



# PHP Básico

## PHP para principiantes

Copyright (c) 2006 Guillermo Valdez Lozano. E-mail: [guivaloz\(en\)gmail.com](mailto:guivaloz(en)gmail.com)  
<http://www.movimientolibre.com>

Se otorga permiso para copiar, distribuir y/o modificar este documento bajo los términos de la Licencia de Documentación Libre de GNU, Versión 1.2 o cualquier otra versión posterior publicada por la Free Software Foundation; sin Secciones Invariantes ni Textos de Cubierta Delantera ni Textos de Cubierta Trasera.

Una copia de la licencia está disponible en <http://www.movimientolibre.com/gfdl-es.php>



# ¿Qué es PHP?

- PHP (acrónimo de "PHP: Hypertext Preprocessor") es un lenguaje de "código abierto" interpretado, de alto nivel, embebido en páginas HTML y ejecutado en el servidor.
- ¿Qué se puede hacer con PHP?
  - Scripts del lado del servidor.
  - Scripts de línea de comandos.
  - Escribir aplicaciones de interfaz gráfica (PHP-GTK).



# Conocimientos previos

- Indispensable
  - HTML
- Recomendado
  - CSS
  - Javascript (validación de formularios)
  - SQL (aplicado a su base de datos favorita)
- Además es bueno saber...
  - Configuración del Apache HTTPD
  - Bash scripts (para hacer scripts de respaldo)



# Aprendizaje

- Documentación PHP
  - <http://www.php.net>
  - Descargar archivo tar.gz
- Programar y probar :-)
- Ver código de otros programas libres

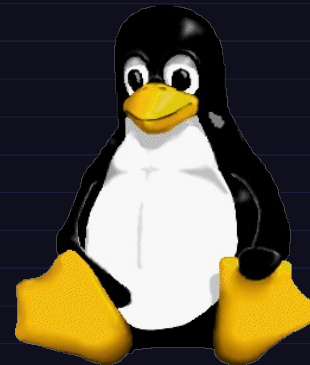
Otros sitios de interés:

- Zend Developer Zone, PHP 101 <http://devzone.zend.com/node/view/id/627>
- PHP video tutorials (en inglés) <http://www.phpvideotutorials.com/>



# Instalación

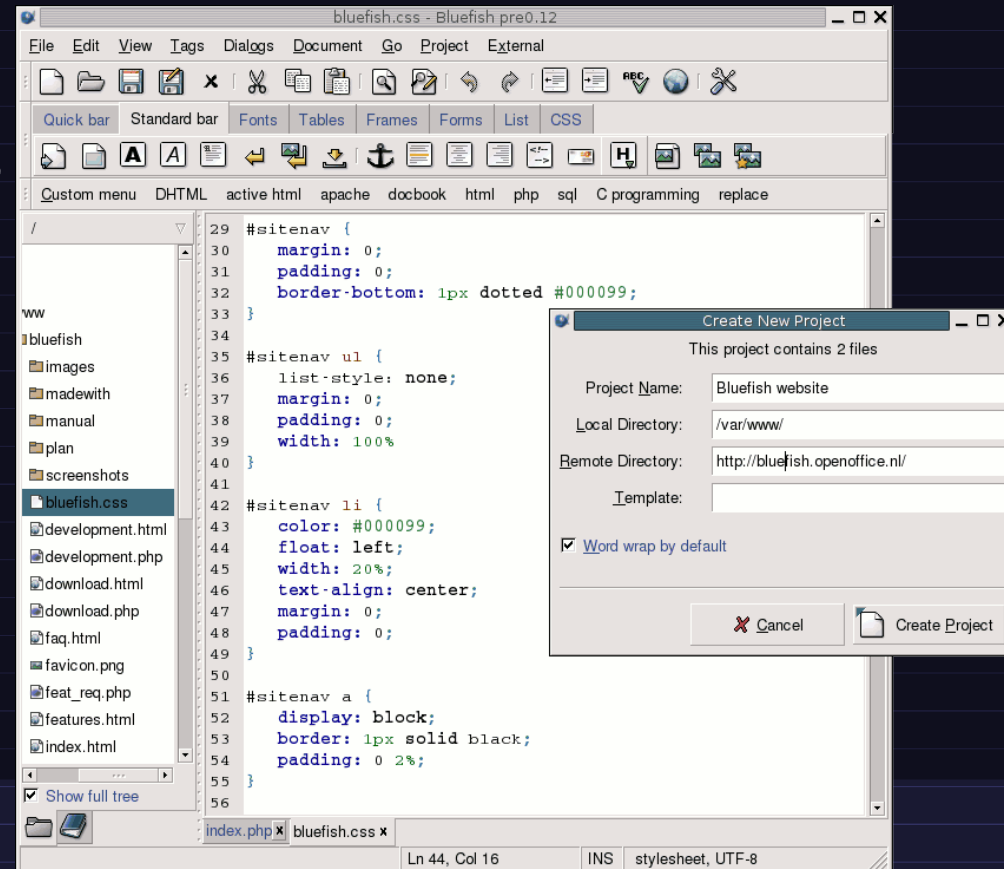
- GNU/Linux con acceso a LAN o internet
- Servidor de páginas web Apache
  - Apache 1.3
  - Apache 2.0
- PHP
  - PHP 4 (descontinuado)
  - PHP 5
- Configurar Apache para que use el PHP
- Base de Datos (opcional)





