



# Los Frameworks de PHP agilizan tu trabajo

EDITORIAL

Aprende más sobre este tema con la guía Zend, la guía Symfony y te recomendamos que veas "Frameworks PHP hispanos". Los Frameworks ayudan en el desarrollo de software, proporcionan una estructura definida la cual ayuda a crear aplicaciones con mayor rapidez. Ayuda a la hora de realizar el mantenimiento del sitio gracias a la organización durante el desarrollo de la aplicación.



Damián Pérez Valdés

julio 31 2007

Aprende más sobre este tema con la [guía Zend](#), la [guía Symfony](#) y te recomendamos que veas ["Frameworks PHP hispanos"](#).

Los Frameworks son desarrollados con el objetivo de brindarles a los programadores y diseñadores una mejor organización y estructura a sus proyectos.

Se utiliza la Programación Orientada a Objetos ([POO](#)), permitiendo la reutilización de nuestro código. A continuación las características de algunos de los Frameworks para [PHP](#) más usados.

## Zend Frameworks

El [Zend Frameworks](#) es simple, no necesita instalación especial, requiere PHP 5 e incorpora el patrón MVC.

Se debe [descargar](#) y copiarlo hacia nuestro servidor local, veamos a través de un pequeño ejemplo cómo podemos crear un lector de RSS. Los ficheros que creemos podemos copiarlos dentro del directorio "library".



```
// Incluimos la dirección de nuestro fichero rss que deseamos imp
$feed = Zend_Feed::import('http://localhost/rss.php');

// Se recorrerán todos los item del fichero, mostrando el titulo
foreach ($feed->items as $item) {
    echo "<p>" . $item->title() . "<br />";
    echo $item->link() . "</p>";
}
?>
```

## Symfony

Diseñado con el objetivo de optimizar la creación de las aplicaciones web, con el uso de sus características. Posee una librería de clases que permiten reducir el tiempo de desarrollo.

[Symfony](#) está desarrollado en PHP5, se puede utilizar en plataformas \*nix (Unix, Linux) y Windows. Requiere de una instalación, configuración y líneas de comando, incorpora el patrón MVC, soporta AJAX, plantillas y un gran número de bases de datos.

Luego de [descargar](#), debemos desempaquetar hacia un directorio dentro de nuestro servidor web. Puede configurar el archivo “symfony.bat” e introducir la ruta completa del archivo “php.exe” en la línea 34.

```
set PHP_COMMAND=c:/xampp/xampp/php/php.exe
```

Si nos encontramos en el directorio “sf\_sandbox” podemos ir a las líneas de comando y conocer la versión del framework: `symfony -V`. Podemos verificar la creación de nuestro proyecto introduciendo en el navegador, por ejemplo:

```
http://localhost/sf_sandbox/web/frontend_dev.php
```

La estructura de los directorios es:

```
sf_sandbox/ // directorio del proyecto
apps/       // archivos de la aplicación
batch/      // procesos
cache/      // cache
config/     // configuración
```



```
log/          // archivos de log
plugins/      // plugins
test/         // tests
web/          // directorio público
```

Si deseamos crear un weblog debemos crear el archivo “`schema.yml`” en el directorio “`sf_sandbox/config/`”. Después de haberlo creado podemos usar las siguientes líneas de comando dentro del directorio “`sf_sandbox/`”:

- `symfony propel-generate-crud frontend post Post` : crear y listar Post.
- `symfony propel-generate-crud frontend comment Comment` : crear y listar comentarios.
- `symfony clear-cache` : limpiar la cache.

Los modulos serán creados en “`sf_sandbox\apps\frontend\modules\`”. Podemos ver el resultado:

- `http://localhost/sf_sandbox/web/frontend_dev.php/post`
- `http://localhost/sf_sandbox/web/frontend_dev.php/comment`

Se puede acceder a los archivos y modificar las plantillas a nuestro gusto.

## Seagull

El framework [Seagull](#), nos permite realizar una programación modular, posee un Sistema de Gestión de Contenidos (CMS).

Entre sus características de encuentran su compatibilidad con PHP 4 y PHP 5, ORM integrado, incorpora el patrón MVC, uso de templates, soporte para múltiples bases de datos, validación de datos, alto nivel de configuración, autenticación, integración de librerías PEAR.

