

INTRODUCCIÓN

- PHP es un lenguaje que se ejecuta en el lado del servidor
- PHP se utiliza para generar páginas web dinámicas
- Páginas creadas con php



facebook

SINTAXIS

- delimitadores más comunes son `<?php` para abrir una sección PHP y `?>` para cerrarla.

```
<html>
  <head>
    <title> Ejemplo básico PHP</title>
  </head>
  <body>
    <?php
      echo 'Hola mundo';
    ?>
  </body>
</html>
```

VARIABLES

- se prefijan con el símbolo del dólar (\$)

Veamos algunos ejemplos:

Posible nombre de variable	Validez
\$4variable	Nombre de variable inválido.
\$_4variable	Nombre de variable válido.
\$variable4	Nombre de variable válido.
\$otra	Nombre de variable válido.
\$1_otra	Nombre de variable inválido.
\$variable_de_nombre_muy_largo	Nombre de variable válido.
\$ABC	Nombre de variable válido.
\$ A B C	Nombre de variable inválido.
\$A_y_B_x_C	Nombre de variable válido.

En cuanto a las palabras clave

- condiciones con *if*
- los bucles con *for* y *while*
- los retornos de funciones.
- las sentencias deben acabar con punto y coma (;)

OTRAS SINTAXIS

- La instrucción echo
- Comillas

Ejemplo:

```
<?php  
echo "Hola mundo";  
?>
```

TIPOS DE DATOS

- los tipos de datos son establecidos directamente por el intérprete PHP

Ahora veremos en una tabla los distintos tipos de datos que se pueden usar en PHP.

TIPO DE DATO	DEFINICIÓN
integer	<p>Los integers, o enteros, pueden tener distintos valores numéricos enteros que se expresan con diferentes notaciones.</p> <p><code>\$variable = 18; // Número entero positivo</code></p> <p><code>\$variable = -18; // Número entero negativo</code></p>
float o double	<p>Este tipo de datos son los números de punto flotante a los que normalmente llamamos “números decimales”, por ejemplo, 9.876. Ambos tienen mucha precisión, pero double es el más preciso (con más decimales). La sintaxis para utilizarlos es bastante simple:</p> <p><code>\$variable = 9.876;</code></p>
string	<p>El tipo de datos string, también conocido como cadena de caracteres, se expresa con la siguiente sintaxis:</p> <p><code>\$variable = "Yo soy una cadena";</code></p>

OPERADORES ARITMÉTICOS

En PHP disponemos de los operadores habituales

Nombre	Ejemplo	Resultado	Ejemplo (con \$a = 8 y \$b = 4)
Suma	$\$a + \b	El resultado de la suma.	12
Resta	$\$a - \b	El resultado de la resta.	4
Multiplicación	$\$a * \b	El resultado de la multiplicación.	32
División	$\$a / \b	El resultado de la división.	2
Resto o módulo	$\$a \% \b	El resto de la división de \$a entre \$b (1)	0

Gestión de memoria

- Una variable en PHP se almacena en un contenedor llamado “zval”
- "is_ref" : indica si la variable es parte o no de un "conjunto de referencias“
- "refcount", contiene el número de variables que apuntan a este contenedor zval.

