

Unidad 8

Seguridad

BIBLIOTECAS DE CLASES

Un recurso importante que podemos encontrar en la web son las bibliotecas o librerías de clases. La idea en torno a esto es poder aprovechar al máximo la característica de modularidad de las clases y la POO en general para desarrollar clases que puedan ser utilizadas por la comunidad de programadores.

La mayoría de los lenguajes de programación orientado a objetos poseen repositorios de este tipo en donde podemos nutrirnos de clases con distintas funcionalidades.

El sitio <https://www.phpclasses.org/> es una gran biblioteca de clases para PHP aportadas por usuarios y ordenadas por categorías, este portal de uso gratuito en inglés también dispone de un buscador de clases siendo necesario registrarse para la descarga de archivos.



A modo de ejemplo veremos 2 de estas clases que pueden sernos de muchísima utilidad, ellas son:

- Pagination
- Adminpro

Generalmente lo que bajamos al querer usar cualquiera de estas clases es un grupo de archivos: uno es el php con la clase propiamente dicha, algún archivo php de ejemplo que hace uso de la clase y algún archivo (que puede ser un .doc, .txt, .html, etc) que contiene una breve explicación de cómo se utiliza la clase.

Vamos a ver entonces estos dos ejemplos. La idea es que puedan utilizar este repositorio a medida que se les vayan presentando situaciones a resolver. Y que si desarrollan una clase que pueda llegar a ser de utilidad al resto de la comunidad la puedan poner a disposición del resto de los programadores subiéndola a este sitio.

PAGINATION

Esta clase nos sirve para hacer el paginado de resultados de una consulta SQL.

Para el ejemplo les pido que importen la tabla de Alumnos que se encuentra en el archivo alumnos.sql

El archivo testpage.php es un ejemplo del llamado a la clase pagination_class.php.

Vamos a ver como:

La clase espera recibir 3 parámetros:

- El query,
- el registro desde el cual se va a mostrar y
- la cantidad de registros que se muestran por página.

Esos son los parámetros que se setean antes de la instanciación de la clase. Y lo único de lo que tenemos que encargarnos nosotros, además del diseño con el cual se mostrarán los datos.

Es la clase quien resuelve el paginado de los registros encontrados según la variable \$recpage que contiene la cantidad de registros a mostrarse por página. Y la encargada de armar las siguientes líneas con los vínculos correspondientes habilitados.

Primera | Anterior | 1 | 2 | Siguiente | Ultima
Página: 1 De 2. Total Registros Encontrados: 20

que se muestran al pie de los registros.

El código es el siguiente:

```
<?php
include('pagination_class.php');
//mysql_connect('localhost', 'user', 'pass') or die(mysql_error());
mysql_connect('localhost', 'root', '') or die(mysql_error());
//mysql_select_db('std_db');
mysql_select_db('alumnos');
?>
<script language="JavaScript">
function pagination(page)
{
```

```
window.location =
"testpage.php?search_text="+document.form1.search_text.value+"&starting="+page;
}
</script>
<?php
//qry a modificar segun tabla que queremos consultar
$qry = "SELECT * FROM alumnos";
if(isset($_REQUEST['search_text'])){
$searchText = $_REQUEST['search_text'];
$qry .= " where apellido like '$searchText%'";
}
else{
$searchText = "";
}
//for pagination
if(isset($_GET['starting'])&& !isset($_REQUEST['submit'])){
$starting=$_GET['starting'];
}else{
$starting=0;
}
$recpage = 10;//number of records per page
$obj = new Pagination_class($qry,$starting,$recpage);
$result = $obj->result;
?>
<form name="form1" action="testpage.php" method="POST">
<table border="1" width="40%">
<tr>
<td colspan="3">
Search <input type="text" name="search_text" value="<?php echo
$searchText;?>">
<input type="submit" value="Search">
</td>
</tr>
<tr><td>Nro.</td><td>Nombre</td><td>Apellido</td></tr>
<?php if(mysql_num_rows($result)!=0){
$count = $starting + 1;
while($data = mysql_fetch_array($result)) {?>
<tr>
<td><?php echo $count; ?></td>
<td><?php echo $data['nombre']; ?></td>
<td><?php echo $data['apellido']; ?></td>
```

```
</tr><?php $counter ++;  
} ?>  
<tr><TD align="center" colspan="3"><?php echo $obj->anchors;  
?></TD></tr>  
<tr><TD align="center" colspan="3"><?php echo $obj->total;  
?></TD></tr>  
<?php }else{  
echo "No Data Found";  
}?>  
</TD></tr></table></form>
```

ADMINPRO

Adminpro es una clase para autenticación de usuarios basada en una base de datos MySQL y funciones PHP para el manejo de sesiones. Es útil para sitios web que necesiten área protegida por contraseña.

