

PHP

Prof. Dely Maybel Gil A.



Contenido



1

Scripts de AUTENTICACIÓN

2

Encriptación de Datos

3

Cookies

4

Sesiones

- ❖ Asegurar contraseña usando encriptación y/o conversión (hashing)
- ❖ La contraseña debe tener de 8 a 10 caracteres y mezcla de letras y dígitos.
 - Ficheros Planos
 - Base de Datos

FICHEROS PLANOS

- ❖ La manera más primitiva y libre de complicaciones para almacenar la contraseña.
- ❖ Son susceptibles frente a intrusos y otros accesos maliciosos (Formato no codificado).
- ❖ Se usa si el número de usuarios es bajo de lo contrario puede llegar a ser inmanejable su información.



BASE DE DATOS

- ❖ Es un método conveniente, razonable y seguro para almacenar contraseñas.

MECANISMO DE CODIFICACIÓN DE CONTRASEÑAS EXTERNAS.

- ❖ Es la seguridad en la transmisión entre la máquina del Usuario y el Servidor.
 - Kerberos
 - Infraestructura de Clave Pública (Public Key Infrastructure PKI)

KERBEROS

- ❖ Mecanismo de Autenticación basado en Claves utilizado para asignar una clave única llamado ticket emitida por Kerberos Authentication Server (AS) a cada Usuario que se registra al Sitio Web. Esta clave se utiliza a lo largo de la Sesión.

PKI

- ❖ Usa Certificados digitales emitidos por un tercero llamado Autoridad Certificadora (Certificate Authority, C.A.). Estos certificados digitales se utilizan para establecer la identidad del remitente y la autenticidad de la Solicitud.
- ❖ Estos mecanismos se incluyen en las aplicaciones de Software o se ofrecen como un Servicio o un Producto.



- ❖ Es la ciencia que se encarga de proteger los datos de acceso desautorizados y maliciosos
 - Se basa en dos componentes:
 - Clave y algoritmo de encriptación (hash)

CODIFICACIÓN MD5 (Message Digest 5)

- ❖ Es un cálculo de conversión que produce un extrato de datos únicos de 128 bits
 - Asegura que la sección de datos que produzca sea extremadamente difícil de reproducir matemáticamente



Crear Tabla login



- ❖ <?php
- ❖ **1.- Crear Tabla login**
- ❖ \$con =mysql_connect("localhost","root")
- ❖ or die("¡No se ha podido establecer la conexión con el servidor!");
- ❖ \$resp = mysql_select_db("curso")
- ❖ or die ("¡No se ha podido seleccionar la Base de datos!".mysql_errno().": ".mysql_error()."
");
- ❖ \$sqlquery = "create table if not exists login (
usuario varchar(30) PRIMARY KEY,
contrasena varchar(100) NOT NULL default "
Email varchar(15) NOT NULL default ")";
- ❖ \$queryresult = mysql_query(\$sqlquery)
- ❖ or die ("¡No se ha podido ejecutar la consulta!".mysql_errno().": ".mysql_error()."
");
- ❖ echo "La Tabla Login se ha creado satisfactoriamente en la Base de Datos.";
- ❖ ?>



Formulario de Inscripción



2.- Cree el siguiente formulario: (add_user.php).

Colocar en el action del form: add.php

Formulación de Inscripción de Usuarios

Usuario	<input type="text"/>
Contraseña	<input type="text"/>
Confirmar Contraseña	<input type="text"/>
email:	<input type="text"/>
<input type="button" value="Enviar"/>	

add.php

3.- Cree el siguiente script

```
<?php
include 'conexion.php';
$link=Conectarse();
$user = $_GET['txtUser'];
$pass = $_GET['txtPassword'];
$email = $_GET['txtEmail'];
$query = "INSERT INTO login VALUES ('$user ','.md5($pass).',
'email)";
echo $query;
mysql_query($query,$link)
    or die ("¡No se ha podido insertar!".mysql_errno().":
.mysql_error()."<BR>");
if (!mysql_affected_rows($link))
    echo "Error: ";
else{
    echo "Usuario: " . $user ;
    echo "<br> Password: " . $pass ;
    echo "<br> Email: " . $email."<br>";
}
mysql_close();
?>
```

Formulario del Ingreso

4.- Cree el siguiente formulario: (login.php).