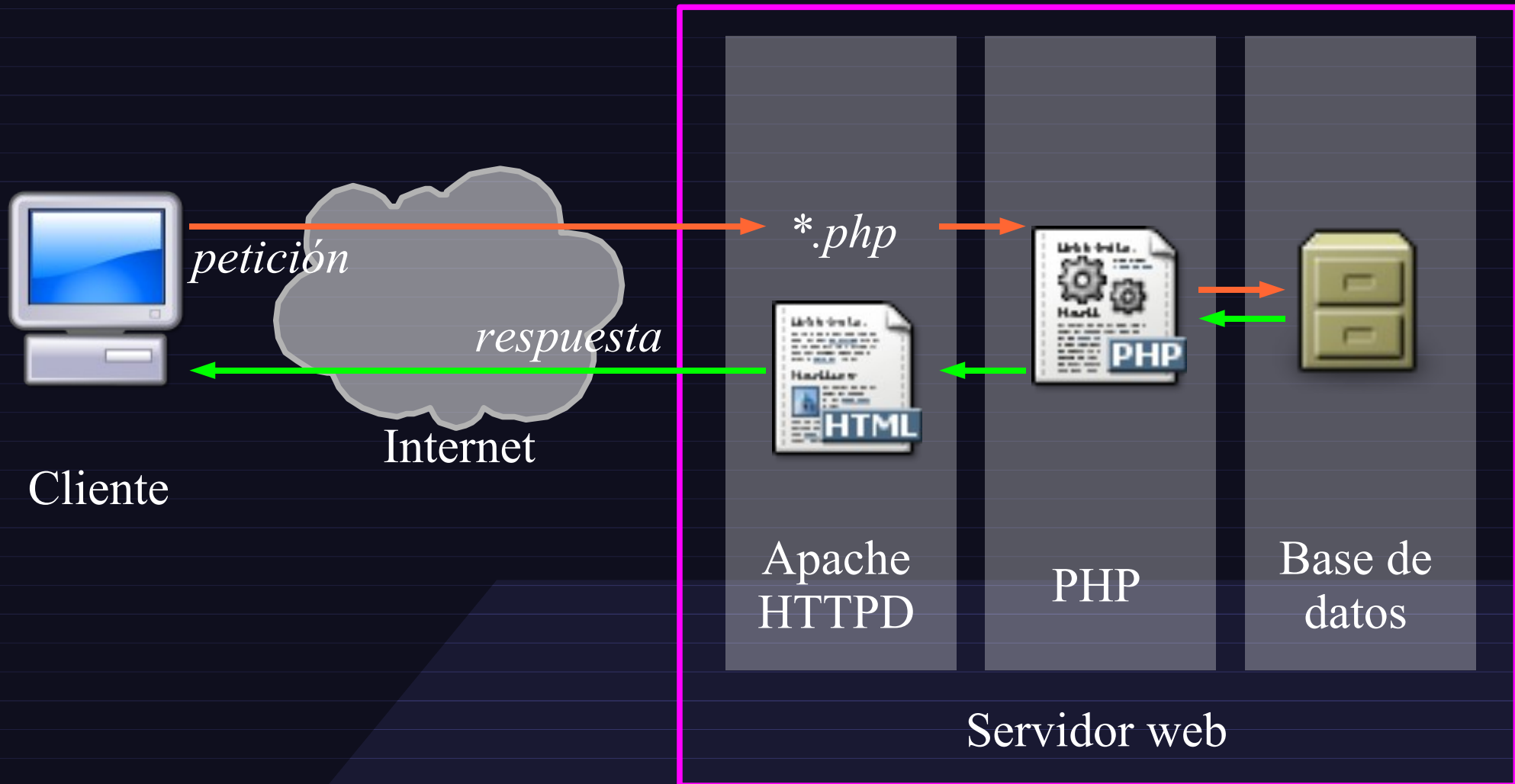
# Herramientas de desarrollo

- Software libre
  - *Cualquier editor de texto*
  - BlueFish (GTK+)
  - Screem (GTK+)
  - Quanta (KDE)
  - VIM (consola)
- Software comercial
  - Zend Platform
  - Adobe Dreamweaver





# Funcionamiento de las Páginas Dinámicas





# Embebiendo PHP en HTML

hola-mundo-1.php

```
<html>
  <title><?php echo "Prueba"; ?></title>
  <body>
    <?php echo "Hola Mundo!"; ?>
  </body>
</html>
```

El cliente recibirá esta página:

```
<html>
  <title>Prueba</title>
  <body>
    Hola Mundo!
  </body>
</html>
```





# Incluyendo archivos

plantilla.php

```
<html>
  <title><?php echo $titulo; ?></title>
  <body>
    <?php echo $contenido; ?>
  </body>
</html>
```

hola-mundo-2.php

```
<?php
  $titulo      = "Prueba";
  $contenido   = "Hola Mundo !";
  include("plantilla.php");
?>
```



# Sintaxis general

- El nombre del archivo con PHP puede tener la extensión .php para diferenciarlo de los html.
- Use `<?php` al inicio y `?>` al final de todos los bloques de instrucciones en PHP.
- Las instrucciones se separan por ";".
- Los espacios, tabulaciones y avances de línea no afectan a la ejecución del programa.
- No hay necesidad de inicializar las variables o de fijar el tipo de dato que vayan a contener.