Este paquete incluye un script para administración de usuarios y un script para inicializar las tablas MySQL.

La clase puede:

- Hacer login y logout de un usuario vía formularios web.
- Expirar la sesión de un usuario autenticado si para la sesión ha transcurrido un tiempo de espera y requiere un nuevo inicio de sesión
- Puede mostrar mensajes de error y enviar al usuario a la página de login cuando el usuario no es autenticado o la sesión expira.
- Verifica cuando un usuario logueado tiene privilegios de administrador o de grupo.
- Permite a los usuarios usar la función “Recordarme” basado en cookies.

Uso de la clase:

1 - El siguiente código se debe usar en cada página que quieras proteger:

Página protegida normal (sin protección de administrador o de grupo de usuario)

```
<?php include("adminpro_class.php");  
$prot=new protect();  
if ($prot->showPage) {  
?>  
<!-- HERE COMES YOUR PHP CODE OR YOUR HTML CODE ---!>  
<?php } ?>
```

Página protegida de administrdor (sin protección de grupo de usuarios): Los administradores tendrán siempre acceso incluso si pertenecen a un grupo de usuarios.

```
<?php include("adminpro_class.php");  
$prot=new protect("1");  
if ($prot->showPage) {  
?>  
<!-- HERE COMES YOUR PHP CODE OR YOUR HTML CODE ---!>
```



```
<?php } ?>
```

Página protegida de grupo de usuarios (sin protección de administrador). En este ejemplo solo los usuarios del grupo 2 tendrán acceso.

```
<?php include("adminpro_class.php");  
$prot=new protect("0","2");  
if ($prot->showPage) {  
?>  
<!-- HERE COMES YOUR PHP CODE OR YOUR HTML CODE ---!>  
<?php } ?>
```

Obtener el usuario actual:

```
<?php include("adminpro_class.php");  
$prot=new protect();  
if ($prot->showPage) {  
$curUser=$prot->getUser();  
?>  
<!-- HERE COMES YOUR PHP CODE OR YOUR HTML CODE ---!>  
<?php } ?>
```

2 - El login debe ser enviado con un formulario usando el método POST y la estructura debe ser la siguiente (prestar atención al campo oculto en el formulario):

Login normal:

```
<form action="first_protected_page_after_login.php"  
method="POST">  
<input type="hidden" name="action" value="login">  
<input type="text" name="userName">  
<input type="password" name="userPass">  
<input type="submit">  
</form>
```

Login con la function "Recordarme": La variable `$globalConfig['enblRemember']` (en `adminpro_config.php`) debe ser seteada a `true` para habilitar esta función.

```
<form action="first_protected_page_after_login.php"  
method="POST">  
<input type="hidden" name="action" value="login">  
<input type="text" name="userName">
```

```
<input type="password" name="userPass">
<input type="radio" name="userRemember" value="no">No
<input type="radio" name="userRemember" value="yes">Yes
<input type="submit">
</form>
```

El Logout debe ser enviado también por POST y debe tener la siguiente estructura:

```
<form action=" " method="POST">
<input type="hidden" name="action" value="logout">
```

En la carpeta adminpro que se encuentra en los ejemplos anexados a esta unidad tienen todos los archivos descargados junto con esta clase.

En la carpeta inmobiliaria-adminpro tienen un ejemplo de cómo anexar esta clase a un sistema al cual le quieren agregar páginas protegidas por contraseña.

Básicamente lo que queremos hacer es proteger por contraseña las secciones propiedades.php y buscar_prop.php.

Para ello lo que tenemos que hacer es:

- colocar el código correspondiente en los mencionados scripts según las instrucciones.
- Hacer los archivos login.html y logout.html como se indica anteriormente
- Incluir los archivos: adminpro_class.php, adminpro_config.php y MySQL_dialog.php
- Agregar la tabla myuser provista con la clase adminpro a nuestra base de daos Inmobiliaria.

