no es capaz de responder la primera pregunta, y es que antes de aventurarnos a crear un sistema de usuarios para nuestra web debemos de pensar si realmente vale la pena, ya que las tematicas de la web y sus contenidos no siempre son adecuados para tener un sistema de registro de usuarios.

Para ello respondemos las 2 preguntas, en el caso de misCODIGOS.com serian:

1- ¿que beneficios tienen los usuarios registrados con respecto a los demas?

R://Posibilidad de enviar scripts, articulos y archivos al sistema, al igual que disponibilidad de un panel de control entre otros.

2- ¿que zonas de tu web quieres que sean de acceso restringido y PORQUE?

R://Quiero que solo los usuarios que esten registrados puedan enviar scripts/articulos/archivos, ya que asi puedo llevar un mejor control sobre los mismos y prevenir el envio indiscriminado de scripts/articulos/archivos al sistema por parte de los usuarios.

Ya tenemos nuestras razones para crear nuestro sistema de registro de usuarios, claro que pueden haber muchas razones dependiendo del tipo de web, podrian ser por ejemplo para que solo usuarios registrados puedan bajar software/musica/videos/etc, para restringir el acceso a ciertas zonas de la web "obligando" al usuario a registrarse para poder "ver" dicha zona, etc...

Siempre teniendo en cuenta que las razones sean validas y que no molesten al usuario, ya que en mi caso hace algunos meses me encuentre con una web de "scripts" la cual para poder ver los "scripts" tenias que registrarte, obviamente lo que hice fue salir de esa web y olvidarla(por ello no les puedo dar la direccion), tenemos que tener muy en cuenta que zonas restringir a usuarios registrados ya que el usuario en general prefiere la comodidad y en este caso que les comento de webs con recursos y scripts para webmasters hay muchas(pero ninguna como esta :-P) y hay que buscar siempre la comodidad de los usuarios.

Ahora a lo que vinimos, crearemos nuestro sistema de registro de usuario, lo primero que debemos pensar es que datos debera introducir el usuario para registrarse, como siempre deberan de haber aunque sea dos campos basicos:

nick

password

Lo mas comun para identificar a los usuarios es por medio de nick/password, aunque tambien podria ser email/password, para este articulo utilizare el primero (nick/password).

Ya sabemos que nuestra tabla de "usuarios" tendra esos dos campos:

```
CREATE TABLE usuarios (  
id bigint(7) NOT NULL auto_increment,  
nick char(100) NOT NULL,  
password char(100) NOT NULL,  
KEY id (id))
```

¿Pero son suficientes?, tambien seria bueno tener el email de los usuarios en caso de querer enviar informacion a los usuarios registrados y tener el nombre de dichos usuarios para

personalizar dichos emails o mensajes que le demos al usuario en la pagina(ejemplo "hola jorge luis"), entonces nuestra tabla creceria un poco:

```
CREATE TABLE usuarios (  
id bigint(7) NOT NULL auto_increment,  
nick char(100) NOT NULL,  
password char(100) NOT NULL,  
nombre char(255) default NULL,  
email char(100) default NULL,  
KEY id (id))
```

Listo, ya tenemos la estructura de nuestra tabla de usuarios, es pequeña y sencilla pero para efectos practicos y de aprendizaje es perfecta(eso depende de ti, segun lo que desees saber de tus usuarios).

Ahora viene lo sencillo, crear el sistema de registro de usuarios, un formulario y una aplicacion que agregue al nuevo usuario a la base de datos, para ello crearemos un formulario del tipo:

```
<FORM ACTION="registrar.php" METHOD="post">  
Nick : <INPUT TYPE="text" NAME="nick" SIZE=20 MAXLENGTH=20>  
<BR>  
Email: <INPUT TYPE="text" NAME="email" SIZE=28 MAXLENGTH=100>  
<BR>  
Password: <INPUT TYPE="password" NAME="password" SIZE=28 MAXLENGTH=20>  
<BR>  
Nombre: <INPUT TYPE="text" NAME="nombre" SIZE=28 MAXLENGTH=255>  
<BR>  
<INPUT TYPE="submit" CLASS="boton" VALUE="Registrar">  
</FORM>
```

