

# PHP



Por: Ing. Edwin Calle Terrazas

Versión 1.1

PHP (acronimo de "PHP: Hypertext Preprocessor")

Es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor.

Al nivel más básico con PHP puedes hacer cualquier cosa a través de un script, como: crear páginas dinámicas, registro de información a través de los formularios, mostrar la información de una base de datos, generar páginas con contenidos dinámicos, mandar y recibir cookies, etc.

Quizás la característica más potente y destacable de PHP es su soporte para una gran cantidad de bases de datos. Escribir un interfaz vía web para una base de datos es una tarea simple con PHP

Para ilustrar esto podemos ver un simple ejemplo:

```
<html>
<head> <title> Example </title> </head>
<body>
    <?
        echo "Hola, esto es un Script PHP";
    ?>
</body>
</html>
```

Esto es muy parecido a cualquier otro Script escrito. El código de PHP está incluido en tags especiales "<? ?>".

Lo que hace diferente a PHP es que el código que se deba ejecutar se ejecuta siempre en el servidor.

Así, al ejecutar el script anterior, el cliente recibirá sólo los resultados de la ejecución por lo que es imposible para el cliente acceder al código que generó la página.

## **CARACTERISTICAS DE PHP**

- Más rápido que ASP
- Lenguaje más fácil y potente
- Integración perfecta con 8 servidores HTTP
- Acceso a 20 tipos de Bases de Datos
- Diseño modular de fácil ampliación
- Licencia abierta es gratuito

## **PLATAFORMAS SOPORTADAS (actualidad)**

- UNIX (todas las variantes)
- Windows (NT / W95 / W98 / W2000 / WINXP / WIN7)
- Mac (WebTen)
- OS/2
- BeOS

## **SERVIDORES CONOCIDOS**

- Apache (UNIX, Windows)
- XAMP

# BASE DE DATOS SOPORTADAS

Adabas D

dBase

Empress

FiclePro

informix

InterBase

Solid

Sybase

Velocis

Unix dbm

SQL Server

MySQL

Oracle

PostgreSQL

Maria DB

# OPERADORES RELACIONALES

Operación	Nombre	Resultado
<code>\$a != \$b</code>	No igual	Cierto si el valor de \$a no es igual al de \$b.
<code>\$a !== \$b</code>	No idéntico	Cierto si \$a no es igual a \$b, o si no tienen el mismo tipo.
<code>\$a &lt; \$b</code>	Menor que	Cierto si \$a es estrictamente menor que \$b.
<code>\$a &gt; \$b</code>	Mayor que	Cierto si \$a es estrictamente mayor que \$b.
<code>\$a &lt;= \$b</code>	Menor o igual que	Cierto si \$a es menor o igual que \$b.

# AUTOINCREMENTO Y DECREMENTO

Operación	Nombre	Resultado
<code>++\$a</code>	Pre-incremento	Incrementa \$a en 1, y devuelve \$a (incrementado).
<code>\$a++</code>	Post-incremento	Devuelve \$a, y después lo

## OPERADORES ARITMETICOS

Operación	Nombre	Resultado
$\$a + \$b$	Suma	Suma de $\$a$ y $\$b$ .
$\$a - \$b$	Resta	Diferencia entre $\$a$ y $\$b$ .
$\$a * \$b$	Multiplicación	Producto de $\$a$ y $\$b$ .
$\$a / \$b$	División	Cociente de $\$a$ y $\$b$ .
$\$a \% \$b$	Módulo	Resto de la operación $\$a/\$b$ .

## OPERADORES LOGICOS

Operación	Nombre	Resultado
$\$a \text{ and } \$b$	Y	Cierto si $\$a$ y $\$b$ son ciertos.
$\$a \text{ or } \$b$	O	Cierto si $\$a$ o $\$b$ es cierto.
$\$a \text{ xor } \$b$	O Exclusivo.	Cierto si $\$a$ o $\$b$ es cierto, pero no ambos.
$! \$a$	No	Cierto si $\$a$ es falso.
$\$a \ \&\& \ \$b$	Y	Cierto si $\$a$ y $\$b$ son ciertos.
$\$a \    \ \$b$	O	Cierto si $\$a$ o $\$b$ es cierto.