Otras dos clases que podemos descargar desde phpclasses.org o desde el sitio oficial que estas clases poseen son:

- PHPMailer y
- FPDF

FPDF



<http://www.fpdf.org/>

¿QUÉ ES FPDF?

FPDF es una clase escrita en PHP que permite generar documentos PDF directamente desde PHP, es decir, sin usar la biblioteca PDFlib. La F de FPDF significa Free (gratis y libre): puede usted usarla para cualquier propósito y modificarla a su gusto para satisfacer sus necesidades.

FPDF tiene otras ventajas: funciones de alto nivel. Esta es una lista de sus principales características:

- Elección de la unidad de medida, formato de página y márgenes
- Gestión de cabeceras y pies de página
- Salto de página automático
- Salto de línea y justificación del texto automáticos
- Admisión de imágenes (JPEG, PNG y GIF)
- Colores
- Enlaces
- Admisión de fuentes TrueType, Type1 y codificación
- Compresión de página

FPDF no necesita de ninguna extensión para PHP (excepto zlib para activar la compresión y GD para soporte a GIF) y funciona con PHP4 ($\geq 4.3.10$) y PHP5.

Los tutoriales le permitirán empezar rápidamente a usar FPDF. La documentación completa (on line) se encuentra aquí y el área de descarga, aquí. Se recomienda encarecidamente que lea las FAQ (o PMF: preguntas más frecuentes), que listan las preguntas y problemas más comunes (especialmente los relacionados con los navegadores).

Existe una sección de scripts que incluye algunas extensiones útiles.

¿QUÉ IDIOMAS PUEDO USAR?

La clase genera documentos en muchos otros idiomas aparte de los europeos occidentales: europeo central, cirílico, griego, báltico y tailandés, si se dispone de fuentes TrueType o Type1 con el conjunto de caracteres adecuado. El chino y el japonés también están admitidos.

UTF-8 también está disponible.

¿Y QUÉ HAY DE LA EFICIENCIA?

Como es lógico, la velocidad de generación de un documento es menor que con PDFlib. Sin embargo, la desventaja en cuanto a rendimiento es muy razonable y se adecúa a la mayoría de los casos, a no ser que sus documentos vayan a ser especialmente complejos o extensos.

La carpeta fpdf16 que encuentren entre los ejemplos de esta unidad contiene todos los archivos que se pueden bajar de la página de FPDF. En la carpeta fpdf16/tutorial se encuentran todos los ejemplos para poder aprender a utilizar la clase con facilidad. Y en la carpeta fpdf16/doc está toda la documentación de los métodos de la clase.

PHPMAILER



<http://phpmailer.worxware.com/>

¿Qué es phpmailer?

PHPMailer es una clase php para enviar emails basada en el componente active server ASPMail. Permite de una forma sencilla tareas complejas como, por ejemplo:

- Enviar mensajes de correo con ficheros adjuntos (attachments)
- enviar mensajes de correo en formato HTML

Con PHPMailer se pueden enviar emails via sendmail, PHP mail(), o con SMTP. Lo más recomendable es usando smtp por dos razones:

- Con phpmailer se pueden usar varios servidores SMTP. Esto permite repartir la carga entre varias computadoras, con lo que se podrán enviar un mayor número de mensajes en un tiempo menor.
- Además, los servidores SMTP permiten mensajes con múltiples to's (destinatarios), cc's (Las direcciones que aparezcan en este campo recibirán el mensaje. Todos los destinatarios verán las direcciones que aparecen en la sección Cc), bcc's (Las direcciones que aparezcan en el campo Bcc recibirán el mensaje. Sin embargo, ninguno de los destinatarios podrá ver las direcciones que fueron incluidas en la sección Bcc) y Replytos (direcciones de respuesta)

¿Por qué usar phpmailer?

Es posible enviar email con la función mail() de php, pero dicha función no permite algunas de las más populares características que proporcionan los clientes de correo usados actualmente. Entre estas características se encuentran el envío de email con ficheros adjuntos.

PHPMailer hace fácil esta difícil tarea de enviar correos con estas características y puedes incluso utilizar tu propio servidor smtp aunque éste requiera autenticación (un nombre de usuario y contraseña), con lo que se podrá usar una cuenta gratuita de correo.

