

# PEAR: Modulos y Extensiones de PHP

**CONSOL 2006.**

**CONGRESO NACIONAL DE SOFTWARE LIBRE**

**INSTITUTO POLITECNICO NACIONAL.**

**Agosto 17, 2006. Ciudad de Mexico**

**Farid Alfredo Bielma Lopez.**

**[fbielma@fbielma.org](mailto:fbielma@fbielma.org)**

**[http://fbielma.org/conferencia/consol/PEAR\\_PHP.pdf](http://fbielma.org/conferencia/consol/PEAR_PHP.pdf)**

# Agenda

- Que es PEAR?
- Paquetes PEAR
- El instalador de PEAR
- HTML\_QuickForm
- PEAR::DB
- Futuro de PEAR
- Conclusiones

# The PEAR project

El PEAR es un proyecto Open Source creado por el PHP Group con el objetivo de crear una gran biblioteca de clases de calidad escritas en PHP. Todas ellas accesibles desde un repositorio web común y fácilmente instalables por medio de un sistema de gestión de paquetes.

# Que es PEAR?

- Es una librería estructurada de Código Abierto para desarrolladores de PHP.
- Un sistema de distribución de código (tgz) y mantenimiento de paquetes.
- El instalador PEAR.
- El sitio web, la lista de correo, y sitios de descargas para servir de soporte a la comunidad PHP/PEAR.

# Funcionalidades agregadas por PEAR

Un Sistema General de Reporte de Errores (PEAR\_Error)

Encriptacion.

Abstraccion de Base de Datos.

Formularios HTML y templates.

Funciones de E-mail (sending, processing)

Networking

XML-Parser + XML-RPC

# Paquetes PEAR

Algunos de los paquetes PEAR mas usados son:

- PEAR::DB
- ADODB
- HTML\_QuickForm
- Smarty
- XML\_RPC
- MAIL/MIME
- X-debug

# El instalador PEAR

Para las versiones de PHP inferiores a la 4.3.0 sera necesario obtener el instalador, en Linux/UNIX/BSD ejecuta:

```
# lynx -source http://go-pear.org | php
```

Este script instala todos los componentes necesarios.

En windows ejecuta el batch go-pear.bat, el setup te hara una preguntas, finalmente quedara instalado en el path que hayas especificado.

# Instalador un paquete manualmente

No es recomendable, pero es util cuando no se tiene acceso a una shell.

A manera de demostracion en nuestro document root crear un directorio llamado incluir al mismo nivel que el htdocs, en donde podemos ingresar via HTTP o FTP.

La instalacion consiste en los siguientes pasos:

5. Descargar el paquete:

Apuntando a la url <http://pear.php.net> y utilizar el package browser para ver los paquetes disponibles.



# Instalador un paquete manualmente

2. Subir el paquete al servidor.

Despues de descargar el archivo .tgz y de extraer su contenido en un directorio temporal, cargar el archivo por el metodo usado y copiarlo en el directorio /htdocs/incluir.

3. Ajustar la directiva `include_path`.

Si tienes acceso al `php.ini` en tu sitio agregar el directorio `incluir` dentro de la directiva indicada, quedaria asi:

```
/var/www/fbielma.org/incluir
```

# Si no tienes acceso a php.ini

Si no tienes acceso al php.ini, tienes que agregar el `include_path` en cualquier script donde quieras usar el paquete:

```
<?php
ini_set("include_path", '/var/www/www.example.com/includes/' . PATH_SEPARATOR .
    ini_get("include_path"));
?>
```

4. Después de realizar la instalación, ya puedes usar el paquete.

# Características del Instalador

Descarga e instala paquetes PEAR.

Crea paquetes que puedan ser subidos al sitio web de PEAR.

Automaticamente revisa las dependencias de los paquetes, versiones de PHP y extensiones de PHP.

Puede crear e instalar paquetes de terceros.

# Funcionamiento del Instalador

Usa XML-RPC para comunicarse con el sitio oficial de PEAR.

Descarga el archivo .tgz de un paquete.

El archivo package.xml describe el contenido del paquete, así como sus dependencias.

Extrae todos los archivos necesarios al directorio de instalación local.

# Comandos del Instalador

```
# pear list
```

```
# pear install Mail_mime
```

```
# pear upgrade
```

```
# pear uninstall
```

```
# pear help uninstall
```

# Uso de paquetes

Las clases pueden ser incluidas usando:

```
<?php require_once 'HTTP/Request.php';?>
```

Muchos paquetes usan una arquitectura basada en conectores:

```
<?php  
require_once 'DB.php';  
$dbh = DB::connect('psql://user:pass@host');  
?>
```

# Manejo de Errores

Los errores son instancias de `PEAR_Error` que contiene:

- Mensaje de error y código opcional
- Función `backtrace` para ayudar con la depuración

El manejo de errores puede ser establecido globalmente o para instancias particulares:

- Impresión del mensaje
- Función `die` al ocurrir un error
- Manejo del error usando una llamada a función
- Será reemplazado con `PEAR_ErrorStack` y/o excepciones en PHP5

# Ejemplo del manejo de Errores

```
Require 'DB.php';  
$db= DB::connect('pgsql://farid:mi_password@host/example');  
$sth= $db->query('select datos from escuela');  
If(DB::isError($sth)){  
    echo "error!";  
}  
Else{  
    //display results  
}
```



# Autenticación

```
Require 'Auth/Auth.php';  
$a= new Auth('File','/etc/passwd');  
$a->start();  
If(! $a->getAuth()){  
    //the user is not logged  
    exit();  
}
```

# Como crear nuevos paquetes

Edita el archivo “package.xml”, este archivo contiene información sobre el paquete, estado, dependencias, changelog, y archivos incluidos:

- PEAR\_PackageFileManager te ayudará
- Abre la terminal y mueve el paquete a una carpeta
- Corre el comando `pear package`
- Prueba el paquete resultante
- Súbelo a [pear.php.net](http://pear.php.net)

# HTML\_QuickForm

HTML\_QuickForm es una clase PEAR. El Repositorio PEAR es una colección para agregar librerías y módulos para PHP.