Recolector de basura

```
<?php
class Foo
{
    public $var = '3.14159265359';
}

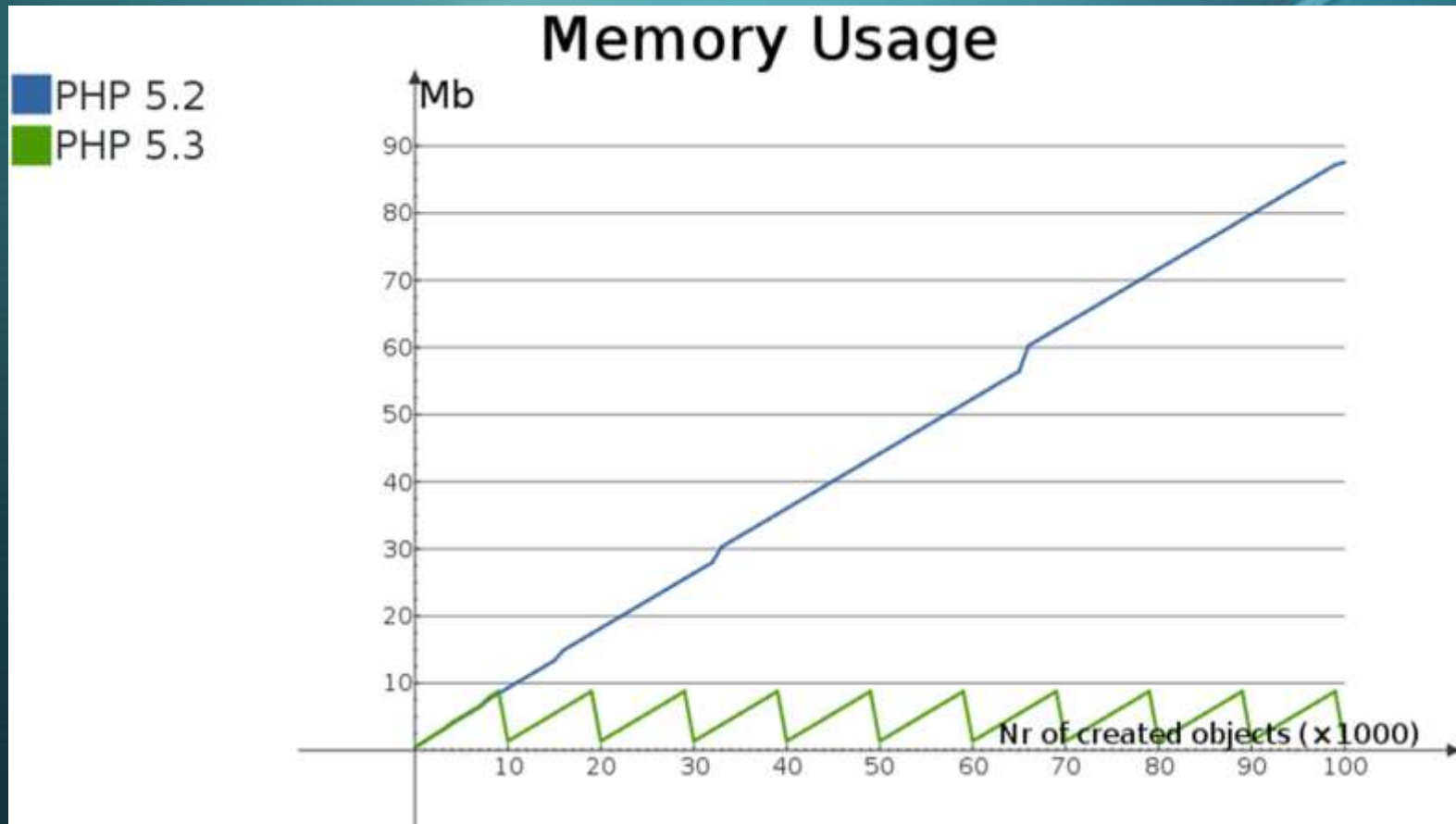
$baseMemory = memory_get_usage();

for ( $i = 0; $i <= 100000; $i++ )
{
    $a = new Foo;
    $a->self = $a;
    if ( $i % 500 === 0 )
    {
        echo sprintf( '%8d: ', $i ), memory_get_usage() - $baseMemory, "\n";
    }
}
?>
```

PROCESO

- En este ejemplo didáctico, estamos creando un objeto en el que una propiedad enlaza de nuevo al propio objeto.
- Cuando la variable $\$a$ del script se reasigna en la siguiente iteración del bucle, típicamente ocurriría una fuga de memoria. En este caso, se fugan dos contenedores zval (el zval del objeto, y el zval de la propiedad), pero sólo se encuentra una posible raíz: la variable que se desasignó.
- Tras 10.000 iteraciones, el buffer se llena (con un total de 10.000 posibles raíces), y se lanza el mecanismo recolector de basura y libera la memoria asociada con esas posibles raíces

GRÁFICO



Manipulación de excepciones

- modelo de excepciones similar al de otros lenguajes
- un bloque [try](#) para facilita la captura de excepciones potenciales
- Cada bloque [try](#) debe tener al menos un bloque [catch](#) o [finally](#) correspondiente.