Contacto: consultas@elearning-total.com

Web: www.elearning-total.com

- Necesitamos el fichero class.phpmailer.php que es la clase PHPMailer propiamente dicha y también el fichero class.smtp.php que nos permite el envío de emails a través de un servidor smtp.
- El siguiente trozo de código es necesario para instanciar desde nuestra página php a la clase PHPMailer. En la primera línea incluimos el fichero class.phpmailer.php y en la segunda creamos un objeto de la clase PHPMailer:

```
require("class.phpmailer.php");
$mail = new PHPMailer();
```

- Una vez instanciado el objeto de la clase PHPMailer es necesario configurar sus propiedades y llamar a sus métodos. A continuación, enumeramos propiedades y métodos principales de la clase PHPMailer.

Nombre	Descripción	Valor por defecto
AltBody	Establece el cuerpo del mensaje como sólo de texto	""
Body	Establece el cuerpo del mensaje. Puede ser texto simple o con formato HTML. Si es con formato HTML hay que ejecutar el método IsHTML(True)	""
ConfirmReadingTo	Establece la dirección de correo a la que se enviará una Confirmación de que el correo ha sido leído	""
ErrorInfo	Informa del error más reciente producido al intentar enviar un email	""
From	Establece la dirección de correo de origen del Mensaje	root@localhost

FromName	Establece el nombre de quien envía el mensaje	Root User
Host	Establece el servidor SMTP. Pueden ser varios separados por ;	Localhost
Mailer	Establece el método para enviar el mensaje puede ser mail, senmail o smtp. Se recomienda usar smtp siempre que sea posible	Mail
Password	Establece la contraseña del servidor SMTP	""
PluginDir	Establece el directorio donde están incluidos los plugins de phpmailer. Muy útil para indicar el directorio en el que se encuentra la clase SMTP	""
Port	Establece el puerto por defecto del servidor SMTP	25
ReplyTo	Establece todas las direcciones ReplyTo	Array()
SMTPAuth	Establece la autenticación SMTP	False
Subject	Establece el asunto del mensaje	""
Timeout	Establece el timeout del servidor STMP en segundos	10
Username	Establece el nombre de usuario SMTP	""
WordWrap	Establece el word Wrapping del cuerpo de un mensaje a un número determinado de caracteres	0

Métodos

Nombre	Sintaxis	Descripción
AddAddress	<code>void AddAddress (\$address, \$name)</code>	Añade una dirección de destino del mensaje. El parámetro \$name es opcional
AddAttachment	<code>bool AddAttachment (\$path, \$name, [\$encoding = "base64"], [\$type = "application/octetstream"])</code>	Añade un fichero adjunto al mensaje. Retorna false si el fichero no pudo ser encontrado. \$path, es la ruta del archivo puede ser relativa al script php (no a la clase PHPMailer) o una ruta absoluta. Se aconseja usar rutas relativas \$name, nombre del fichero \$encoding, tipo de codificación. Se aconseja dejar la predeterminada \$type, el valor por defecto funciona con cualquier clase de archivo. Se puede usar un tipo específico como por ejemplo image/jpeg
AddBCC	<code>void AddBCC (\$address, \$name)</code>	Añade una dirección BCC.
AddCC	<code>void AddCC (\$address, \$name)</code>	Añade una dirección CC

AddReplyTo	void AddReplyTo (\$address, \$name)	Añade una dirección ReplyTo
ClearAddresses	void ClearAddresses ()	Borra todas las direcciones de destino establecidas anteriormente
ClearAttachments	void ClearAttachments ()	Borra todos los ficheros adjuntos establecidos anteriormente
IsHTML	void IsHTML (\$bool)	Establece el tipo de mensaje a HTML
Send	bool Send ()	Envía el mensaje, devuelve false si ha habido algún problema. Consultando la propiedad ErrorInfo se puede saber cuál ha sido el problema

Ejemplo sencillo

Esta primera aplicación de ejemplo va a estar formada por:

- Un archivo ejemplo1.html que contendría el formulario a completar por el usuario
- Un archivo llamado ejemplo1.php que recogerá los datos del formulario y utilizará la clase PHPMailer para hacer el envío de mail
- Un subdirectorío llamado includes donde estarán los ficheros class.phpmailer.php y class.smtp.php

Código de ejemplo.php:

```
<?php
// primero hay que incluir la clase phpmailer para poder instanciar
```