# ASIGNACIONES CORTAS

`+= -= *= /= %= &= ^= .= >>= y <<=`

```
$var1 += 3;      // $var1 = $var1 + 3;  
$var2 /= 2;      // $var2 = $var2 / 2;  
$var3 >>= 1;     // $var3 = $var3 >> 1;
```



# TIPOS DE DATOS

El tipo de una variable normalmente no lo indica el programador; en su lugar, lo decide PHP en tiempo de ejecución dependiendo del contexto en el que se utilice esa variable.

## Entero

```
$a = 1234; //entero decimal
```

```
$a = -123; //entero negativo
```

## Flotantes

Los números en punto flotante ("double") se pueden especificar utilizando cualquiera de las siguientes sintaxis:

```
$a = 1.234;
```

```
$a = 1.2e3;
```

## Cadena

```
$str = "Esto es una cadena";
```

```
/* Asignando una cadena. */
```

```
$str = $str . " con algo más de texto";
```

```
/* Añadiendo a la cadena. */
```

```
$str .= " Y un carácter de nueva línea al final.\n"; /* Otra forma de añadir, incluye un carácter de nueva línea. */
```

## Booleano

```
$bandera=true;
```

## Fecha

```
$f=date(d -m-y);
```

## FORZADO DE TIPOS

El forzado de tipos en PHP funciona como en C, el nombre del tipo deseado se escribe entre paréntesis antes de la variable a la que se pretende forzar.

`$x = 10;`                      `// $x es un entero`

`$d = (double) $x;`        `// $d es un doble`

Los forzados de tipo permitidos son:

- `(int)`, `(integer)`                      `// fuerza a entero (integer)`
- `(real)`, `(double)`, `(float)`            `// fuerza a doble (double)`
- `(string)`                                  `// fuerza a cadena (string)`
- `(array)`                                   `// fuerza a array (array)`
- `(object)`                                  `// fuerza a objeto (object)`

## FUNCIONES MATEMÁTICAS

<b>pi</b>	Devuelve el valor 3.14159
<b>sin</b> (numero)	Devuelve el seno de un número, en radianes.
<b>cos</b> (numero)	Devuelve el coseno de un número, en radianes.
<b>tan</b> (numero)	Devuelve la tangente de un número, en radianes. Ver también: <b>asin</b> , <b>acos</b> y <b>atan</b> .
<b>abs</b> (numero)	Devuelve del valor absoluto de un número.
<b>bindec</b> (numBinario)	Devuelve el numBinario introducido por parámetro a decimal. Vea también: <b>octdec</b> y <b>hexdec</b> .
<b>decbin</b> (numDecimal)	Devuelve el numero decimal introducido por parámetro a binario. Vea también: <b>dechec</b> y <b>decoct</b>
<b>ceil</b> (numero)	Redondea fracciones hacia arriba. De igual forma abajo <b>floor</b>
<b>pow</b> (numero,numero)	Devuelve base elevado a la potencia del exponente
<b>rand</b> (inicio,fin)	Devuelve un valor aleatorio entre un número inicio y fin.
<b>round</b> (numeroReal)	Devuelve el valor redondeado de un número real.
<b>sqrt</b> (numero)	Devuelve la raiz cuadrada de un numero.
<b>min</b> (num1,num2,...)	Devuelve el valor mínimo de una lista de números. Ver también <b>max</b>
<b>log</b> (numero)	Devuelve el logaritmo de un número. Ver también <b>log10</b>

## FUNCIONES DE FECHA Y HORA

<b>checkdate</b>	Valida una fecha u hora. Devuelve un valor verdadero si la fecha dada es válida; en caso contrario, devuelve un valor falso. Su formato es: <code>checkdate ( int month, int day, int year)</code>
<b>getdate</b>	obtiene información de fecha y hora.
<b>gettimeofday</b>	Obtiene la hora actual. Devuelve un array asociativo que contiene los datos devueltos por esta llamada al sistema.
<b>date</b>	da formato a la fecha/hora local. Su formato es: <b>date</b> ( string format [, int timestamp]). Ejemplo: <code>echo date("l dS of F Y h:i:s A");</code> <code>echo "July 1, 2000 is on a " . date("l", mktime(0,0,0,7,1,2000));</code>
<b>mktime</b>	Se utiliza para realizar cálculos y validaciones con fechas, ya que calcula automáticamente el valor correcto para una entrada fuera de rango. Su formato es: <b>mktime</b> ( int hour, int minute, int second, int month, int day, int year [, int is_dst]). Ejemplo1: <code>echo date( "M-d-Y", mktime(0,0,0,12,32,1997) );</code> Ejemplo2: <code>\$lastday=mktime(0,0,0,3,0,2000);</code> <code>echo strftime("Last day in Feb 2000 is: %d",\$lastday);</code>
<b>gmdate</b>	da formato a una fecha/hora. Ejemplos: <code>echo date( "M d Y H:i:s",mktime(0,0,0,1,1,1998) );</code> <code>echo gmdate( "M d Y H:i:s",mktime(0,0,0,1,1,1998) );</code>
<b>strftime</b>	da formato a la hora o fecha local de acuerdo con las convenciones locales. Su formato es: string <b>strftime</b> ( string format, int timestamp)
<b>time</b>	Devuelve la hora actual como número de segundos transcurridos desde las 00:00:00 del 1 de enero de 1970 GMT.