Colocar en el action del form: login_rs.php

Ingresa tu usuario y contraseña

Usuario	<input type="text"/>
Contraseña	<input type="password"/>
<input type="button" value="Entrar"/>	

txtUser

txtPassword

```
<?php
❖ include 'conexion.php';
❖ $link=Conectarse();
❖ $user = $_GET['txtUser'];
❖ $pass = $_GET['txtPassword'];
❖ $query = "SELECT contraseña FROM login WHERE usuario='".$user.'";
❖ $rs= mysql_query($query , $link)
❖     or die ("¡No se ha podido insertar!".mysql_errno().": ".mysql_error());
❖ if (!mysql_affected_rows($link))
❖     die("No existe un Usuario con ese login y password");
❖ else
❖     if ($row=mysql_fetch_array($rs))
❖         if (!(md5($pass) == $row['contrasena']))
❖             die("Contraseña inválida");
❖         else
❖             die("Usuario autorizado");
❖         echo "<br> Email:  ".$email."<br>";
❖ mysql_close();
❖ ?>
```

- ❖ Son cadenas de texto que son enviadas desde el servidor al cliente (navegador) y almacenadas en este, luego el navegador envía estas cookies al servidor permitiendo así la identificación del cliente en el servidor.

- ❖ Pequeños archivos de texto
- ❖ Almacenados en el cliente
- ❖ Información persistente
- ❖ Poco seguros

- ❖ Setcookie() define una cookie que es enviada junto con el resto de la información de la cabecera(header).
- ❖ Las cookies deben ser enviadas antes de cualquier tag de html, por lo tanto deberemos realizar la llamada a estas funciones antes de cualquier tag <HTML> o <HEAD>. Esta es una restricción de las cookies no de PHP.

- ❖ `setcookie(<nombre>, [<valor>, <caducidad>, <ruta>, <dominio>, secure])` : Crear cookie
- ❖ `$HTTP_COOKIE_VARS("<nombre_variable>")`:
Lee contenido de una cookie
- ❖ `setcookie(<nombre>)`:
Borra una cookie, sólo se coloca nombre no se le asocia valor
- ❖ Array de cookies: Cookie de múltiples valores
`setcookie("datos[0]", "Jonas")`
`setcookie("datos[1]", "Job")`

La función para crear cookies debe ser lo primero en ejecutarse una línea vacía o un espacio en blanco es suficiente para generar un error



Cookies



```
int setcookie (string Nombre [, string Valor [, int Expire [,  
string Path [, string Dominio [, int Secure]]]])
```

- ❖ **Nombre.** Nombre de la cookie. Si creamos una cookie solamente con el nombre, en el cliente se eliminará la cookie que exista con ese nombre. También podemos reemplazar cualquier argumento con una cadena vacía ("").
- ❖ **Value.** Valor que almacenará la cookie en el cliente.
- ❖ **Expire.** El argumento expire es un argumento entero que indica la hora en que se eliminara la cookie en el formato de hora que devuelven las funciones UNIX time() y mktime(). Normalmente se usa time() + N. segundos de duración, para especificar la duración de una cookie.
- ❖ **Path.** Subdirectorío en donde tiene valor la cookie.
- ❖ **Dominio.** Dominio en donde tiene valor la cookie. Si ponemos como dominio www.domain.com la cookie no se transmite para domain.com, mientras que si ponemos domain.com la cookie se transmite tanto para domain.com como para www.domain.com
- ❖ **Secure.** El argumento secure indica que la cookie sólo se transmitirá a través de una conexión segura HTTPS.

- ❖ `header(nombre_cabecera:valor_cabecera)` : Permite enviar órdenes o acciones a través de la cabecera del protocolo HTTP
 - `header("Location:index.php ")`
 - `header ("Cache-Control:no-cache, must-revalite ")`: La página no sea cacheada → Obliga al Servidor a proporcionar la última versión de dicha página
 - `header("Content-Type: <tipo>text/html ")`: Establece el tipo de documento que será enviado al navegador del cliente.

