# Introducción a PHP

# Qué es PHP

## La historia



PHP es un <u>acrónimo recursivo</u> que significa PHP Hypertext Pre-processor (inicialmente PHP Tools, o, Personal Home Page Tools).

Fue creado originalmente por Rasmus Lerdorf en 1994. Sin embargo la implementación principal de PHP es producida ahora por Zend Company y sirve como el estándar para PHP al no haber una especificación formal. Publicado bajo la PHP License, la Free Software Foundation considera esta licencia como software libre.

PHP surgió como una mezcla entre los tradicionales lenguajes de C y Perl, intentando dar una solución practica a los problemas propios del desarrollo Web en vez del resto de los lenguajes que eran de propósito general.

Luego, PHP es reescrito (por la empresa Zend Company) dando vida al PHP4.

PHP es un lenguaje interpretado de propósito general ampliamente usado y que está diseñado especialmente para desarrollo Web y puede ser incrustado dentro de código HTML. Generalmente se ejecuta en un servidor Web, tomando el código en PHP como su entrada y creando páginas Web como salida. Puede ser desplegado en la mayoría de los servidores Web y en casi todos los sistemas operativos y plataformas sin costo alguno.

## ¿Qué se puede hacer con PHP?

- Generar páginas con contenidos dinámicos.
- Enviar y recibir cookies.
- Manejo de archivos y directorios. Lectura y modificación de archivos de texto.
- Acceso y manipulación de Bases de Datos.
- Manejo de gráficos, sonidos y animaciones.
- Creación de imágenes, archivos PDF y películas Flash, XHTML y archivos XML. Envío de e-mails y mucho más...

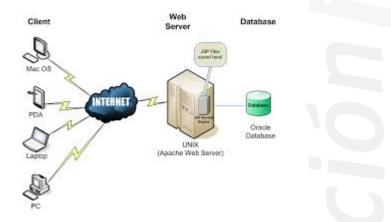
# Organización y Conceptos

#### El Modelo Cliente - Servidor

Una de las arquitecturas más expandidas en las aplicaciones de hoy en día es la que se conoce como modelo cliente-servidor.

En este modelo existen básicamente dos procesos, los cuales se comunican mediante algún mecanismo diseñado a tales efectos (por ejemplo, una red que los conecta). Estos procesos son conocidos como cliente y servidor. El objetivo de este tipo de arquitecturas es el de repartir la carga entre distintas capas de la aplicación, lo cual conlleva un mejor aprovechamiento del hardware entre otros.

El cliente tiene a su cargo todo el procesamiento referido a la interacción con el usuario, mientras que el servidor se encarga de resolver procesos complejos, como acceder a bases de datos o realizar cálculos basados en reglas de negocio. Generalmente un servidor da servicio a más de un cliente a la vez.



El funcionamiento del modelo se basa en un esquema de solicitud-respuesta. El cliente realiza una solicitud al servidor y éste le responde dándole el resultado de la operación solicitada. Toda esta interacción debe ser realizada siguiendo un conjunto de reglas llamado protocolo.

#### Protocolo HTTP

HTTP ( HyperText Transfer Protocol ) o Protocolo de Transferencia de Hipertexto es un conjunto de reglas que rigen la transferencia de datos en una comunicación Web. Su finalidad es la transferencia de Hipertexto ( texto con componentes y enlaces a otros textos ) pero en la actualidad es utilizado tanto para la transferencia de Hipertexto (HTML) cómo para transferencia de Ficheros, datos ( XML ), etc.

HTTP es un protocolo sin estado, es decir, que no guarda ninguna información sobre conexiones anteriores. El desarrollo de aplicaciones Web necesita frecuentemente mantener estado. Para esto se usan las cookies (tema que veremos mas adelante), que es información que un servidor puede almacenar en el sistema cliente. Esto le permite a las aplicaciones web instituir la noción de "sesión", y también permite rastrear usuarios ya que las cookies pueden guardarse en el cliente por tiempo indeterminado.