# Estructuras de control

<?

\$a=2;

\$b=3;

if (\$a > \$b) {

    echo "el mayor es:" . \$a;

} else {

    echo "el mayor es:" . \$b ;

}

?>

<?

\$n=5;

\$i=1;

for (\$i=1; \$i<=\$n; \$i++) {

    echo \$i;

?>

<?

\$n=5;

\$i=1;

while (\$i<=\$n){

    echo \$i;

    \$i++;

}

?>

<?

\$k=5;

\$j=1;

do {

    echo \$j;

    \$j++;

} while ( \$j <= \$k );

?>

<?

```
$i=2;
```

```
switch ($i) {
```

```
    case 1:{
```

```
        echo "es igual a 1";
```

```
    }break;
```

```
    case 2:{
```

```
        echo "es igual a 2";
```

```
    } break;
```

```
    case 3:{
```

```
        echo "es igual a 3";
```

```
    } break;
```

```
    default: break;
```

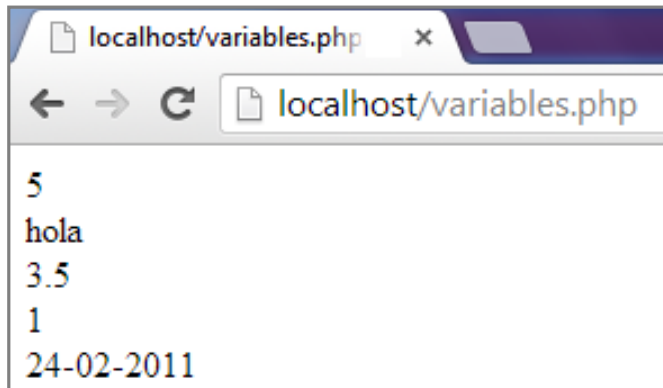
```
}
```

?>

# Ejemplo de variables

```
<?
$entero=5;
$cadena="hola";
$real=3.5;
$booleano=true;
$f=date("d-m-Y");

echo $entero; echo "<BR>";
echo $cadena; echo "<BR>";
echo $real;   echo "<BR>";
echo $booleano; echo "<BR>";
echo $f; echo "<BR>";
?>
```



# Ejemplo de fecha y hora

<?

```
$f=getdate();

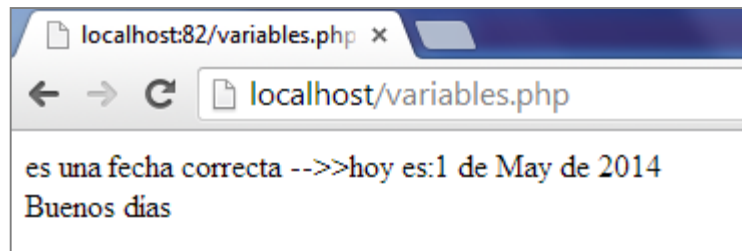
$dia = $f["mday"];
$mes = $f["mon"]; //mes en numero
$mes1 = $f["month"]; //mes en texto, ejemplo January
$anio = $f["year"];
$hora = $f["hours"];
$minuto = $f["minutes"];
$segundos = $f["seconds"];

if (checkdate($mes,$dia,$anio)) //verifica si es una fecha
    echo "es una fecha correcta -->>";
else
    echo "no es una fecha <br>";

echo "hoy es:". $dia ." de ". $mes1 ." de ". $anio; echo "<br>";

if ($hora <= 12)
    echo "Buenos días \n";
else
    echo "Buenas tardes";
```