# Sintaxis de variables y comentarios

- Toda variable comienza con \$, seguida por una letra o guion bajo, luego puede tener cualquier carácter alfanumérico.

Ejemplos: \$arreglo, \$nombre\_2, \$Usuario, \$\_GET

- Los nombres de las variables son sensibles a las mayúsculas y minúsculas.

\$a y \$A no son la misma variable.

- Pueden agregarse comentarios:

```
/* Estilo lenguaje C */
```

```
// Estilo C++
```

```
# Estilo shell script
```



# Tipos de datos

- Enteros
- Punto Flotante
- Cadenas
- Boleano
- Arreglos
- Objetos

```
<?php
$contador    = 5;
$pi          = 3.1416;
$mensaje     = 'Hola Mundo !';
$html        = "<b>$mensaje</b>";
$resultado   = true;
$vocales     = array('a', 'e',
                    'i', 'o', 'u');
$status_desc = array(
    'A' => 'Activos',
    'B' => 'Inactivos';
echo 'Usuarios '. $status_desc['A'];
?>
```



# Operadores más usados

- `!`, `++`, `--`
- `*`, `/`, `%`
- `+`, `-`, `.`
- `<`, `<=`, `>=`, `>`
- `==`, `!=`, `===`, `!==`
- `&&`, `||`
- `=`, `+=`, `-=`

minutos-de-hoy.php

```
<?php
$hoy      = date('H:i:s');
$arr      = explode(':', $hoy);
$minutos  = intval($arr[0]*60
                  + $arr[1]);
echo "Minutos: <b>$minutos</b>";
?>
```