En el siguiente articulo podrás ver mas información: Acerca del protocolo HTTP

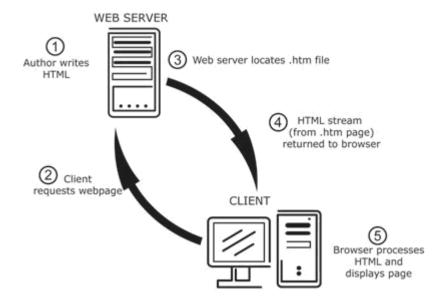
#### Páginas Estáticas

Cuando un cliente Web ha recibido todo el flujo de texto desde el servidor, debe presentar de alguna forma esa

información al usuario a través de un navegador de Internet. Para especificar el formato de las páginas se utiliza el lenguaje HTML, el cual es un conjunto de etiquetas que especifican los diferentes aspectos de la visualización, como ser tipo de letra, tamaño, color, etc. y las cuales son interpretadas por el navegador.

Las páginas desarrolladas íntegramente en HTML se conocen como páginas estáticas, debido a que su contenido no se altera en el momento en que el servidor resuelve el pedido.

Esto significa que lo que el cliente recibe es exactamente lo que está almacenado en el servidor Web.

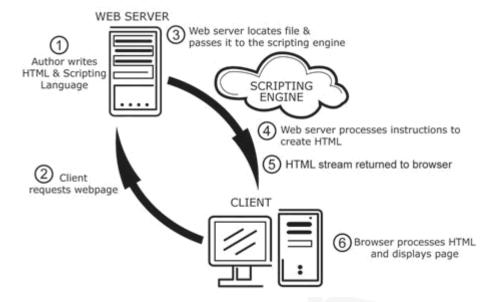


# Páginas Dinámicas

Los archivos PHP son ejecutados en el navegador e interpretados por el servidor Web Apache.

Las aplicaciones de PHP no se necesitan compilar. El navegador interpreta el HTML y puede recibir una respuesta en HTML del script de PHP una vez procesado en el Web Server. Se llaman scripts ya que son interpretados al momento de la ejecución ó carga de la página Web.

PHP interpreta el texto que se encuentra entre los caracteres especiales que delimitan el inicio y fin de código. El intérprete ejecuta entonces todo este código contenido entre los delimitadores. Al encontrar una etiqueta de fin de código, esto le dice al intérprete que siga ignorando el código siguiente. Este mecanismo permite embeber código PHP dentro de HTML: todo lo que está fuera de las etiquetas PHP se deja tal como está, mientras que el resto se interpreta como código.

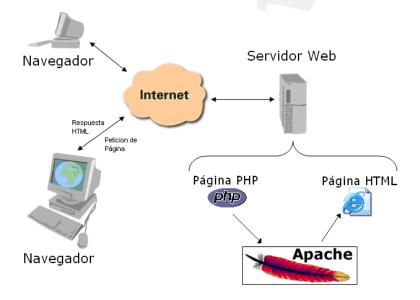


## Servidor Web Apache

El programa PHP es ejecutado en el servidor y el resultado es una pagina HTML. Este resultado HTML es enviado al navegador en forma de respuesta.

Al ser PHP un lenguaje que se ejecuta en el servidor no es necesario que su navegador lo soporte.

Los navegadores solo puede recibir código HTML, JavaScript y CSS (y otras sintaxis como XML, JSON, etc). El servidor donde están alojados debe soportar PHP (Web Server Apache). PHP se distingue de la tecnología JavaScript, la cual se ejecuta en la máquina cliente, y el código PHP es ejecutado en el servidor.



# Por qué PHP

# Ventajas

Open Source y Gratuito. Lo cual significa que el usuario no depende de una compañía específica para arreglar

cosas que no funcionan. Además no estás forzado a pagar actualizaciones anuales para tener una versión que funcione.