?>





# Funciones definidas por el usuario

<?

```
function mayor ($a, $b)
{
    if ($a>$b)
        return $a;
    else
        return $b;
};
```

//llamada

```
$n1=5;
$n2=10;
$may=mayor($n1,$n2);
echo $may;
```

?>

<?

```
function factorial($n)
{
    $f=1;
    for($i=1; $i<=$n; $i++)
        $f=$f * $i;
    return $f;
}
```

?>

<?

```
function factorial2($n)
```

```
{
```

```
  $f=1;
```

```
  $i=1;
```

```
  while ($i<=$n)
```

```
  {
```

```
    $f=$f * $i;
```

```
    $i++;
```

```
  }
```

```
  return $f;
```

```
}
```

?>

# MANEJO DE CONTROLES DE FORMULARIOS

```

<html>
<head> <title> Programa potencia </title> </head>
<body>
<form action="ejemploboton.php" method="post">
<input type="text" name="t1" size="8" value=""/> <br>
<input type="text" name="t2" size="8" value=""/> <br>
<input type="submit" name="b1" value="Potencia"/>
</form>

```

```
<?>
```

```

$a=$_POST[t1];      $b=$_POST[t2];
function potencia($x,$y){
    $p=1;  $i=1;
    while($i<=$y){
        $p=$p * $x;
        $i++;
    }
    return $p;
}
if( $_POST[b1] ) {
    echo "La potencia es: " . potencia($a,$b);
}

```

```
?>
```

```

<body>
<html>

```


La potencia es: 16

## USO DE CONTROL SELECT y BUTTON

```
<html>
<head> <title>Programa 1</title> </head>
<body>
<form action="controlselect.php" method="post">
<input type="text" name="t1" size="8" value="" >
<select name="sel1" id="">
  <option> Cubo
  <option> Factorial
</select>
<input type="submit" name="b1" value="Ok">
</form>
```

A screenshot of a web form. It contains a text input field with the number '4' inside. To its right is a dropdown menu with 'Factorial' selected and a small downward arrow. Further right is a blue button with the text 'Ok'.

El factorial es: 24

<?

```
$a=$_POST[t1];
```

```
function cubo($x){
  $r=$x*$x*$x;
  return $r;
}
```

```
function fact($n){  
  $f=1;  
  for($i=1;$i<=$n;$i++)  
    $f=$f*$i;  
  return $f;  
}
```

```
if (isset($b1)) {
```

```
  if ($sel1=="Cubo"){  
    echo "El cubo del numero es: " . cubo($a);  
  }
```

```
  if ($sel1=="Factorial")  
    echo "El factorial es: " . fact($a);  
}
```

```
?>
```

```
</body>
```

```
<html>
```

## USO DEL CONTROL RADIO

```
<html>
<head> <title> Control grupo </title> </head>
<body>
<!--Este es un comentario-->
<form name="f1" action="controlradio.php" method="POST">
<input type="radio" name="g1" value="1"> Casado <br>
<input type="radio" name="g1" value="2"> Soltero <br>
<input type="radio" name="g1" value="3"> Divorciado <br>
<input type="submit" name="b1" value="Elegir">
</form>
```

<?

```
switch ($_POST[g1]) {
    case 1: { echo "Seleccionaste casado"; } break;
    case 2: { echo "Seleccionaste soltero"; } break;
    case 3: { echo "Seleccionaste divorciado"; } break;
}
```

?>

```
</body>
</html>
```

☐ Casado  
☒ Soltero  
☐ Divorciado

Elegir

Seleccionaste soltero

## USO DEL CONTROL CHECKBOX

☒ Seno ☐ Cubo ☒ Factorial

45  5

Mostrar Resultados

```
<html> <body>
<form name="f1" action="controlCheckBox.php" method="POST">
<input type="checkbox" name="ch1" value="1" <? if($_POST[ch1]) echo"checked='checked' "; ?> >Seno
<input type="checkbox" name="ch2" value="2" <? if($_POST[ch2]) echo"checked='checked' "; ?> >Cubo
<input type="checkbox" name="ch3" value="3" <? if($_POST[ch3]) echo"checked='checked'"; ?> >
Factorial<br>
<input type="text" name="t1" size="3" value="<? if($_POST[ch1]) echo seno($_POST[t1]); ?>">
<input type="text" name="t2" size="3" value="<? if($_POST[ch2]) echo cubo($_POST[t2]); ?>">
<input type="text" name="t3" size="6" value="<? if($_POST[ch3]) echo fact($_POST[t3]); ?> "> <br>
<input type="submit" name="b1" value=" Mostrar Resultados "> </form>

<?
function seno($x) {
    return round(sin($x),4); }

function cubo($k) {
    return $k*$k*$k; }

function fact($n){
    $f=1;
    for($i=1;$i<=$n; $i++)
        $f=$f * $i;
    return $f; }

?>
</body> </html>
```