Y ahora necesitamos una aplicacion que lo procese(registrar.php), por motivos practicos haremos algo sencillo sin demasiadas comprobaciones(dependera de ti mejorarlo si quieres algo mas personalizado), primero el codigo luego la explicacion:

```
<?php
```

```
//AQUI CONEXION O include() DE ARCHIVO DE CONEXION CON BASE DE DATOS.
```

```
function quitar( $mensaje)  
{  
$mensaje = str_replace("<","&lt;",$mensaje);  
$mensaje = str_replace(">","&gt;",$mensaje);  
$mensaje = str_replace("\'", "&#39;",$mensaje);  
$mensaje = str_replace("\'", "&quot;",$mensaje);  
$mensaje = str_replace("\\\\", "&#92;",$mensaje);
```

```

return $mensaje;
}

if(trim($HTTP_POST_VARS["nick"]) != "" && trim($HTTP_POST_VARS["email"]) !=
"")
{
$sql = "SELECT id FROM usuarios WHERE
nick='".quitar($HTTP_POST_VARS["nick"])."'";
$result = mysql_query($sql);
if($row = mysql_fetch_array($result))
{
echo "Error, nick escogido por otro usuario";
}
else
{
$sql = "INSERT INTO usuarios (nick,password,nombre,email) VALUES (";
$sql .= "'".quitar($HTTP_POST_VARS["nick"])."'";
$sql .= ",".quitar($HTTP_POST_VARS["password"])."'";
$sql .= ",".quitar($HTTP_POST_VARS["nombre"])."'";
$sql .= ",".quitar($HTTP_POST_VARS["email"])."'";
$sql .= ")";
mysql_query($sql);
echo "Registro exitoso!";
}
mysql_free_result($result);
}
else
{
echo "Debe llenar como minimo los campos de email y password";
}
mysql_close();
?>

```

Al principio no he echo conexion con la base de datos pero he puesto el comentario `///AQUI CONEXION O include() DE ARCHIVO DE CONEXION CON BASE DE DATOS.`", ya que la conexion depende de ti, si no sabes como hacer una conexion te aconsejo buscarte un manual antes de estar queriendo hacer un sistema como este.

Vemos que he declarado una funcion, llamada `quitar($mensaje)`, la cual recibe el parametro "mensaje", esta funcion se encargara de deshabilitar el codigo html("<" y ">") al reemplazarlos por su respectivo codigo("<" y ">"), ademas de reemplazar el apostrofe(') y las comillas dobles(") por sus respectivos codigos HTML, esto para no tener conflictos con la instruccion sql, y al final retorna la cadena con las nuevas modificaciones en caso de haberlas:

```

function quitar($mensaje)
{

```

```

$mensaje = str_replace("<","&lt;",$mensaje);
$mensaje = str_replace(">","&gt;",$mensaje);
$mensaje = str_replace("\'","&#39;",$mensaje);
$mensaje = str_replace(\'"',\'&quot;',$mensaje);
$mensaje = str_replace("\\\\","&#92;",$mensaje);
return $mensaje;
}

```

Ahora comprobaremos que los campos de nick y password que seran los "requeridos" como minimo para completar el registro no esten vacios, para ello utilizamos la funcion "trim()" la cual quita los espacios al comienzo y al final de la cadena, por lo que en caso de ser una cadena vacia o con solo espacios en blanco retornara "", en caso de ser valido continuamos con el registro, de lo contrario retornamos un mensaje de error:

```

if(trim($HTTP_POST_VARS["nick"]) != "" && trim($HTTP_POST_VARS["password"])
!= "")

```

Ahora suponiendo que las cadenas estaban correctas comprobaremos que el nick seleccionado por el usuario no exista en la base de datos, para ello hacemos una instruccion sql para pedirle a la base de datos que nos retorne la fila donde el nick sea igual al nick introducido con el usuario(quitandole el apostrofe, html y demas con la funcion quitar()):

```

$sql = "SELECT id FROM usuarios WHERE
nick='".quitar($HTTP_POST_VARS["nick"])."'";
$result = mysql_query($sql);

```

En caso de existir dicho nick, osea que nos devuelve una fila:

```

if($row = mysql_fetch_array($result))

```

Mostraremos un mensaje de error al usuario informandole del "incidente", de lo contrario continuaremos con el proceso de registro.