PHP, es el lenguaje de programacion Web más utilizado en el mundo. Esta supremacía se nota también en el mercado de alojamiento donde hay mayor ofrecimiento de servicios que para otros lenguajes.

- Sintaxis clara. Es similar a la del lenguaje C.
- Es muy seguro y posee el respaldo de una gran comunidad que encuentra soluciones y mejoras permanentemente.
- Multiplataforma. Se pueden desarrollar aplicaciones en Linux, Windows Mac, Solaris, etc. Este posee una gran portabilidad y disponibilidad.
- Tiene soporte para diversos servidores Web y puede correr tanto en IIS como en Apache.
- Soporte Nativo para todo tipo de bases de datos. MySQL, MSSQL, Informix, DBASE, SYBASE, ADABAS, Oracle, Postgre, SQLite y también ODBC.
- Utiliza el Paradigma Orientado a Objetos POO.
- Posee gran cantidad de funciones, librerías adicionales, paquetes, códigos y hasta desarrollos completos generados y disponibles gratuitamente.
- Es el lenguaje que tiene mayor disponibilidad de plataformas de sistemas operativos y servidores Web.

# Desventajas

- No nació como un lenguaje puramente POO. Es un lenguaje interpretado.
- No es un framework sino un lenguaje que al ser interpretado pierde en calidad y rendimiento comparado con las plataformas compiladas como J2EE y .NET
- PHP es una de las pocas plataformas que no tiene gran respaldo corporativo como pueden ser Microsoft, SUN o Adobe.
- No tiene la posibilidad de ocultar código fuente

#### Propuestas del mercado

Las siguientes son las propuestas mas utilizadas del mercado de sistemas para desarrollo Web:

- PHP
- ASP.NET / MVC (Microcoft .\_NET)
- JSP (Java)
- Ruby
- Python
- NodeJS

# **Proyecto Integrador**

#### Introducción

El proyecto integrador es un desarrollo de un Sitio Web Completo de registración electrónica, administración interna de usuarios, administración de perfiles, sesiones, manejo de visitas, confirmaciones con envío de emails,

El proyecto se desarrollará a lo largo del curso y se irá complementando con los diferentes laboratorios que hagan mención al mismo.

# **Objetivos**

- Desarrollar un proyecto completo de un Sitio Web.
- Manejar el concepto de árbol de un sitio y hojas (páginas)
- Ejercitar a través de un proyecto con fines claros y determinados todos los conceptos teóricos aprendidos a lo largo del curso.
- Dominar la utilización correcta de las diferentes herramientas de programación Web.
- Respetar los estándares de desarrollo y programación Web profesional.
- Aprender a realizar una inscripción Web, tomando todos los recaudos de dobles confirmaciones a través de envió de mails, para no recaer un SPAM.
- Dominar los conceptos de registración, logueo y autentificación de sesiones.
- Adquirir experiencia en el desarrollo y programación de un Sitio Web Completo

# **Links Downloads**

#### Sitios institucionales

http://www.php.net/

http://www.zend.com/

http://www.apache.net/

http://www.mysql.com/

# Instalación

# Entorno de trabajo

Cuando tengamos nuestro primer desarrollo PHP y queramos subirlo a la Web, lo más probable es que necesitemos depender de una infraestructura de alojamiento de terceros, lo que llamamos HOSTING. Sin embargo, como la mayoría de los desarrolladores, primero necesitaremos programar nuestras aplicaciones y probarlas localmente. Osea en una estación de trabajo / entorno local o en un servidor de desarrollo dedicado. Es por eso que tienes que saber por lo menos a instalar y configurar PHP y un servidor web (Apache y MySQL) de manera local.