Ejemplo:

```
header("Content-Type:text/html ");  
header(" Content-Type:text/vnd.wap.wml ");
```



- ❖ Es el tiempo en el que un usuario se mantiene conectado en un sitio Web.
 - ❖ Memoria del servidor
 - ❖ Más seguro
 - ❖ Almacenamiento volátil

ALMACENAMIENTO DE LAS SESIONES

Es una cadena denominada **SID**: ID de Sesión

SEGUIMIENTO:

- ❖ Se enviará a través de una cookie llamada PHPSESSID
- ❖ Se guarda en el Servidor como un archivo temporal

ALMACENA

- ❖ Enteros.
- ❖ Cadenas de Carácter.
- ❖ Arrays
- ❖ Objetos

```
Habilitar (php.ini)  
session.use_cookies= 1  
session.use_trans_sid= 1.
```



Funciones Sesiones



<code>session_start()</code>	Crea un sesión sino existía antes y si la hubiese, la retomaría gracias al ID de sesión
<code>session_unset()</code>	Elimina todas las variables de la sesión.
<code>session_destroy()</code>	Elimina la sesión y del archivo temporal donde la almacena, siempre se debe invocar primero a <code>session_unset()</code> .
<code>session_register()</code>	Registra una variable en una sesión. Ejemplo: <code>session_register("var")</code> .
<code>session_unregister()</code>	Elimina una variable de sesión <code>session_unregister("var")</code>
<code>\$HTTP_SESSION_VARS</code>	Devuelve una variable de una sesión. Ejemplo: <code>\$HTTP_SESSION_VARS["var"] = valor</code> .
<code>session_is_registered</code>	Devuelve true si una variable existe en la sesión. Ejemplo: <code>session_is_registered("var")</code> .



Funciones Sesiones



session_id()	Devuelve el identificador de la sesión que ha creado el usuario y que le identifica en forma única
session_name()	Devuelve el nombre de la sesión utilizada en ese momento. Por defecto es PHPSESSID (php.ini).
session_encode()	Codificar la información de la sesión actual en una cadena de caracteres.
session_decode()	Descodifica a su estado original la información de la sesión, codificada en su momento por session_encode
session_module_name	Devuelve le valor de la variable session.save_handler situada en el fichero php.ini encargada de definir el tipo de handler para grabar y recuperar el manejo de las sesiones (Defecto:files)
Session_save_path	Permite conocer la ruta en que serán guardadas las sesiones



login_rs.php



```
❖ <?php
❖ .....
❖ if (!mysql_affected_rows($link))
❖     die("No existe un Usuario con ese login y password");
❖ else
❖     if ($row=mysql_fetch_array($rs))
❖         if (!(md5($pass) == $row['contrasena']))
❖             die("Contraseña inválida");
❖ mysql_close();
❖ //Implementación de la codificación de la sesión
❖ session_name();
❖ session_start();
❖ session_register("user");
❖ session_encode();
❖ $url = "Location: mypag.php?PHPSESSID=".session_id();
❖ header($url);
❖ ?>
```



mypag.php / mypag2.php



- ❖ <?php
- ❖ session_name();
- ❖ session_start("user");
- ❖ if (!(session_is_registered("user"))){
- ❖ session_unset();
- ❖ session_destroy();
- ❖ die ("Por favor, !inicie
primero!
");
- ❖ }
- ❖ ?>

Bienvenido a su área personal:::: maygil
!!!MI PÁGINA 1!!
<u>Página 2</u>
<u>Cerrar la sesión</u>

```
Bienvenido a su &acute;rea personal:::: <?php echo $user; ?>
<?php
echo "<a href=\"mypag2.php?PHPSESSID=\".$PHPSESSID.\"\"> Página 2 </a>";
?>
<a href="logout.php">Cerrar la sesi&oacute;n </a>
```



logout.php



```
❖ <?php
❖ session_name();
❖ session_start("user");
❖ if (session_is_registered("user")){
❖     session_unset();
❖     session_destroy();
❖     $url = "Location:login.php";
❖     header($url);
❖ }else{
❖     die ("Sólo los usuarios que iniciaron una sesión pueden salir fuera
del sistema");
❖ }
❖ ?>
```