Ejemplo Lanzar una Excepción

- ```
<?php
function inverso($x) {
 if (!$x) {
 throw new Exception('División por cero.');
```

```
 }
 return 1/$x;
}

try {
 echo inverso(5) . "\n";
 echo inverso(0) . "\n";
}
catch (Exception $e) {
 echo 'Excepción capturada: ', $e->getMessage(), "\n";
}
```

```
// Continuar la ejecución
echo 'Hola Mundo\n';
?>
```

El resultado del ejemplo sería:

0.2

Excepción capturada: División por cero.

Hola Mundo

El concepto de paradigma: se utiliza en la vida cotidiana como sinónimo de **“ejemplo”** o para hacer referencia en caso de algo que se toma como **“modelo digno de seguir”**.

Un **paradigma de programación** es una propuesta tecnológica adoptada por una comunidad de **programadores** y **desarrolladores** cuyo núcleo central es incuestionable en cuanto que únicamente trata de resolver uno o varios problemas claramente delimitados



## SUS CARACTERÍSTICAS

- representan un enfoque particular o filosofía para diseñar soluciones
- Los paradigmas difieren unos de otros, en los conceptos así como en los pasos que integran su solución del problema
- Tiene una estrecha relación con la formalización de determinados lenguajes en su momento de definición
- Es un estilo de programación empleado.



## PRINCIPALES PARADIGMAS

- **La programación imperativa:** el programa es una serie de pasos los cuales realizan cálculos y se obtiene como resultado una salida. Ejemplo: Cobol, fortran y C++
- **Programación orientada objeto:** es un conjunto de objetos que interactúan entre sí trasladándose mensajes que transforman su estado. Ejemplo: Java y C++

- **Programación lógica:** Es un conjunto de declaraciones lógicas sobre el resultado que debería conseguir una función. Ejemplo: el más importante es el Prolog
- **Programación guiada por eventos:** el programa es un bucle continuo que responde a los eventos generados en un orden no predecible. Ejemplo: Visual Basic y Java.
- **Programación concurrente:** El programa es una colección de procesos cooperativos que comparten información pero operan de forma asíncrona. Ejemplo: SR, Linda y Fortran de alto rendimiento

## PARADIGMAS DEL LENGUAJE PHP:

- ES UN LENGUAJE MULTIPARADIGMATICO
- **PHP:** Es imperativo, orientado a objetos y funcional
- **Php es Imperativo:** En la programación imperativa de php se **describen sentencias que modifican el estado de un programa.**
- En php este paradigma expresa **como debe solucionarse** un problema especificando una **secuencia de acciones** a realizar a través de uno o más procedimientos denominados subrutinas o funciones

- **Php orientado a objeto:** PHP ofrece funcionalidades propias de la POO. vamos a tratar de explicar la sintaxis básica de PHP para utilizar objetos, sin meternos en mucha teoría de programación orientada a objetos en general.

En php se utilizan las clases para manejar el tema de la programación orientada a objetos.

- **Las clases:** Una clase es un conjunto de variables, llamados atributos, y funciones, llamadas métodos, que trabajan sobre esas variables.

- Las clases en PHP se definen de la siguiente manera:

```
<?
class Caja{
 var $alto;
 var $ancho;
 var $largo;
 var $contenido;
 var $color;

 function introduce($cosa){
 $this->contenido = $cosa;
 }

 function muestra_contenido(){
 echo $this->contenido;
 }
}
?>
```

## DOMINIOS DE APLICACIÓN

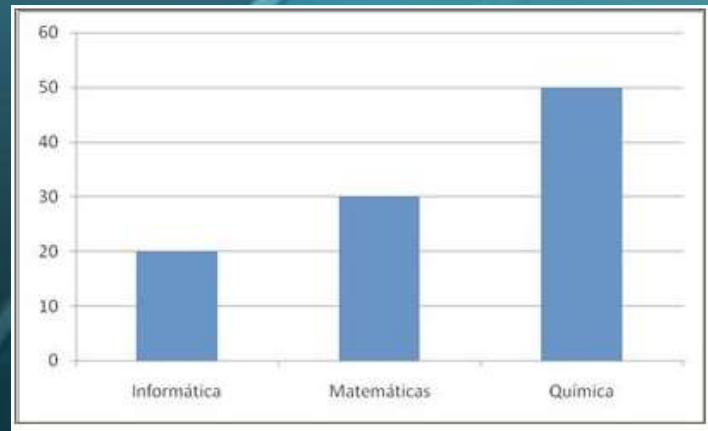
- Computación científica: se encarga de la realización de cálculos complejos de forma muy rápida y precisa para representar fenómenos científicos.
- Sistemas de gestión de información: diseñados para que las instituciones los utilicen para administrar sus altos contenidos de información. (contabilidad, ventas e inventario)
- Centrada en la Web: Nos permite el comercio electrónico y una gran cantidad de aplicaciones académicas, gubernamentales e industriales.



