

- ❑ PHP ofrece muchas funciones predefinidas:
 - Unas pertenecen al núcleo de PHP
 - Otras las aportan los módulos que cargamos sobre el intérprete PHP
- ❑ Se puede encontrar información de las funciones de cadenas y ejemplos de uso en :

<http://php.net/manual/es/ref.strings.php>

- ❑ Leer atentamente la declaración de las funciones:

```
string substr ( string $string , int $start [, int $length ] )
```

Qué tipo de dato devuelve

Nombre de la función

Parámetros o argumentos
Cuántos, en qué orden y de qué tipo

`$a = substr('ies tetuan' , 4);`

`$b = substr($a , 2 , 2);`

Funciones de CADENAS



❑ Longitudes , conversión mayúsculas / minúsculas, código ASCII

Función	Funcionalidad	Ejemplo
strlen (cadena)	Devuelve la <i>longitud</i> (nº de caracteres) de la <i>cadena</i> . Los espacios se consideran como un carácter más.	echo strlen ('mide la longitud') ; 16
strtolower (cadena)	Devuelve la cadena con todos los caracteres en <i>minúsculas</i>	echo strtolower ('MaRiA') ; maria
strtoupper (cadena)	Devuelve la cadena con todos los caracteres en <i>mayúsculas</i>	echo strtoupper ('MaRiA') ; MARIA
ucwords (cadena)	Devuelve la cadena con <i>el primer carácter de cada palabra</i> en <i>mayúsculas</i>	echo ucwords ('hola y aDIOS') ; Hola Y ADIOS
ucfirst (cadena)	Devuelve la cadena con <i>el primer carácter</i> en <i>mayúsculas</i>	echo ucfirst ('hola y aDIOS') ; Hola y aDIOS
chr (n)	Devuelve el carácter cuyo código ASCII es <i>n</i> .	echo chr (97) ; a
ord (cadena)	Devuelve código ASCII del primer carácter de la cadena.	echo ord ('abadesa') ; 97

Funciones de CADENAS



❑ Eliminar espacios

Función	Funcionalidad	Ejemplo
ltrim (cadena)	Elimina los espacios existentes al <i>principio</i> de la cadena.	echo ltrim (' quita espacios ');
rtrim (cadena)	Elimina los espacios existentes al <i>final</i> de la cadena.	echo rtrim (' quita espacios ');
trim (cadena)	Elimina los espacios tanto al <i>principio</i> como al <i>final</i> de la cadena	echo trim (' quita espacios ');
chop (cadena)	Elimina los espacios existentes al <i>final</i> de la cadena.	echo chop (' quita espacios ');

```
echo strlen( ltrim('   quita espacios   ') ), '<br>',  
strlen( rtrim('   quita espacios   ') ), '<br>',  
strlen( trim('   quita espacios   ') ) ;
```

Funciones de CADENAS



❑ Extraer porciones de una cadena

Función	Funcionalidad	Ejemplo
substr (cadena , n)	<p>Si n es positivo extrae todos los caracteres de la cadena a partir del que ocupa la posición enésima a contar desde la izquierda.</p> <p>Si n es negativo extrae los n últimos caracteres de la cadena.</p>	<p>echo substr('Extrae caracteres' , 3) ;</p> <p>echo substr('Extrae caracteres' , 0) ;</p> <p>echo substr('Extrae caracteres' , -5) ;</p>
substr (cadena , n , m)	<p>Si n y m son positivos extrae m caracteres a partir del que ocupa la posición enésima, de izquierda a derecha.</p> <p>Si n es negativo y m es positivo extrae m caracteres a partir del que ocupa la posición enésima contada de derecha a izquierda.</p> <p>Si n es positivo y m es negativo extrae la cadena comprendida entre el enésimo carácter hasta el enésimo, contando en este caso de derecha a izquierda</p> <p>Si n es negativo y m es negativo extrae la porción de cadena comprendida entre el enésimo y el enésimo caracteres contando, en ambos casos, de derecha a izquierda.</p> <p>Si el valor absoluto de n es menor que el de m devuelve una cadena vacía.</p>	<p>echo substr('Extrae caracteres' , 2,6) ;</p> <p>echo substr('Extrae caracteres' , 0,8) ;</p> <p>echo substr('Extrae caracteres' , 2,-3) ;</p> <p>echo substr('Extrae caracteres' , -7,5) ;</p> <p>echo substr('Extrae caracteres' , -7,-5) ;</p> <p>echo substr('Extrae caracteres' , -5,-7) ;</p>