## MANEJO DE TABLAS

```
<html>
<head> <title>Programa tabla de multiplicar</title> </head>
<body>
<form action="tablas.php" method="post">
Numero:<input type="text" name="t1" size="5" value=" " >
<input type="submit" name="b1" value="Mostrar">
</form>
```

<?

```
$n=$_POST[t1];
$i=1;
echo "<table border=\"2\">";
echo "<tr>";
while($i<=10){
    echo "<td> $n x $i = </td> <td>". $n*$i . "</td>";
    echo "</tr>";
    $i++;
}
echo "</table>"
```

?>

```
</body>
</html>
```

Numero:

5 x 1 =	5
5 x 2 =	10
5 x 3 =	15
5 x 4 =	20
5 x 5 =	25
5 x 6 =	30
5 x 7 =	35
5 x 8 =	40
5 x 9 =	45
5 x 10 =	50

## MANEJO DE IMAGENES

```
<html>
<head>Tabla de imagenes </head>
<body>
<?
$f=$_POST[t1]; $c=$_POST[t2];
?>
<form id="form1" name="form1" method="post" action="tabla2.php">
  <input name="t1" type="text" id="t1" size="3" value="<? echo $f; ?>" />
  <input name="t2" type="text" id="t2" size="3" value="<? echo $c; ?>" />
  <input name="btn" type="submit" id="btn" value="Mostrar" />
</form>
<?
  echo "<table border='3'>";
  for($i=1; $i<=$f ; $i++) {
    echo "<tr>";
    for($j=1; $j<=$c; $j++)
      echo "<td> <img src='imagen.jpg'> </td>";
    echo "</tr>";
  }
  echo "</table>";
?>
</body> </html>
```

Tabla de imagenes

<input type="text" value="2"/>	<input type="text" value="3"/>	<input type="button" value="Mostrar"/>
--------------------------------	--------------------------------	--



&lt;?

```

function cargar($m,$f,$c){
    for($i=0; $i<$f; $i++)
        for($j=0; $j<$c; $j++)
            $m[$i][$j]=rand(1,9);
}

function imprimir($m,$f,$c){
    echo "<table border='1'>";
    for($i=0; $i<$f ; $i++) {
        echo "<tr>";
            for($j=0; $j<$c; $j++)
                echo "<td>" . $m[$i][$j] . "</td>";
        echo "</tr>";
    }
    echo "</table>";
}

function suma($m,$f,$c) {
    $s=0;
    for($i=0; $i<$f ; $i++)
        for($j=0; $j<$c; $j++)
            $s=$s + $m[$i][$j];
    return $s;
}

```

```
$f=(int)$_POST[t1];  
$c=(int)$_POST[t2];  
echo "<form action=\"matrices.php\" method=\"post\">";  
echo "Filas: <input type=\"text\" name=\"t1\" size=\"5\"> <BR>";  
echo "Cols: <input type=\"text\" name=\"t2\" size=\"5\">";  
  
echo "<input type=\"submit\" name=\"b1\" value=\"Mostrar\">";  
echo "</form>";  
  
$m=array();  
cargar(&$m,$f,$c);  
imprimir($m,$f,$c);  
echo "<br>";  
echo "La suma es:" . suma($m,$f,$c);  
?>
```

# CREACION DE CLASES

Una clase es un conjunto de objetos que comparten los mismos atributos y métodos. Una clase se define con la siguiente sintaxis:

```
class nombreClase{  
    private atributo1;  
    private atributo2;  
  
    public function metodo1(){  
        //cuerpo de la funcion  
    }  
    public function metodo2(){  
        //cuerpo de la funcion  
    }  
} // fin clase
```