# PHP EN LA CIENCIA

Entre las habilidades de PHP se incluye la creación de imágenes a partir de datos.

Ejemplo: supongamos que tenemos un colegio con 20 alumnos matriculados en un curso de informática, 50 alumnos matriculados en un curso de matemáticas y 30 alumnos matriculados en un curso de química. Con PHP podemos generar automáticamente imágenes similares a esta:





# PHP EN SISTEMAS DE GESTIÓN DE INFORMACIÓN

Una de las principales ventajas que presenta el trabajar con php es el poder trabajar con contenidos que están en bases de datos. De esta forma, podemos organizarlos, actualizarlos y buscarlos de una manera mucho más simple.

Las Bases de Datos tienen una gran relevancia a nivel personal, pero más a nivel empresarial, y se consideran una de las mayores aportaciones que ha dado la informática a las empresas.

Las principales utilidades que ofrece php a las empresas son las siguientes:

- **Agrupar y almacenar todos los datos de la empresa en un único lugar.**
- **Facilitar que se compartan los datos entre los diferentes miembros de la empresa.**
- **Evitar la redundancia y mejorar la organización de la agenda.**



GRACIAS PHP!!

# PHP EN LA PROGRAMACIÓN CENTRADA EN LA WEB

**El mundo del desarrollo web y del diseño web se ha venido expandiendo en esta última época.**

PHP es un lenguaje de código abierto muy popular adecuado para desarrollo web

Es popular porque un gran número de páginas y portales web están creadas con PHP.

Código abierto significa que es de uso libre y gratuito para todos los programadores que quieran usarlo



Php cuenta con muchas herramientas que permiten un mejor desarrollo de la programación centrada en la web.

A continuación mencionaremos algunas de estas herramientas:

**React:**

Con esta librería en PHP puedes crear un servidor web así como lo hace NODE.js para tus desarrollos web.



# Opauth

Esta librería te facilita la tarea de la autenticación de usuarios en tu aplicación web.



## Laravel:

Framework de PHP que te facilita la creación de aplicaciones web.



## PHP: La manera correcta

Este sitio te informa de las mejores prácticas que debes seguir al desarrollar en P.



## Ratchet

Librería en PHP que te permite utilizar WebSockets para crear aplicaciones que respondan en tiempo real.





# Authmanager:

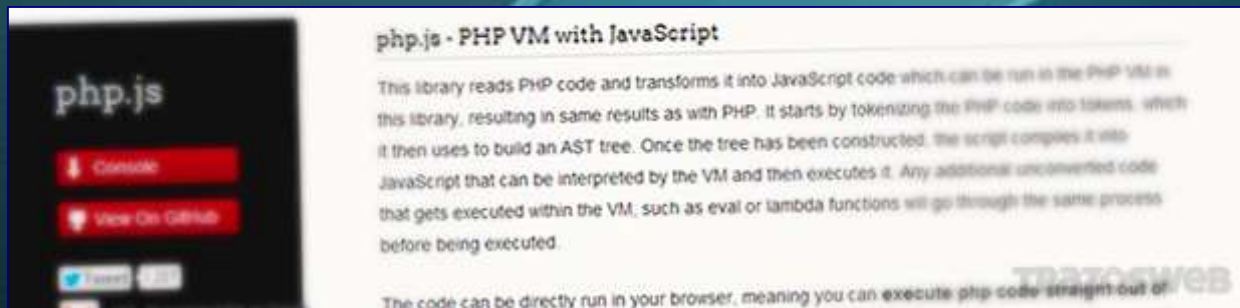
Con esta librería puedes administrar los usuarios y su autenticación en tus desarrollos web



Y por último

## PHP.js:

Librería que se encarga de transformar el código PHP en código JavaScript para ser ejecutado en tus proyectos.