# Estructuras de control: If

- Para mantener un conjunto de instrucciones dentro de una estructura de control use las llaves. Puede omitirlas si sólo hay una instrucción.

```
<?php
    if ($error != ' ') {
        echo "Mensaje error: $error";
    } elseif ($contador == 0) {
        echo "No hay datos.";
    } else {
        echo "Hay $contador datos.";
    }
?>
```

```
<?php
    if ($error != ' ')
        echo "Mensaje error: $error";
    elseif ($contador == 0)
        echo "No hay datos.";
    else
        echo "Hay $contador datos.";
?>
```



# Estructuras de control: Switch

```
<?php
// Estructura del switch
switch ($estatus) {
    case 'A':
        $titulo = 'Usuarios activos';
        break;
    case 'B':
        $titulo = 'Usuarios inactivos';
        break;
    default:
        $titulo = 'Estatus incorrecto';
}
?>
```



# Estructuras de control: for y foreach

```
<?php
// Estructura del for
for ($i=1; $i<10; $i++) {
    echo "<p>Fila No. $i</p>\n";
}
?>
```

```
<?php
// Estructura del foreach
$horarios = array('Matutinos', 'Vespertinos',
    'Nocturnos');
foreach ($horarios as $h) {
    echo "<p>Horario $h</p>\n";
}
?>
```





# Funciones

- PHP cuenta con miles de funciones para todos los propósitos. Consulte la documentación.
- Puede definir sus propias funciones y colocarlas en cualquier parte del archivo o en los archivos que incluya.

```
<?php
function min_a_seg($seg) {
    return $seg*60;
}
// Calcular la cantidad de seg. de 30 min.
echo "Media hora tiene".min_a_seg(30)." segundos.";
?>
```



# Funciones

- Las variables que se declaren dentro de las funciones no afectan a las que están fuera de ella.
- Puede definir los valores por defecto de los parámetros de la función, de esta forma:

```
<?php
function tres_numeros($a=1, $b=2, $c=3) {
    return "<p>Tres números: $a, $b, $c.</p>\n";
}
// Muestra 1, 2 y 3
echo tres_numeros();
// Muestra 10, 20 y 30
echo tres_numeros(10, 20, 30);
?>
```



# Variables del entorno web

## **`$_GET`**

Si el URL es `usuario.php?id=14` con `$_GET['id']` obtendrá el 14.

## **`$_POST`**

Para obtener los campos de un formulario.

## **`$_COOKIE`**

## **`$_REQUEST`**

Obtener un parámetro por los tres métodos, primero por `$_GET`, luego por `$_POST` y al final por `$_COOKIE`



# Variables del servidor

`$ _SERVER [ "SERVER_NAME" ]`

Nombre del servidor.

`$ _SERVER [ "PHP_SELF" ]`

Nombre del archivo PHP *en ejecución*.

`$ _SERVER [ "REMOTE_ADDR" ]`

IP del cliente.



# Sesiones

- Las sesiones sirven para pasar valores de una página a otra(s).
- No es método seguro para sustituir un sistema de autenticación.

sesion-primer.php

```
<?php
    session_start();
    session_register('nombre');
    session_register('edad');
    $nombre = "Fulano de Tal";
    $edad   = "30";
?>
```

sesion-segundo.php

```
<?php
    session_start();
    echo "Nombre: $nombre<br>";
    echo "Edad: $edad<br>";
?>
```



# Objetos

- Un objeto empaqueta *propiedades y métodos* (variables y funciones).
- Respecto a PHP 4, en PHP 5 se cambia la sintaxis para los objetos y mejora el manejo de éstos.

```
<?php
class Usuario {
    public $nombre;
    public $edad;
    public function cletras() {
        return strlen($nombre);
    }
}
?>
```

```
<?php
$usted = new Usuario();
$usted->nombre = 'Memo';
$usted->edad = 30;
$n = $usted->cletras();
?>
```



# Preguntas, comentarios o dudas

```
<?php echo "FIN"; ?>
```