Atributos

Métodos

Ejemplo, la clase numero

```
<?  
class numero {  
    private $valor;           //atributos  
  
    public function numero(){ //constructor  
        $this->valor=0;  
    }  
  
    public function esPar() { //metodos  
        if($this->valor % 2==0)  
            return true;  
        else  
            return false;  
    }  
}  
?>
```

## CLASE ALUMNO

<?

```
class alumno
{
    private $registro;
    private $nombre;
    private $paterno;
    private $materno;
    private $fecha_nac;
```

```
//constructor
public function alumno(){
    $registro=0;
    $nombre="";
    $paterno="";
    $materno="";
    $fecha_nac="";
}
```

```
//funciones insertoras
public function setRegistro($r){
    $this->registro=$r;
}

public function setNombre($n){
    $this->nombre=$n;
}

public function setPaterno($p){
    $this->paterno=$p;
}

public function setMaterno($m){
    $this->materno=$m;
}

public function setFechaNac($fn){
    $this->fecha_nac=$fn;
}
```

//funciones extractoras

```
public function getRegistro(){  
    return $this->registro;  
}
```

```
public function getNombre(){  
    return $this->nombre;  
}
```

```
public function getPaterno(){  
    return $this->paterno;  
}
```

```
public function getMaterno(){  
    return $this->materno;  
}
```

```
public function getFechaNac(){  
    return $this->fecha_nac;  
}
```

//metodos de la clase

```
public function nombreCompleto(){  
    echo $this->nombre ." ". $this->paterno ." ".  
    $this->materno;  
}
```

} //fin clase

```
$a1 = new alumno();  
$a1->setRegistro(1);  
$a1->setNombre("Bruno");  
$a1->setPaterno("Dias");  
$a1->setMaterno("Suarez");
```

```
echo "Registro:" . $a1->getRegistro(); echo "<br>";  
echo $a1->nombreCompleto(); echo "<br>";
```

?>



<?

```
class vector {
    private $n;
    private $v;
    public function vector() {
        $this->n=0;
        $this->v=array();
    }
    public function setDim($d){
        $this->n=$d;
    }
    public function getDim(){
        return $this->n;
    }
    public function setElem($x,$p){
        $this->v[$p]=$x;
    }
    public function getElem($p){
        return $this->v[$p];
    }
}
```

## CLASE VECTOR

```
public function cargar(){
    for($i=0;$i< $this->n;$i++) {
        $this->setElem(rand(10,99),$i); }
}

public function mostrar()
{
    echo "<table border=\"3\">";
    echo "<tr>";
    for($i=0;$i<$this->n;$i++)
    {
        echo "<td>" . $this->getElem($i) . "</td>";
    }
    echo "</tr>";
    echo "</table>";
}

} //end class
?>
```

&lt;?

```
include_once('clsVectores.php');
```

```
echo "<form action=\"frmVectores.php\" method=\"post\">";
```

```
echo "Dimension:";
```

```
echo "<input type=\"text\" size=\"4\" name=\"t1\" >";
```

```
echo "<input type=\"submit\" name=\"b1\" value=\"Mostrar\"> <br>";
```

```
echo "</form>";
```

```
if(isset($b1)){
```

```
    $d=(int)$_POST[t1];
```

```
    $vec1=new vector();
```

```
    $vec1->setDim($d);
```

```
    $vec1->cargar();
```

```
    $vec1->mostrar();
```

```
}
```

?&gt;

----- clsFraccion.php -----  
<?

```
class fraccion
{
    private $numerador;
    private $denominador;

    public function fraccion() {
        $this->numerador=1;
        $this->denominador=1;
    }
    public function setNumerador($valor) {
        $this->numerador=$valor;
    }
    public function getNumerador() {
        return $this->numerador;
    }
    public function setDenominador($valor) {
        $this->denominador=$valor;
    }
    public function getDenominador() {
        return $this->denominador;
    }
}
```

## CLASE FRACCION

```
public function simplificar() {
    $c=2;
    while (($c <= abs($this->numerador))&&($c <=
        abs($this->denominador))) {
        $r1= (int)($this->numerador % $c);
        $r2= (int)($this->denominador % $c);
        if(($r1==0)&&($r2==0)) {
            $this->numerador= $this->numerador / $c;
            $this->denominador= $this->denominador / $c;
        }
        else
            $c++;
    }
}

public function sumar(fraccion $a,fraccion$b) {
    $n1=$a->getnumerador();    $n2=$b->getnumerador();
    $d1=$a->getdenominador();    $d2=$b->getdenominador();
    $mc=$d1*$d2;
    $x=($mc/$d1)*$n1 + ($mc/$d2)*$n2;
    $this->numerador=$x;
    $this->denominador=$mc;
}
} //fin clase
```