Funciones de CADENAS



❑ Modificaciones de cadenas

Función	Funcionalidad	Ejemplo
strrev (<i>cadena</i>)	Devuelve la cadena invertida	echo strrev('Invierte cadena'); <i>anedac etreivnl</i>
str_repeat (<i>cadena</i> , <i>n</i>)	Devuelve la cadena repetida tantas veces como indica n .	echo str_repeat('Rep', 3); <i>RepRepRep</i>
str_pad (<i>cad</i> , <i>n</i> , <i>rell</i> , <i>tipo</i>)	Añade a la cadena <i>cad</i> los caracteres indicados en <i>rell</i> hasta alcanzar la longitud que indica n . El parámetro tipo puede tomar uno de estos valores: STR_PAD_BOTH (rellena por ambos lados) STR_PAD_RIGHT (<i>rellena</i> por la derecha) STR_PAD_LEFT (rellena por la izquierda). Si se omite la cadena de <i>Relleno</i> usará <i>espacios</i> y si se omite el <i>tipo</i> rellenará por la <i>derecha</i>	echo str_pad('Rep',5,'@',STR_PAD_BOTH); <i>@Rep@</i> echo str_pad('Rep',5,'@',STR_PAD_RIGHT); <i>Rep@@</i> echo str_pad('Rep',5,'@',STR_PAD_LEFT); <i>@@Rep</i> echo str_pad('Rep',5,'@'); <i>Rep@@</i>
str_replace (<i>cad1</i> , <i>cad2</i> , <i>cadena</i>)	Sustituye en <i>cadena</i> las apariciones de <i>cad1</i> por <i>cad2</i>	echo str_replace('e','ari','PepE'); <i>ParipE</i>
substr_replace (<i>cadena</i> , <i>cad2</i> , <i>n</i> , <i>m</i>)	Sustituye en <i>cadena</i> los <i>m</i> caracteres a partir de la posición <i>enesima</i> por <i>cad2</i>	echo substr_replace('Pepe','Luis', 2, -1); <i>PeLuisse</i>

❑ División de cadenas

explode(*separador*, *cadena* [, *n*])

Devuelve un **array** construido a partir de la **cadena** dividiéndola en fragmentos delimitados por el carácter **separador** hasta el máximo de porciones señaladas por **n**.

Los caracteres separadores no se incluyen en las cadenas resultantes.

Si no se indica la cantidad de porciones, será fraccionada toda la cadena. Si se indica número, el último *trozo* contendrá **toda** la cadena restante

```
<?php
$a = 'dividir la cadena';
$partes1=explode(' ', $a );
echo $partes1[0], $partes1[1], $partes1[2]; // dividir la cadena
$partes2=explode('a', $a );
echo $partes2[0], $partes2[1], $partes2[2]; // dividir l c den
$partes3=explode(' ', $a , 2 );
echo $partes3[0], $partes3[1]; // dividir la cadena
var_dump( explode('@', '1@2@3@4@5' ) );
?>
```

❑ División de cadenas

parse_str (cadena)

Interpreta cadena como si contuviera la cadena de consulta generada por los datos enviados desde un formulario con el método GET, y crea automáticamente variables para cada uno de ellos.

```
<?php
parse_str( 'pagina=1&seccion=2' );
// la sentencia anterior crea 2 variables con el nombre de los elementos enviados y contenido el valor asignado
echo $pagina . ' ' . $seccion;
?>
```

❑ Unión de cadenas

implode(*separador*, *array*)

Devuelve una cadena formada por todos los elementos del **array** separados mediante los caracteres indicados en **separador**.

```
<?php
$lista = explode('@', '1@2@3@4@5' ) ;
$mensaje = implode(' - ' , $lista );
echo $mensaje ; // muestra 1 - 2 - 3 - 4 - 5
?>
```


❑ Búsqueda

strpos(*cadena* , *cad*)

Devuelve la posición de la primera ocurrencia de **cad** dentro de **cadena**.

Sí no se encuentra devuelve FALSE, no 0

Para usar la respuesta de esta función en una comparación es recomendable usar el operador **===** pues de otro modo podría confundirse la no existencia con que se encuentra en posición 0

```
<?php
$_mail = $_GET['e_mail'];
$pos = strpos($_mail, '@'); // obtiene posición del primer carácter @
If ( $pos == 0 ) // es true tanto sí no está como sí está en la posición 0
{ .....
}
?>
```

Funciones de CADENAS



❑ Búsqueda

strstr(*cadena* , *cad* , [*anterior* = FALSE])

Devuelve contenido de **cadena** desde la posición donde encuentre la primera ocurrencia de **cad**.