# Simplicidad y Claridad

¿Es fácil para un programador escribir un programa en php?

Escribir un programa en php es muy sencillo debido a su sintaxis sin complejidad

No requiere definición de tipos de variables

Modelo Vista/Controlador (MVC)

podemos tener de manera independiente el código que se encarga de mover los datos del que se enlaza a la interfaz

## Hola mundo en 3 líneas

hola\_mundo.php

```
<?php
echo '¡Hola Mundo!';
?>
```

- El código php se escribe entre etiquetas `<?php` y `?>`
- Para escribir un mensaje en la página, podemos usar `echo '¡Hola Mundo!';`
- Las cadenas se escriben entre apóstrofes `"`, ó entre comillas `"`



# ¿hasta que punto es inteligible este programa para un nivel medio?

## Ejemplo

```
html>
<head>
<title>Ejemplo de PHP</title>
</head>
<body>
 <?php
 $a = 8;
 $b = 3;
 echo $a + $b,"
";
 echo $a - $b,"
";
 echo $a * $b,"
";
 echo $a / $b,"
";
 $a++;
 echo $a,"
";
 $b--;
 echo $b,"
";
 ?>
</body>
</html>
```

Php es sumamente comprensible, inclusive a un punto intermedio.

Logicamente a medida que el nivel de complejidad aumente , inversamente proporcional será la facilidad de entenderlo,

aún así seguirá siendo bastante sencillo de comprender cualquier código



# ¿Es fácil aprender y enseñar este lenguaje?

Aprender y enseñar php es muy sencillo.  
a nivel básico es sumamente fácil el programar

y estando en un nivel intermedio se pueden  
crear Grandes aplicaciones (sitios web)  
con suma sencillez

Además que las sintaxis básicas  
para el manejo de Base de datos son  
Fáciles de comprender y explicar



Ejemplo de sintaxis para insertar un registro en una base de datos:

```
$q="INSERT INTO tabla1 (clave,nombre,edad,estatura) VALUES (".$CLAVE.", " ".$NOMBRE.", " ".$EDAD.", " ".$ESTATURA.)";
```

# Uniones



Las cadenas, como ya sabemos, son un tipo de dato; concretamente una sucesión de caracteres; y que pueden ser número, letras o símbolos.

En PHP toda cadena debe ir entre comillas; pudiendo ser comillas simples o comillas dobles según la necesidad lo requiera.

Veremos como operar con ellas de forma básica; y como trabajar con ellas junto con las variables.





La primera operación con cadenas que podemos realizar es la de **INCORPORAR VARIABLES**. De esta forma, podemos hacer que aparezca el valor de la variable. Aquí un código de ejemplo:

```
<?php
// variable
$numero = 8;
// mostramos el valor de la variable en la cadena
echo "el número es $numero";
?>
```

El resultado que mostrará el navegador será **el número es 8**; ya que 8 es el valor de la variable que hemos escrito dentro de la cadena. Ahora,

si sustituyes las dobles comillas por comillas simples, el resultado cambia a **el número es \$numero**.

Cuando escribimos cadenas con comillas simples PHP interpreta que queremos escribir de forma literal, por eso no muestra el valor de la variable.



## La concatenación

Esta técnica sirve para unir dos o más datos; por ejemplo dos cadenas, o una cadena y una variable.

Para concatenar tenemos que hacer uso de un operador, este operador se escribe poniendo **un punto**. Veamos un ejemplo donde concatenamos dos cadenas a través de variables:

```
<?php
$cadena1 = 'esto es una cadena';
$cadena2 = ' y esto una segunda cadena';
// concatenamos
echo $cadena1 . $cadena2;
?>
```



El resultado que debe aparecer en el navegador es: **esto es una cadena y esto una segunda cadena**. Como ves en la sentencia echo, hemos juntado las dos cadenas a través del punto, que es el operador de la concatenación.

También hay que darse cuenta que se ha dejado **un espacio** al principio de la segunda cadena para que, al aparecer las dos concatenadas, se respeten todos los espacios entre palabras.

# Ortogonalidad

Se refiere a la capacidad de combinar varias características de un **lenguaje de programación** en todas las combinaciones posibles, de manera que todas ellas tengan significado.