Debemos descargar el framework, luego descomprimirlo hacia un directorio dentro de nuestro servidor web. Luego introducimos la dirección en nuestro navegador para realizar su instalación. Ejemplo: `http://localhost/seagull/www/`.

Debemos completar los 6 pasos siguientes:



generado o descargarlo en “[seagull/](#)”. Si se crea correctamente podremos continuar al siguiente paso.

- Detección: aquí se detectará los módulos disponibles, así como la configuración y versión de PHP.
- Conexión con las bases de datos: debemos configurar los datos necesarios para realizar la conexión con el servidor de base de datos.
- Configuración de la base de datos: introducir el nombre de la base de datos. Se puede utilizar una existente.
- Creación del Usuario Administrador: se creará la cuenta de usuario que utilizaremos para administrar a Seagull Framework.

Luego de Finalizar su instalación podemos crear contenidos visitando

[http://localhost/ seagull/www/](http://localhost/seagull/www/) e introduciendo el usuario y clave de administración.

## Prado

[Prado](#) está basado en componentes eventos con el objetivo de acelerar el desarrollo de aplicaciones web usando PHP 5.

El concepto del desarrollo de aplicaciones en Prado es diferente, se utilizan componentes, eventos y propiedades en vez de procedimientos, URL y parámetros.

Este Framework combina especificaciones en un archivo [XML](#), plantillas [HTML](#) y una clase PHP. Prado, cuenta con soporte para AJAX, validación, autenticación, plantillas, múltiples bases de datos.

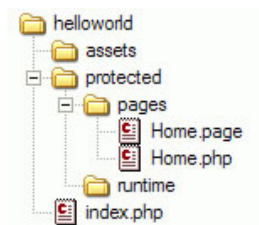
Entre los beneficios que podemos encontrar para el desarrollo de aplicaciones web se encuentran:

- Reutilización: los códigos y componentes pueden ser reutilizados.
- Fácil uso: la creación y uso de componentes es fácil.
- Funcionamiento: utiliza una técnica de caché para asegurar el funcionamiento de aplicaciones basadas en el.
- Integración: permite la separación del contenido y la presentación.



## Hello World:

Esta aplicación cuenta con los archivos `index.php`, `Home.page` y `Home.php`, organizados con la siguiente estructura, la cual puede ser modificada:



- `assets` : almacena archivos privados publicados.
- `protected`: aquí se almacenarán los archivos que serán usados.
- `runtime`: se almacena la información runtime de la aplicación, así como su estado, este directorio debe tener acceso de escritura por el servidor web.
- `pages`: aquí se almacenan todas las páginas de Prado.

## Descripción del contenido de los archivos:

**Index.php:** código de entrada para la aplicación en Prado. Este fichero es necesario para todas las aplicaciones. El contenido del archivo es el siguiente:

```
require_once('path/to/prado.php'); // incluye el código de prado
$application=new TApplication; // se crea una instancia de la apl
$application->run(); // ejecuta la aplicación
```

**Home.page:** Plantilla de la página. La misma describe la posición de los componentes. Aquí se utilizan los componentes `TForm` y `TButton`.

```
<html>
<head>
<title>Hello World Demo- Prado</title>
</head>
<body>
<com:TForm>
<com:TButton Text="Click me" OnClick="buttonClicked" />
</com:TForm>
```



**Home.php:** clase para la página Home.page, contiene los métodos para el evento OnClick del botón.

```
class Home extends TPage
{
    public function buttonClicked($sender,$param)
    {
        $sender->Text="Hello World!";
    }
}
```

En el ejemplo anterior tendremos un botón con el texto “Click me” al oprimirlo cambiará el texto por “Hello World!”. Podemos ver los demos que incluye este framework

<http://localhost//prado/demos/>.

Si presentamos algún problema podemos verificar sus requerimientos

<http://localhost/prado/requirements/index.php>.

Espera [la segunda parte](#) de este artículo sobre los Frameworks de PHP, en donde hablaremos sobre CodeIgniter, CakePHP, PHP on TRAX y entraremos en detalle sobre MVC.

0

Compartir  
Compartir

Twittear



Damián Pérez Valdés

Webmaster, Administrador de Sistemas, con experiencia en desarrollo web y de aplicaciones.

<http://damianadriel.blogspot.com/>

Otros artículos que podrían interesarte