# Descargar e instalar

Existen varias opciones para instalar localmente y de manera rápida un servidor Web completo (Apache, MySQL, PHP) y todo auto configurado. Entre las variadas opciones se encuentan el XAMPP y el APPSERV. En la sección descargas de esta clase encontrarás unos documentos PDF con la explicación paso a paso de cómo hacer la instalación. ATENCIÓN: debes elegir una de las dos opciones para instalar, no las dos. A continuación detallamos una breve explicación de cada una de ellas:

#### **XAMPP**



XAMPP es un paquete de instalación de un conjunto de servidores independientes de software libre y otras herramientas, que contiene principalmente el servidor de base de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script: PHP y Perl. El nombre proviene del acrónimo de X (para cualquiera de los diferentes sistemas operativos), Apache, MySQL, PHP, Perl.

El programa está liberado bajo la licencia GNU y actúa como un servidor web libre, fácil de usar y capaz de interpretar páginas dinámicas. Actualmente XAMPP está disponible para Microsoft Windows, GNU/Linux, Solaris y MacOS X.

#### **APPSERV**



Appserv es una herramienta OpenSource para Windows con Apache, MySQL, PHP y otras adiciones, en la cual estas aplicaciones se configuran en forma automática, lo que permite ejecutar un servidor web completo. Como extra incorpora phpMyAdmin para el manejo de MySQL

# Comprobación de la instalación

La mejor manera de verificar la instalación de PHP es tratando de ejecutar la siguiente URL: http://localhost/phpinfo.php

Esta pagina llama a un archivo php que contiene el siguiente código:

<?php
phpinfo();</pre>

?>

# PHP: Lenguaje de Programación

# **Sintaxis**

# **Etiquetas de PHP**

Los delimitadores que encierran código PHP y a los cuales nos referíamos en el punto anterior son los siguier
---

- <?php Apertura
- ?> Cierre

En muchos manuales y documentos de PHP encontrarás diferentes especificaciones sobre la sintaxis de los delimitadores pero el primero de todos es el mas formal y que comúnmente se utiliza.

echo " Hola Mundo \n";

</body>

</html>

En este ejemplo se encuentra el esqueleto compuesto por información en HTML y entre este se encuentran comandos que son interpretados y ejecutados por el "Servidor de Páginas (Apache)". Los comandos ejecutables se encuentran entre <?php y ?>

La sentencia echo (que se puede observar en el ejemplo anterior) se encarga de imprimir la cadena de caracteres que recibe como parámetro entre comillas dobles "". Obsérvese también que se debe colocar después de "Hola a todos \n" el "\n" final. Es para que se formatee la salida y agregue un carácter de nueva línea para que se pueda leer más fácilmente el código.

Todos los finales de línea de PHP llevan la terminación ";". Como en C o en Perl, PHP requiere que las instrucciones terminen en punto y coma al final de cada sentencia. La etiqueta de cierre de un bloque de código de PHP automáticamente implica un punto y coma; no es necesario usar un punto y coma para cerrar la última línea de un bloque de PHP. La etiqueta de cierre del bloque incluirá la nueva línea final inmediata si está presente.

# Comentarios en PHP

Un comentario es un texto adicional que se añade al código para explicar su funcionalidad, bien a otras personas que lean el programa, o al propio autor como recordatorio. Los comentarios son una parte importante de la documentación de un programa. Estos son ignorados por el compilador, por lo que no incrementan el tamaño del archivo ejecutable; se pueden, por tanto, añadir libremente al código para que pueda entenderse mejor.

Los comentarios en PHP se pueden escribir de dos formas:

#### Línea comentada

// Línea comentada

Habitualmente, usaremos comentarios en una sola línea //, ya que no tiene el inconveniente de aprendernos los símbolos de comienzo y terminación del bloque, u olvidarnos de poner este último, dando lugar a un error en el momento de la compilación. En los editores especializados para PHP, los comentarios se distinguen del resto del código por el color del texto, generalmente en verde.

# Párrafo comentado

/\*

Esto es un ejemplo de párrafo comentado.

Como podemos observar, un comentario en varias líneas es un bloque de texto situado entre el símbolo de comienzo del bloque /\*, y otro de terminación del mismo \*/

Nótese que PHP soporta el mismo estilo de comentarios que 'C' y 'C++'

# PHP Embebido

#### Definición

Se dice que PHP está como en la casa de HTML, de hecho, por lo general incrustado dentro del HTML.