¿Un simbolo o una palabra reservada tienen siempre el mismo significado Independientemente el contexto en que la usemos?

En php las palabras resevadas representan cosas que se parecen a funciones, otras se parecen a constantes, etc.; pero no lo son en realidad: son constructores del lenguaje. No se puede usar ninguna de las

palabras reservadas como constantes, nombres de clases, nombres de funciones o métodos. Se pueden usar como nombres de variables, pero podría dar lugar a confusiones.

En php La no generalidad de la comparación de igualdad puede ser vista como una no ortogonalidad ya que la aplicación de “=” depende del tipo de los valores a comparar.



# Fiabilidad de los programas

¿El programa se comporta igual cada vez que lo ejecutamos con los mismos datos de entrada?

Mientras se inserten los mismos datos de entrada y en el mismo orden, y no exista ningún tipo de error en el código, cualquier programa en php

se comporta igual, mientras no se realice ningún cambio en el código

¿se comporta igual en plataformas distintas?

Gracias a que php es un lenguaje multiplataforma,

no existen diferencias al Momento de ser ejecutado en

cualquiera de las plataformas disponible



# Aplicabilidad

¿El lenguaje proporciona un soporte adecuado para las aplicaciones del dominio en el que se esta utilizando?

Php proporciona un gran soporte para todos los dominios en el que se utiliza, Principalmente en la gestión de información y la orientación a web. Php es de gran

popularidad debido a que grandes sitios web como facebook y wikipedia estan hechos en php.

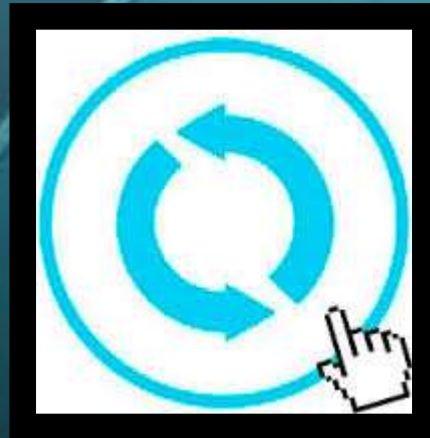
Además php permite la conexión a las bases de datos como pueden ser Mysql, oracle, etc.



# Abstracción

¿Cuántas veces hemos reinventado un algoritmo de ordenación  
O estructura de datos enlazados que probablemente haya sido utilizado  
Miles de veces?

Debido a que php y su programación esta casi o completamente orienta al  
Diseño de aplicaciones y servicios web, cada vez que un programador construye o  
Crea un programa el código o parte de él, por lo general, ya ha sido utilizado dentro  
de otro programa creado, Editado o rediseñado con anterioridad.  
independientemente de la lógica de programación, los códigos suelen ser reciclados  
Y reutilizados miles de veces para futuros proyectos con el fin de  
Optimizar el tiempo de realización



# Implementación eficiente

¿Nos permiten las contrucciones y las caractersticas de un lenguaje  
Su implementación practica y eficiente de las plataformas coetaneas?

Php cuenta con varias características, entre las cuales podemos mencionar principalmente:

Es un lenguaje multiplataforma

Posee una amplia documentación en su página oficial

Es libre, por lo que se presenta como una alternativa de fácil acceso para todos

Estas y otras características permiten una implementación eficiente de las plataformas  
Coetaneas.

Ademas , PHP, como cualquier otro sistema de tamaño considerable,  
está bajo constante escrutinio y mejoramiento. Cada nueva versión incluye  
con frecuencia cambios mayores y menores para mejorar la seguridad y

reparar cualquier fallo, problemas de configuración, y otros asuntos que  
puedan afectar la seguridad y estabilidad global del sistema.





# Recomendación

Como cualquier lenguaje y programa de script a nivel de sistema, la mejor estrategia es actualizar con frecuencia y mantenerse alerta sobre las últimas versiones y sus cambios.



# Infografía

<http://php.net/manual/es/security.current.php>

<http://foros.monografias.com/showthread.php/60249-Caracteristicas-del-PHP>

<http://www.alegsa.com.ar/Dic/palabra%20reservada.php>

<http://www.php-hispano.net/foro/7659/php-estructuraunionesenumeraciones>

# MUCHAS GRACIAS