?>

```
<html>
<body>
<?
include_once('clsFraccion.php');
$q1=new fraccion();
$q2=new fraccion();
$q3=new fraccion();
if($_POST['txbn1']) {
    $q1->setNumerador($_POST['txbn1']);
    $q1->setDenominador($_POST['txbd1']);
    $q2->setNumerador($_POST['txbn2']);
    $q2->setDenominador($_POST['txbd2']);
    $q3->sumar($q1,$q2);
    $q3->simplificar();
    //para q no borrar los valores de entrada
    $n= $q3->getNumerador();
    $d= $q3->getDenominador();

    $n1= $q1->getNumerador();
    $d1= $q1->getDenominador();
    $n2= $q2->getNumerador();
    $d2= $q2->getDenominador();
}
```

?>

```
<form id="form1" name="form1" method="post" action="frmFracciones.php">
  <label>Fraccion 1
  <input type="text" size="1" name="txbn1" id="txbn1" value="<?php echo $n1;?>" />
</label>/
<label>
<input type="text" size="1" name="txbd1" id="txbd1" value="<?php echo $d1;?>" />
</label>
<label>Fraccion 2
<input type="text" size="1" name="txbn2" id="txbn2" value="<?php echo $n2;?>"/>
</label>/
<label>
<input type="text" size="1" name="txbd2" id="txbd2" value="<?php echo $d2;?>"/>
</label>

<label>
<input type="submit" name="Submit" value="Sumar" />
</label>
<input type="text" size="1" name="textfield5" id="txb" value="<?php echo $n;?>"/>
</label>/
<label>
<input type="text" size="1" name="textfield6" id="txb" value="<?php echo $d;?>" />
</label>
</form>
</body>
</html>
```

## USO DE SESIONES

Una sesión es la secuencia de páginas que un usuario visita en un sitio web, desde que entra a un sitio hasta que no lo abandona. A este identificador de sesión se lo denomina, como la sesión. Para utilizar sesiones en PHP lo primero es inicializarlas. Esto lo hacemos mediante la función `session_start()`, o al registrar un variable en una sesión mediante la función `session_register('tu variable')`.

Veamos el siguiente ejemplo:

```
<?
```

```
session_start();
```

```
?>
```

```
<head>
```

```
<title>PHP</title>
```

```
<meta name="language" content="es">
```

```
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
```

```
</head>
```

```
<body>
```

<?

```
echo "Identificador de sesion: ", session_id(),"<br />";  
//Registramos una variable de sesión asignándole un nombre  
session_register('tu_variable');  
  
//Asignamos un valor a esa variable de sesión  
$tu_variable="Sergio García";  
  
//Registramos otra nueva variable de sesión  
session_register('otra_variable');  
  
//Asignamos un valor a esa nueva variable de sesión  
$otra_variable="Ramón García";  
  
//Comprobamos la existencia de la variables de sesión  
echo "Mi variable esta registrada(1):", session_is_registered('tu_variable'),"<br />";  
echo "Otra variable esta registrada(1):",session_is_registered('otra_variable'),"<br />";  
  
//recogemos las variables de sesión y sus valores en una cadena  
$codifica=session_encode();  
  
//Visualizamos la cadena resultante  
echo "Variables de sesion codificadas en una cadena: ", $codifica,"<br />";
```

```
//Eliminamos todas la variables de sesión  
session_unset();
```

```
//Comprobamos que han sido destruidas  
echo "Tu variable ha sido eliminada, aquí no hay nada: ",  
session_is_registered('tu_variable'),"<br/>";  
echo "Otra variable ha sido eliminada, aquí no hay nada: ",  
session_is_registered('otra_variable'),"<br/>";
```

```
//Recuperamos las variables de sesión desde la cadena  
session_decode($codifica);
```

```
//Comprobamos que las variables de sesión han sido recuperadas  
echo "Tu variable ha sido recuperada: ",$tu_variable,"<br />";  
echo "Otra variable ha sido recuperada: ",$otra_variable,"<br />";
```

```
?>
```

```
</body>
```

```
</html>
```