¿Cómo reconoce el intérprete de PHP cuál es el código PHP dentro de su documento HTML?

La respuesta es que el programa cuando entra en acción mediante el uso de etiquetas especiales de PHP al principio y al final de cada sección de PHP. Este proceso se llama saliendo de HTML o intento de fuga en PHP.

<?php .. ?> Todo dentro de estas etiquetas es entendido por el intérprete como código PHP. Todo fuera de estas no se refiere al servidor y simplemente se pasará al cliente como HTML, JavaScript, CSS o algo más que sea soportado por la W3C y por lo tanto entendido por el navegador.

# **Significado**

PHP es embebido dentro de HTML.

Esto significa que las páginas PHP son originalmente páginas HTML que contienen bloques de código PHP solo cuando es especificado.

```
<HTML>
<HEADER>
<TITLE> Ejemplo </TITLE>
</HEADER>
<BODY>
<P>Hello
<?php
echo "Word";
```

?>

</P>

</HTML>

Cuando un cliente solicita esta página (alojada en el servidor), el Web Server la preprocesa.

Esto significa que el Web Server recorre la página desde el inicio hasta el final, buscando las secciones (bloques) de PHP para luego resolverlas (interpretarlas).

#### **Consideraciones**

El código PHP puede ser rápidamente agregado por cualquier editor de texto a una página ya formateada en HTML.

PHP presta la posibilidad de dividir perfectamente la labor entre diseñadores y programadores Web.

Cada línea escrita en HTML no tiene la necesidad de ser rescrita en un lenguaje de programación.

Embeber código PHP permite reducir los costos en términos de labor e incrementar la eficacia y el mantenimiento de las páginas. Sobre todo en proyectos donde trabajan de manera mancomunada Programadores y Diseñadores Web.

#### HTML no es PHP

Es importante comprender que la sintaxis de PHP es sólo relevante dentro de las máscaras de PHP. Dado que PHP está incrustado en HTML, cada parte del documento se interpreta como PHP o HTML, dependiendo de si esa sección del documento es entre etiquetas de PHP o no.

#### Comentarios en HTML

Los comentarios en HTML se realizan de la siguiente manera: Se abren <!— y se cierran  $\rightarrow$ 

Poner comentarios como HTML, tiene como limitante que la información comentada en la página, viaja de todas maneras a la máquina del cliente una vez que el mismo la solicita. Esta información al estar comentada (en el código HTML) no se muestra en el navegador, pero sí está presente y el usuario podría consultarla al seleccionar la opción de ver el código fuente.

Esto se llama Comentar del lado del Cliente.



A partir de ahora, lo ideal es comentar los códigos utilizando los comentarios de PHP. Esto se llama Comentar del Lado del Servidor.

# **Variables**

#### Introducción

Una variable es un nombre que se asocia con una porción de la memoria del ordenador, en la que se guarda un valor asignado a dicha variable. Consiste en un elemento al cual le damos un nombre y le atribuimos determinado tipo de información. Las variables pueden ser consideradas como la base de la programación.

Los datos que se manejan en nuestro programa se almacenan en variables. El concepto de variable debe verse como un contenedor de información, la cual puede ir variando, por eso se llama variable. Todas las variables en PHP se denotan con un signo de pesos (\$) como prefijo del nombre.

a=10;

\$b=5;

La variable que nosotros llamamos "\$a" posee un elemento de información de tipo numero entero que es 10. Asimismo, la variable "\$b" contiene el valor 5. Podríamos definir una tercera variable que fuese la suma de estas dos:

c=a+b;

Si introdujéramos una petición de impresión de esta variable en nuestro lenguaje ficticio:

echo \$c

El resultado podría ser:

#### Definición de Variables

- Las variables se representan con un signo de pesos \$ seguido por el nombre de la variable.
- Las variables en PHP no se definen y dependen del tipo de datos que contienen.
- El nombre de la variable es sensible a minúsculas y mayúsculas.
- A una variable se le puede cambiar su definición con solo re-asignarle un valor con el tipo de datos al cual se desea cambiar.
- La principal forma de almacenar información en el medio de un programa en PHP es mediante el uso de una variable. El nombre es la manera de aferrarse a cualquier valor que desea utilizar más tarde.
- Las variables se asignan con el operador =, con la variable del lado izquierdo y la expresión a ser evaluada a la derecha. \$variable = valor;

## **Tipos de Datos**

#### **Escalares**

- boolean
- integer
- float (double)
- string

#### **Compuestos**

- array
- object

#### **Especiales**

- resource
- (referencias a archivos abiertos, conexiones a bases de datos, etc.)
- NULL

# Asignación de Variables

La asignación de variables es simple. Sólo escribir el nombre de variable y añadir un signo de igual simple (=). A continuación, agregar la expresión que desea asignar a esa variable:

$$pi = 3 + 0.14159;$$

## Ejemplo de Variables

- 1. \$variable = 123; // La variable es de tipo integer.
- 2. \$variable = "123"; // La variable es de tipo texto.

3. \$variable = 1.23; // La variable es de tipo double.

## Ejemplo de Definición de Variables

```
<?php
                            $titulo = "Definición de Variables";
                            $encabezado = "Ejemplo de Definición de Variables";
?>
<html>
                            <head>
                                 <title> <?php echo $titulo; ?> </title>
                            </head>
                            <body>
                                    <h1 align="center"> <?php echo $encabezado; ?> </h1>
                                    \langle BR \rangle
                                    <?php
                                            $variable = "123";
                                            echo "El valor es: ". $variable . "<BR> \n" ;
                            </body>
</html>
```

La sentencia echo (que se puede observar en el ejemplo anterior) se encarga de imprimir la cadena de caracteres que se encuentra en la variable "\$variable". Obsérvese también que se utiliza el "." (punto) para concatenar las cadenas (tanto las que se encuentran entre comillas dobles como las variables).

#### Redefinición de Variables

No hay una distinción interesante en PHP entre la asignación de una variable para la primera vez y cambiar su valor más adelante. Esto es cierto incluso si los valores asignados son de diferentes tipos. Por ejemplo, el siguiente es perfectamente legal:

```
$my_num_var = "Esto debería ser un numero";
$my_num_var = 5;
```

# Ejemplo de Redefinición de Variables

```
<?php
$titulo = 'Redefinicion de Variables';
$encabezado = 'Ejemplo de Redefinicion de Variables';
?>
<html>
                          <head>
                            <title> <?php echo $titulo; ?> </title>
                          </head>
                          <body>
                                 <h1> <?php echo $encabezado; ?> </h1>
                                 <?php
                                 $variable = '123';
                                 echo 'El valor es: '. $variable . '';
                                 variable = 1.23;
                                 echo 'El nuevo valor es: '. $variable . '';
                                 ?>
                          </body>
</html>
```

# **Operadores**

#### Definición

Un operador es una expresión que produce otro valor (así como las funciones o construcciones que devuelven un valor).

Existen operadores de comparación, de negación o de incremento y decremento.

Las operaciones matemáticas se comportan de igual manera en PHP. Las operaciones \* y / tienen precedencia sobre la suma y la resta y se pueden utilizar paréntesis para expresiones más complejas.

Un operador es un símbolo especial que indica al compilador que debe efectuar una operación matemática o lógica.

En todo lenguaje de programación existe un conjunto de operadores que permiten realizar operaciones con los datos. Nos referimos a operaciones aritméticas, comparaciones, operaciones lógicas y otras operaciones (por ejemplo, concatenación de cadenas, operaciones algebraicas entre valores números, etc.).

