Autenticación de Usuarios y Control de Acceso.

Para poder identificar al usuario, resulta necesario depender de un mecanismo de reconocimiento. Este reconocimiento puede utilizar muchas alternativas. El sistema más común y generalizado es mediante el uso de un nombre y una contraseña. Ambos campos de información, son almacenados en una base de datos protegida y segura.

Para dar mayor margen operativo de seguridad, dado que las contraseñas pueden ser objeto de vulnerabilidades operativas, la información de las contraseñas se suelen encriptar “codificar” mediante un código de cifrado específico. Se puede utilizar MD5, SHA y sus versiones. El lenguaje PHP y el gestor MySQL utilizan modelo de cifrado llamado PASSWORD. Este sistema de cifrado es básico pero bastante eficiente. Para garantizar un mayor modelo de seguridad, se recomienda utilizar MD5 o SHA y algunas de sus versiones. Tanto MD5 como SHA y sus versiones, no pueden ser descifradas. Esto permite garantizar seguridad.

Los valores de información se almacenan en una base de datos protegidas, las cuales, tienen un control más refinado y seguro que los que proporcionan los archivos **.htaccess** del servidor Apache. Además de ello, el gestor de base de datos permite determinar quién puede controlar la tabla o base de datos y otorgar a su vez privilegios totalitarios, parcializados o restringidos. Tal es el caso de un control totalitario por parte de un administrador, de un control parcializado para el control de un administrador operativo del recurso y privilegios diferenciados o restringidos para los usuarios del recurso.

Puede haber sitios web en los que los usuarios autentificados pueden utilizar sólo una parte de la información (como los bancos, que permiten a sus clientes acceder únicamente a la información relativa a sus cuentas). Otros sitios web necesitan separar en grupos a los usuarios autentificados, de

tal forma que la información a la que accede un usuario depende del grupo en que éste se encuentre. Por ejemplo, una aplicación de gestión de una empresa puede tener un grupo de usuarios a los que permite visualizar la información, y otro grupo de usuarios que, además de visualizar la información, también la pueden modificar.

***Mecanismos de autentificación.***

Desde PHP puedes acceder a la información de autentificación HTTP que ha introducido el usuario utilizando el array **superglobal $\_SERVER**.

**Valor Contenido**

**Valor Contenido**

**$\_SERVER['PHP\_AUTH\_USER']** Nombre de usuario que se ha introducido.

**$\_SERVER['PHP\_AUTH\_PW']** Contraseña introducida.

**$\_SERVER['AUTH\_TYPE']** Se refiere a mecanismos de autenticación. Puede ser Basic o Digest.

Basic.- Es un mecanismo de autenticación rápido, fácil de configurar, pero inseguro.

Digest.- Es considerado un poco más seguro que el Basic, el mecanismo

de autenticación consta de 4 pasos por lo que es considerablemente más lento que el Basic.

Con la función **header()** se fuerza a que el servidor envíe un error de "Acceso no autorizado" (código 401).

Ejercicio:

<?php

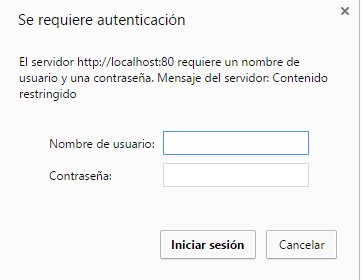
header('WWW-Authenticate: Basic Realm="Contenido restringido"');

header('HTTP/1.0 401 Unauthorized');

echo "Usuario no identificado";

exit;

?>



La función header() envía encabezados HTTP (bloque de datos que forma parte del protocolo HTTP y se envía antes de la información propia que se transmite. Permite especificar códigos de estado, acciones requeridas al servidor, o el tipo de información que se transmite), pero debe utilizarse antes de que se muestre nada por pantalla. En caso contrario, obtendrás un error.

Con el código anterior, la página envía un error 401, lo que fuerza al navegador a solicitar las

credenciales de acceso (nombre de usuario y contraseña).

Ejercicio:

Realiza una página que solicite la autenticación al usuario y visualice según el caso:

* Usuario no reconocido
* Has pulsado INICIAR SESION y visualice el usuario y la contraseña.

Nota: con la función: unset($\_SERVER) eliminamos $\_SERVER.

<?php

//unset($\_SERVER);

header('Content-Type: text/html; charset=UTF-8');

if (!isset($\_SERVER['PHP\_AUTH\_USER'])) {

header('WWW-Authenticate: Basic Realm="Contenido restringido"');

header('HTTP/1.0 401 Unauthorized');

echo "Usuario no reconocido!";

exit;

}

else {

echo "<p>Has pulsado INICIAR SESION</p>";

echo "<p>Usuario: {$\_SERVER['PHP\_AUTH\_USER']}</p>";

echo "<p>Contraseña: {$\_SERVER['PHP\_AUTH\_PW']} </p>";

}

?>

1.-Ejercicio:

Modificar la página anterior para que visualice:

* Has pulsado CANCELAR
* En el caso de que el usuario sea: dwes y contraseña: abc123. debe aparecer ENTRANDO…

2.- Ejercicio:

Crea una base de datos en MySql llamada dwesTema5

Crea la tabla usuarios con usuario(PK), contrasenya varchar(32)

INSERT INTO usuarios VALUES('dwes',md5('abc123.'));

Aunque se podrían almacenar las contraseñas en texto plano, es mejor hacerlo en formato

encriptado. En el ejemplo anterior, para el usuario "dwes" se almacena el hash MD5 (método para generar un resumen de un texto o un documento, de tal forma que a partir del resumen obtenido no es posible recuperar el texto original, ni hallar otro texto a partir del cual se obtenga el mismo resumen. Se llama hash al resumen obtenido al aplicar una función hash. Una de las funciones hash más extendidas es MD5, que genera 128 bits como resumen (normalmente se representa mediante una cadena de texto de 28 caracteres o mediante 32 dígitos hexadecimales).En PHP puedes usar la función md5 para calcular el hash MD5 de una cadena de texto.

[**http://es.php.net/manual/es/function.md5.php**](http://es.php.net/manual/es/function.md5.php)

3.-Ejercicio

Teniendo en cuenta el ejercicio anterior, realizar una página que nos pida usuario y contraseña. Si introduces un usuario o contraseña incorrectos, el comportamiento depende del navegador que utilices, algunos te pedirán las credenciales de forma indefinida, y otros un número limitado de veces.