HTML\_QuickForm hace el trabajo con formularios de manera más fácil, como reservar valores enviados, muestra mensajes de error, agrega una vista consistente para los elementos del Formulario y simplifica las capacidades de File Uploads.

# Instalacion

Las instrucciones generales de instalación puede encontrarla en el Sitio Oficial:

[http://pear.php.net/package/HTML\\_QuickForm](http://pear.php.net/package/HTML_QuickForm)

En Unix la instalación se realiza:

```
# pear install -a HTML_QuickForm.version.tgz
```

# Ejemplo HTML\_QuickForm

```
<?
```

```
Require 'HTML/QuickForm.php';
```

```
//Se crea el contenedor llamado $form
```

```
$form=new HTML_QuickForm('nombre');
```

```
//Se agregan elementos al formulario:
```

```
$form-> addElement('text','nombre','Tu nombre:');
```

```
$form-> addElement('submit','enviar','Enviar');
```

# Ejemplo HTML\_QuickForm

```
// Se llama a la Funcion process()
```

```
If($form-> validate()){  
    $form->process('consol');  
}  
Else{  
    $form->display();  
}
```

# Ejemplo HTML\_QuickForm

```
// El metodo process() llama a la funcion consol, la cual  
recoge // los datos del formulario como array.
```

```
function consol($data){  
    echo $data['nombre'].', recibe un saludo del consol';  
}
```

# Tipos de Elementos soportados

Los tipos de elementos soportados por HTML\_QuickForm son:

text	autocomplete	password
textarea	hidden	select
radio	submit	reset
button	image	file
header	link	html



# Reglas de validacion

El método `addRule()` agrega reglas de validacion, para un elemento

especifico del Formulario.

```
$form->addElement('text','nombre','Nombre: ');
```

//Agrega la regla "required" al elemento nombre

```
$form->addRule('nombre','Teclea tu nombre.', 'required');
```

//Agrega la regla "minlength" al elemento nombre

```
$form->addRule('nombre','Mayor a 3 letras.', 'minlength',3);
```

# Reglas de validacion

required		La Entrada de datos es requerido para el elemento.
Maxlengt h	\$lengt h	El dato puede tener \$length caracteres como máximo.
Minlength	\$lengt h	Los caracteres que puede tener como mínimo el elemento.
Email		El dato debe tener sintacticamente una dirección de correo válida.

# Reglas de validacion

Numeric		El elemento debe contener solamente un entero valido o un numero decimal.
Uploadedfile		El elemento debe contener un archivo exitosamente cargado al servidor.
maxfilesize	\$size	El archivo debe tener no mas de \$size tamaño permitido para subirlo al servidor.
filename	\$file_rx	El archivo cargado debe tener un nombre de acuerdo a la expresión: \$file_rx

# File Uploads

```
require 'HTML/QuickForm.php';  
$form = new HTML_QuickForm();  
$form->addElement('file','uploaded_file','Tu archivo:');  
//Archivo menor o igual a 128kbytes  
$max_size = 131072;  
// Estar seguro que se subio al Servidor.  
$form->addRule('uploaded_file','Por favor, sube el archivo','uploadedfile');  
$form->addRule('uploaded_file','Demasiado grande','maxfilesize',$max_size);  
$form->addRule('uploaded_file','Enviar solamente HTML ','mimetype','text/html');  
$form->addElement('submit','save','Enviar archivo');  
//No esta permitido subir un archivo mayor que 128k  
$form->setMaxFileSize($max_size);
```

# File Uploads

```
if ($form->validate()) {
    $form->process('Mover');
} else {
    $form->display();
}function Mover($data) {
    global $form;
    $new_name = strstr($data['uploaded_file']['name'],'\\',");
    $new_name = str_replace('..',",$new_name);
    $file =& $form->getElement('uploaded_file');
    if ($file->moveUploadedFile('/tmp',$new_name)) {
        print "El archivo ha sido cargado en /tmp/$new_name.";
    }
    ?>
```

# PEAR::DB

PEAR::DB provee una estable relación de métodos para usar una base de datos relacional, no importando la que estes usando.

# Ejemplo PEAR::DB

```
requiere 'DB.php';  
$dbh=DB::connect('mysql://user:password@host/database');  
$sth=$dbh->query('Select nombre, carrera from escuela');  
  
if($sth->numRows()){  
    while($row = $sth->fetchRow()){  
        print "<br>$row[0]-$row[1]";  
    }  
}
```

# Ejemplo PEAR::DB

```
else{  
    print "No results";  
}
```



# Introduccion DSN

msql	Microsoft SQL Server
mysql	MySQL
odbc	ODBC
pgsql	PostgreSQL
oci8	Oracle 8

# El futuro de PEAR

Canal de soporte:

- Te permite distribuir tu propia aplicación usando PEAR al proveer un servidor de paquetes
- Mejor soporte para PHP5
- Manejo de errores mejorado
- Interfaces
- Excepciones
- Aseguramiento de la calidad mejorado

# Preguntas

**Farid Alfredo Bielma Lopez**

**Email: fbielma@fbielma.org**

**MSN: fbielma@hotmail.com**

**[http://fbielma.org/conferencia/consol/PEAR\\_PHP.pdf](http://fbielma.org/conferencia/consol/PEAR_PHP.pdf)**