## **Operadores de Incremento y Decremento**

Al igual que C, PHP ofrece economía de expresiones en lo que concierne a las operaciones de incremento y decremento.

```
$valor++; //incrementa el contenido numérico de la variable valor en 1
$valor--; //decremento el contenido numérico de la variable valor en 1
$valor+=2; //incrementa el contenido numérico de la variable valor en 2
$valor+=2; //incrementa el contenido numérico de la variable valor en 2
$valor-=2; //decremento el contenido numérico de la variable valor en 2
```

#### Ejemplo Completo de Incremento y decremento

```
<?php
$titulo = "Incrementos y Decrementos";
$encabezado = "Ejemplo de Incrementos y Decrementos";
?>
<html>
```

```
<head>
<title> <?php echo $titulo; ?> </title>
</head>
<body>
<h1 align="center"> <?php echo $encabezado; ?> </h1>
<BR>
<?php
num1 = 10;
num2 = 15;
$x=6;
echo "La variable 'x' tiene el siguiente valor: " . $x . "<br>\n" ;
$x++;
echo "La variable 'x' tiene el siguiente valor incrementado en 1: " . $x . "<br>\n" ;
$x--;
echo "La variable 'x' tiene el siguiente valor decrementado en 1: " . $x . "<br>\n" ;
x=8;
echo "La variable 'x' tiene el siguiente valor incrementado en 8: " . $x . "<br>\n" ;
$x-=4;
echo "La variable 'x' tiene el siguiente valor decrementado en 4: " . x . "< br> \n";
?>
</body>
</html>
```

Ejemplo	Descripcion	Resultado
++\$a, \$a++	Incrementa	Incrementa \$a en 1
\$a, \$a	Decrementa	\$a, \$a Decrementa \$a en 1

Estos operadores se pueden utilizar de ambos lados de la variable. Del lado que se aplique dependerá de lo siguiente:

\$inv = 15; // Asigna el valor entero 15 en \$inv.

\$oldInv = \$inv--; // Asigna a la variable \$oldInv el valor de \$inv, y luego decrementa \$inv

\$origInv = ++\$inv; // Incrementa \$inv, luego asigna el nuevo valor de \$inv a \$origInv.

# Operadores de Asignación

Operador	Equivale a	Descripción
x = y	x = y	Asigna el valor de \$y a \$x
x += y	x = x + y	Suma \$x a \$y y lo guarda en \$x
\$x -= \$y	x = x - y	Resta \$x a \$y y lo guarda en \$x
\$x *= \$y	x = x * y	Multiplica \$x a \$y y lo guarda en \$x
\$x /= \$y	x = x / y	Divide \$x a \$y y lo guarda en \$x
x := y	$x = x \cdot y$	Concatena las cadenas de caracteres de \$x e \$y
		y lo guarda en \$x

# Paso de Variables por URL

#### Protocolo HTTP

HTTP (Hypertext Transfer Protocol) o (en español protocolo de transferencia de hipertexto) es el protocolo usado para la transacción de hipertexto osea paginas Web.

HTTP es un protocolo sin estado, es decir, que no guarda ninguna información sobre conexiones anteriores. El desarrollo de aplicaciones web necesita frecuentemente mantener un estado. Para esto se usan las cookies, que es información que un servidor puede almacenar en el sistema cliente (este punto lo veremos mas adelante).

#### Petición HTTP

Cada petición (HTTP Request), es, en la mayoría de los casos, una petición de un recurso (HTML, jpg/jpeg/gif/png, CSS o js) que se solicita y es entregado al cliente. El servidor no sabe nada de fondo acerca de la identidad del cliente y no tiene memoria.

Cada solicitud genera un discreto proceso que lleva a cabo su tarea de servir a un archivo solicitado. Incluso si el diseño de su sitio tiene una navegación muy estricta (o sea, que la página 1 sólo conduce a la página 2 y que ésta última sólo conduce a la página 3, y así sucesivamente). El protocolo HTTP nunca se sabe ni le importa que alguien viendo la página 2 debe haber venido de la página 1. No se puede establecer el valor de una variable en la página 1

y esperar que sea importado a la página 2 por las exigencias de HTML.