Ahora lo que hacemos es simplemente una instruccion sql donde insertamos en la tabla usuarios el nick,password,nombre y email introducidos por el usuario en el formulario, a todos estos les quitamos el html,apostrofe y demas haciendo uso de nuestra instruccion "quitar()", y mostramos un mensaje de "exito" al usuario:

```

$sql = "INSERT INTO usuarios (nick,password,nombre,email) VALUES (";
$sql .= "'".quitar($HTTP_POST_VARS["nick"])."'";
$sql .= ",".quitar($HTTP_POST_VARS["password"])."'";
$sql .= ",".quitar($HTTP_POST_VARS["nombre"])."'";
$sql .= ",".quitar($HTTP_POST_VARS["email"])."'";
$sql .= ")";
mysql_query($sql);
echo "Registro exitoso!";

```

Nota: Por efectos practicos hemos echo esto muy sencillo, podrian agregar mas comprobaciones como la del email o manejo de errores mysql.

Tambien podria hacerse mas seguro mediante el uso de la funcion de encriptacion md5() para el password, aunque se esta haciendo lo mas sencillo posible ya que la finalidad de este articulo es la de enseñar como reconocer a los usuarios, de ustedes dependen estos "detalles" a la hora de "depurar" este codigo.

Bien, listo ya tenemos la aplicacion de registro, ¿pero y ahora que?, que hago con todos esos usuarios, ahora necesitamos que los usuarios ingresen y sean "reconocidos" y como lo dije lo haremos por medio de cookies.

Crearemos un formulario HTML para que los usuarios introduzcan su nick y password al igual que deberemos de realizar una aplicacion ingresar.php que sera la encargada de procesar si el usuario es valido y darle "ingreso" al sistema por medio de una cookie.

Es recomendable tener este archivo en la carpeta raiz de nuestra web, el formulario html seria algo asi:

```
<FORM ACTION="ingresar.php" METHOD="post">
Nick : <INPUT TYPE="text" NAME="nick" SIZE=20 MAXLENGTH=20>
<BR>
Password: <INPUT TYPE="password" NAME="password" SIZE=28 MAXLENGTH=20>
<BR>
<INPUT TYPE="submit" CLASS="boton" VALUE="Ingresar">
</FORM>
```

Y ahora el codigo de nuestra aplicacion ingresar.php, luego la explicacion:

```
<?php
```

```
//AQUI CONEXION O include() DE ARCHIVO DE CONEXION CON BASE DE DATOS.
```

```
function quitar($mensaje)
{
    $mensaje = str_replace("<","&lt;",$mensaje);
    $mensaje = str_replace(">","&gt;",$mensaje);
    $mensaje = str_replace("\'", "&#39;",$mensaje);
    $mensaje = str_replace("\"", "&quot;",$mensaje);
    $mensaje = str_replace("\\\\", "&#92;",$mensaje);
    return $mensaje;
}
```

```

if(trim($HTTP_POST_VARS["nick"]) != "" && trim($HTTP_POST_VARS["password"])
!= "")
{
$nickN = quitar($HTTP_POST_VARS["nick"]);
$passN = quitar($HTTP_POST_VARS["password"]);

$result = mysql_query("SELECT password FROM usuarios WHERE nick='$nickN'");
if($row = mysql_fetch_array($result))
{
if($row["password"] == $passN)
{
//90 dias dura la cookie
setcookie("usNick",$nickN,time()+7776000);
setcookie("usPass",$passN,time()+7776000);
?>
Ingreso exitoso, ahora sera dirigido a la pagina principal.
<SCRIPT LANGUAGE="javascript">
location.href = "index.php";
</SCRIPT>
<?
}
else
{
echo "Password incorrecto";
}
}
else
{
echo "Usuario no existente en la base de datos";
}
mysql_free_result($result);
}
else
{
echo "Debe especificar un nick y password";
}
mysql_close();
?>

```

Bien, arriba ya explique que debes de realizar la conexion y explique el funcionamiento de la funcion quitar(\$mensaje), asi que saltamos hasta el primer "if".