Contacto: consultas@elearning-total.com

Web: www.elearning-total.com

```
//un objeto de la misma
require ("includes/class.phpmailer.php");
require ("includes/class.smtp.php");
$mail->Host = "smtp.live.com"; // sets HOTMAIL as the SMTP server
$mail->Port = 25; // alternate is "26" - set the SMTP port for the HOTMAIL server
$mail->Username = "usuario@hotmail.com"; // HOTMAIL username
$mail->Password = "password"; // HOTMAIL password
*/
/*
//GMAIL
$mail->SMTPSecure = "tls";
$mail->Host = "smtp.gmail.com";
$mail->Port = 587; // si setemos SMTPSecure = "ssl"; tenemos que setear Port =
465;
$mail->Username = "usuario@gmail.com"; // GMAIL username
$mail->Password = "password"; // GMAIL password
*/
//Cualquier otro servidor
$mail->Host = "mail.midominio.com.ar";
$mail->Username="info@midominio.com.ar";
$mail->Password = "mipass";
$mail->IsHTML(true);
$mail->From = $_POST['email'];
$mail->FromName = $_POST['email'];
$mail->AddAddress("veropineyro@hotmail.com");
$mail->Subject = "Contacto a traves de su sitio web";
$mail->Body = $_POST['nombre']." ha enviado un mail por el siguiente motivo:
<br />".$_POST['asunto']."<br /> Con los siguientes comentarios: <br
/>".$_POST['comentarios']."<br
/>Su telefono es: ".$_POST['telefono'];
$exito=$mail->send();
if ($exito){
print("<br /><h2>Gracias por contactarnos. </h2>");
}
else{
print("<br /><h2>Error al enviar el mensaje. Por favor intentelo
nuevamente. </h2>");
}
?>
```

Comentarios al ejemplo

Dos cosas a destacar:

Podemos hacer uso simultáneo de las propiedades Body y AltBody cuando queremos mandar un mensaje con formato Html y es posible que el destinatario acepte sólo correos en formato texto. Al definir las dos propiedades, Body y AltBody, no es necesario ejecutar el método IsHTML(True) cuando a Body se le asigna un mensaje en formato Html.

La segunda y más importante a tener en cuenta es el hecho de que el éxito en el envío del mail no depende sólo de la clase PHPMailer sino que es vital el tiempo que el servidor smtp necesita para autenticar al usuario así como el tiempo que tarda dicho servidor en enviar un correo.

Por tanto es fundamental proporcionarle al servidor el tiempo que le hace falta, ello se consigue de dos formas: Por un lado aumentando el valor de la propiedad Timeout pasando por ejemplo de 10 a 30 segundos; Y por otro lado reintentando el envío del mensaje un número razonable de veces (por ejemplo 5), dejando un tiempo de espera (por ejemplo 5 segundos), entre cada intento.

Mediante la combinación de estos 3 factores: Propiedad Timeout, número de intentos para enviar el mensaje y tiempo de espera entre cada intento evitaremos los dos mensajes de error más frecuentes que nos puede dar la clase PHPMailer, relacionados con el tiempo que necesita el servidor smtp para realizar su función, y que son:

- SMTP Error: The following recipients failed
- SMTP Error: Could not authenticate

Ejemplo algo más complicado

El objetivo de esta segunda aplicación de ejemplo va a ser enviar un correo con un fichero adjunto.

Esta aplicación va a estar formada por:

- Un archivo ejemplo2.html que contendría el formulario a completar por el usuario y que incluirá un archivo adjunto
- Un archivo llamado ejemplo2.php que recogerá los datos del formulario y utilizará la clase PHPMailer para hacer el envío de mail
- Un subdirectorío llamado includes donde estarán los ficheros class.phpmailer.php y class.smtp.php

Para poder enviar el fichero adjunto vamos a emplear en el formulario una etiqueta INPUT de tipo FILE. Para que funcione es necesario:

- Que el formulario use el método post
- Que el atributo enctype tenga el valor multipart/form-data
- Además, al formulario hay que añadirle un campo oculto de nombre MAX_FILE_SIZE, al cual le daremos el valor en bytes del tamaño máximo del archivo a descargar. Voy a limitar a 300kb (307200 bytes) el tamaño máximo del archivo.

Esto en el archivo ejemplo2.html

Código de ejemplo2.php:

El código del ejemplo lo encuentran en la carpeta PHPMailer adjunta con los ejemplos.