Aquí es donde una forma de manipulación de la tecnología PHP entra en la captura de variables de una página a la siguiente y ponerla a disposición para su uso posterior. La mayoría de las técnicas básicas para pasar datos (variables) entre las páginas Web, utilizan los Métodos GET y POST en HTTP.

## Parámetros en la página (Método GET)

Cuando solicitamos un recurso al servidor, osea una pagina, se pueden pasar parámetros en la llamada a la pagina. Las variables son enviadas junto con la dirección de la pagina web, formando parate de la URL. Se realiza de la siguiente manera:

pagina.php?variable1=valor1&variable2=valor2&variableN=valorN

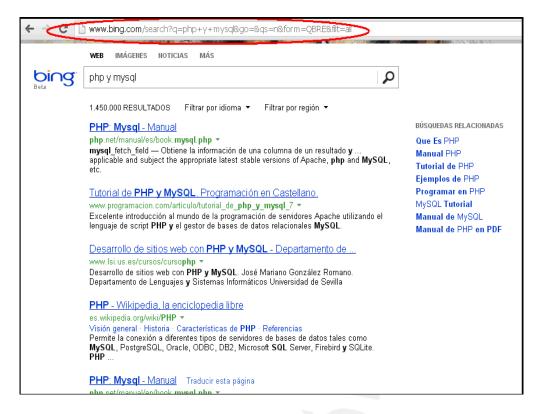
Enviar parámetros a una pagina como se explico en el punto anterior se lo llama Método GET.

El método GET pasa los argumentos de una página a la siguiente parte como de la URL (uniform resource location, localización uniforme de recursos). Cuando se utiliza GET, se añade a la URL el nombre de la variable y el valor. Separando la pagina del nombre de la variable con un signo de interrogación (?). Por ejemplo: http://www.noticias\_online.php?codigoArticulo=121433

Si se necesita enviar mas de una variable en la URL, los siguientes parámetros no se separa con el signo de interrogación (?) sino que se separan con el signo ampersan (&). Por ejemplo:

http://www.noticias\_online.php?codigo=121433&fecha=2015.01.01&color=rojo

En el siguiente ejemplo se ve como Bing utiliza parámetros para especificar la búsqueda que solicito el usuario.



## Argumento \$\_GET

Lo que sigue ahora es poder obtener con PHP los valores por parámetros que son enviados desde la URL. Para eso utilizamos el Argumento \$\_GET

```
<ppe cho $_GET['fecha']; ?>
</body>
</html>
```

Al introducir en el navegador una URL similar a

```
clase1/parametros-por-GET.php?color=red&codigo=20&fecha=2015
```

PHP utiliza las variables obtenidas por GET a través de \$\_GET y genera el siguiente HTML

```
<html>
<head>
<title>Parametros por metodo GET</title>
</head>
<body>

<h1>Ejemplo de Parametros por metodo GET</h1>
20
2015 
</body>
</html>
```

# Resumen

Los delimitadores <?php nos indican que comienza un script en PHP, y colocamos ?> para indicar el final del

script. Hay que destacar que todas las líneas que se encuentren entre estos delimitadores deben acabar en punto y coma, excepto las sentencias de control (if, switch, while, etc, que veremos en las próximas clases).

Es importante utilizar los comentarios para otorgar más claridad al código, para lo cual si queremos comentar una sola línea tenemos que poner al principio de la línea //, si lo que queremos es comentar varias utilizaremos los delimitadores /\* - \*/.

Todas las variables en PHP, deben ser precedidas por signo pesos (\$). Para la asignación de contenido en una variable se utiliza el signo igual (=). PHP es sensitivo a la hora de nombrar las variables por lo que distingue entre mayúsculas y minúsculas.

Si el cliente adjunta variables por GET al pedido del recurso (utilizando la sintaxis recurso.php?nombre=valor) puedo recuperar esos valores a través del argumento \$\_GET['nombre'].