Lo primero que hacemos es comprobar que los campos de "nick" y "password" para que no esten vacios, ya sabemos para que sirve trim() asi que no lo repetire:

```

if(trim($HTTP_POST_VARS["nick"]) != "" && trim($HTTP_POST_VARS["password"])
!= "")

```

Asi que continuamos suponiendo que todo este bien, ahora lo que hacemos es quitarle el html, apostrofe comillas y demas al nick y password enviados por el usuario para su comprobacion con la base de datos, para ello utilizamos la funcion quitar(\$mensaje) que explicamos antes, y guardamos las nuevas cadenas en dos nuevas variables:

```
$nickN = quitar($HTTP_POST_VARS["nick"]);  
$passN = quitar($HTTP_POST_VARS["password"]);
```

Ahora ejecutamos una instruccion sql, en la cual le pedimos a la base de datos que de la tabla usuarios nos devuelva la fila cuyo nick corresponda por el enviado por el usuario:

```
$result = mysql_query("SELECT password FROM usuarios WHERE nick='$nickN'");
```

Si devuelve una columna quiere decir que el nick SI existe en la base de datos, de lo contrario devolvemos un mensaje de error:

```
if($row = mysql_fetch_array($result))  
{  
  ...  
}  
else  
{  
  echo "Usuario no existente en la base de datos";  
}
```

Ahora continuamos con el proceso suponiendo que el usuario exista en la base de datos ahora debemos comprobar que el password sea correcto, para ello hacemos una simple comparacion, si es correcto continuamos de lo contrario mostramos el mensaje de password incorrecto:

```
if($row["password"] == $passN)  
{  
  ...  
}  
else  
{  
  echo "Password incorrecto";  
}
```

Ahora suponiendo que el password es correcto continuamos y lo proximo a hacer es establecer las cookies de nick y password(Las cuales nombramos "usNick"/"usPass" y contienen el nick y password respectivamente) en la maquina del usuario, informamos que se ha realizado correctamente el login y lo enviamos a la pagina principal, una vez mas recalco que estamos haciendo esto lo mas sencillo posible, de ti depende personalizar y mejorar los mensajes :


```

setcookie("usNick",$nickN,time()+7776000);
setcookie("usPass",$passN,time()+7776000);
?>
<SCRIPT LANGUAGE="javascript">
alert("Ingreso exitoso");
location.href = "index.php";
</SCRIPT>
<?

```

Listo, el usuario ya tiene las cookies en su sistema, ¿pero ahora que?, como hacemos para comprobar en las paginas que queramos si el login es correcto, osea que el usuario posee las cookies y son validas?.

Para ello crearemos un ultimo archivo, lo llamaremos login.php, este se encargara de leer las cookies y decirnos si el usuario es un usuario registrado y que ha echo el respectivo "ingreso" al sistema, el contenido de dicho archivo seria el siguiente:

```

<?php
$loginCorrecto = false;
$idUsuarioL;
$nickUsuarioL;
$emailUsuarioL;
$nombreUsuarioL;

if(isset($_HTTP_COOKIE_VARS["usNick"]) &&
isset($_HTTP_COOKIE_VARS["usPass"]))
{
$result = mysql_query("SELECT * FROM usuarios WHERE
nick='".$_HTTP_COOKIE_VARS["usNick"]."' AND
password='".$_HTTP_COOKIE_VARS["usPass"]."'");

if($row = mysql_fetch_array($result))
{
setcookie("usNick",$_HTTP_COOKIE_VARS["usNick"],time()+7776000);
setcookie("usPass",$_HTTP_COOKIE_VARS["usPass"],time()+7776000);
$loginCorrecto = true;
$idUsuarioL = $row["id"];
$nickUsuarioL = $row["nick"];
$emailUsuarioL = $row["email"];
$nombreUsuarioL = $row["nombre"];
}
else
{
//Destruimos las cookies.
setcookie("usNick","x",time()-3600);
setcookie("usPass","x",time()-3600);
}
}

```

```
mysql_free_result($result);
}
?>
```

Lo primero que hacemos es declarar unas variables que utilizaremos para guardar los datos del usuario en caso de tener las cookies en su sistema y de estas ser correctas:

```
$loginCorrecto = false;
$idUsuarioL;
$nickUsuarioL;
$emailUsuarioL;
$nombreUsuarioL;
```

Hay una variable booleana que tenemos que mirar de cerca y es la de \$loginCorrecto ya que esta sera la que utilizaremos para comprobar el login en nuestros archivos, mas adelante veremos como.

Seguidamente lo que hacemos es comprobar que las cookies existan, si no existen pues no hay problema simplemente no hay "ingreso" :

```
if(isset($_HTTP_COOKIE_VARS["usNick"]) &&
isset($_HTTP_COOKIE_VARS["usPass"]))
```

Ahora lo que hacemos es una sentencia sql, en la cual le pedimos a la base de datos que nos devuelva la fila donde el nick y password correspondan al nick y password guardado en las cookies:

```
$result = mysql_query("SELECT * FROM usuarios WHERE
nick='".$_HTTP_COOKIE_VARS["usNick"]."' AND
password='".$_HTTP_COOKIE_VARS["usPass"]."'");
```

Si ese nick y password corresponden a una fila en la base de datos esto quiere decir que son correctos de lo contrario la informacion de las cookies esta mal y debemos destruirlas:

```
if($row = mysql_fetch_array($result))
{
...
}
else
{
//Destruimos las cookies.
setcookie("usNick","x",time()-3600);
setcookie("usPass","x",time()-3600);
}
```

Si la informacion es correcta procedemos con el ingreso, establecemos de nuevo las cookies con lo que se actualiza el tiempo de destruccion a otros noventa dias apartir de ahora:

```
setcookie("usNick",$HTTP_COOKIE_VARS["usNick"],time()+7776000);
setcookie("usPass",$HTTP_COOKIE_VARS["usPass"],time()+7776000);
```

Seguido de esto establecemos la variable \$loginCorrecto a "true", y las otras variables las llenamos con la informacion del usuario:

```
$loginCorrecto = true;
$idUsuarioL = $row["id"];
$nickUsuarioL = $row["nick"];
$emailUsuarioL = $row["email"];
$nombreUsuarioL = $row["nombre"];
```

Listo ya tenemos nuestro archivo "login.php", ahora ¿como lo utilizamos?, bien supongamos que queremos en nuestro "index.php" darle la bienvenida al usuario en caso de estar ingresado en el sistema deseamos saludarlo por su nick o nombre en caso contrario lo saludaremos como visitante, para ello tendríamos que incluir el archivo "login.php" al comienzo de nuestro archivo "index.php" asi:

Nota: Antes de incluir el "login.php" debe recordar realizar la conexion con la base de datos, de lo contrario obtendra errores.

```
<?php
//AQUI CONEXION O include() DE ARCHIVO DE CONEXION CON BASE DE
DATOS.
```

```
include("login.php");
```

...

Con eso el "login.php" se encargara de realizar las acciones que vimos arriba, ahora en nuestro "index.php" donde queramos saludar al visitante ponemos lo siguiente:

...

```
if($loginCorrecto)
{
echo "Estas identificado en el sistema como ".$nickUsuarioL;
}
else
{
echo "Bienvenido visitante, el sistema no te ha reconocido";
}

```

...

Como ven, hicimos uso de la variable \$loginCorrecto la cual sera "true" en caso de que el usuario fuera reconocido por el sistema, de lo contrario sera "false".

Y asi podemos usarlo en muchas areas, supongamos que tenemos una seccion donde solo pueden acceder los usuarios registrados, la llamaremos "restringida.php", entonces lo que haríamos para prevenir que usuarios no registrados vean el contenido seria algo asi:

```
<?php
//AQUI CONEXION O include() DE ARCHIVO DE CONEXION CON BASE DE
DATOS.

include("login.php");

if($loginCorrecto)
{
echo "Aqui el contenido solo para usuarios registrados";
}
else
{
echo "El sistema no lo ha identificado, solo los usuarios registrados tienen acceso a esta
area";
}
?>
```

Eso es todo, espero haya quedado claro. PD: Sobre el proceso de "logout" o "salir" no lo explico ya que es algo muy sencillo, simplemente se "borran" las cookies.