Sí no se encuentra devuelve FALSE, no 0

Sí se da el valor TRUE para el parámetro anterior, devuelve contenido de cadena antes de encontrar la primera ocurrencia de cad.

```
<?php
$email = 'name@example.com';
$dominio = strstr($email, '@');
echo $dominio; // mostrará @example.com

$user = strstr($email, '@', true);
echo $user; // mostrará name
?>
```

Funciones de CADENAS



❑ Búsqueda y recuento

substr_count(*cadena* , *cad* [, *ini* [, *longitud*]])

Devuelve cuantas veces aparece el contenido dado en **cad** dentro de **cadena**.

Se puede indicar posición de inicio (**ini**) y extensión (**longitud**) de la búsqueda

Diferencia entre mayúsculas y minúsculas

```
<?php
$texto = 'Esto es un test' ;
echo substr_count( $texto , 'es' ) ; // muestra 2
echo substr_count( $texto , 'es' , 6 ) ; // busca y cuenta a partir de posición 6
echo substr_count( $texto , 'es' , 6 , 4 ) ; // busca y cuenta en 4 caracteres a partir de posición 6
?>
```

Funciones de CADENAS



❑ Búsqueda y recuento

count_chars(*cadena* , *modo*)

Cuenta apariciones de caracteres en la **cadena**. El modo establece qué va a contar:

Modo	Devuelve
0	array cuyos índices son códigos ASCII de caracteres y los valores son el nº de veces que se repite
1	Igual que 0 , pero solo de los que aparecen al menos 1 vez
2	Igual que 0 , pero solo de los que aparecen 0 veces
3	Cadena con los caracteres que aparecen solo 1 vez
4	Cadena con los caracteres que no aparecen

```
<?php
```

```
$texto = 'Contando caracteres' ;
```

```
$frecuencia = count_chars ( $texto , 0 ) ;
```

```
echo " el carácter a aparece " . $frecuencia[ ord('a') ] . " veces en el texto <br> " ;
```

```
?>
```



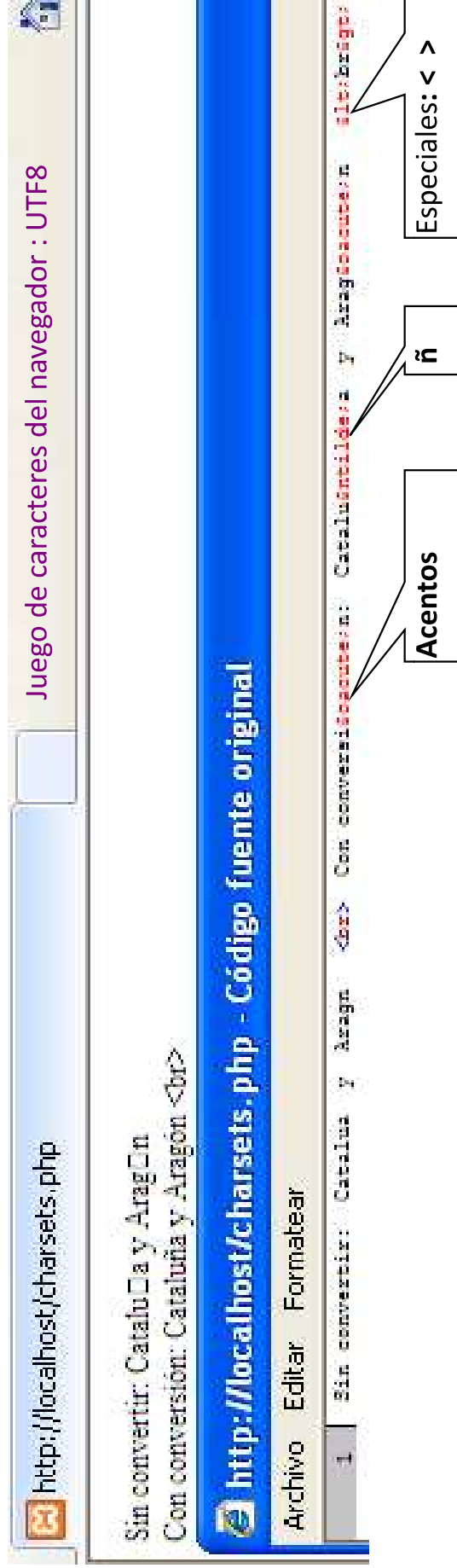
Funciones de CADENAS

❑ Marcas y caracteres en HTML

htmlspecialchars(*cadena* [, *flags*])

Devuelve una cadena con los caracteres no pertenecientes al estándar ANSI convertidos en entidades HTML.

```
<?php // Guardado como TEST.php en codificación ANSI
echo " Sin convertir: Cataluña y Aragón <br> ";
echo htmlspecialchars( "Con conversión: Cataluña y Aragón <br> " );
?>
```



Funciones de CADENAS



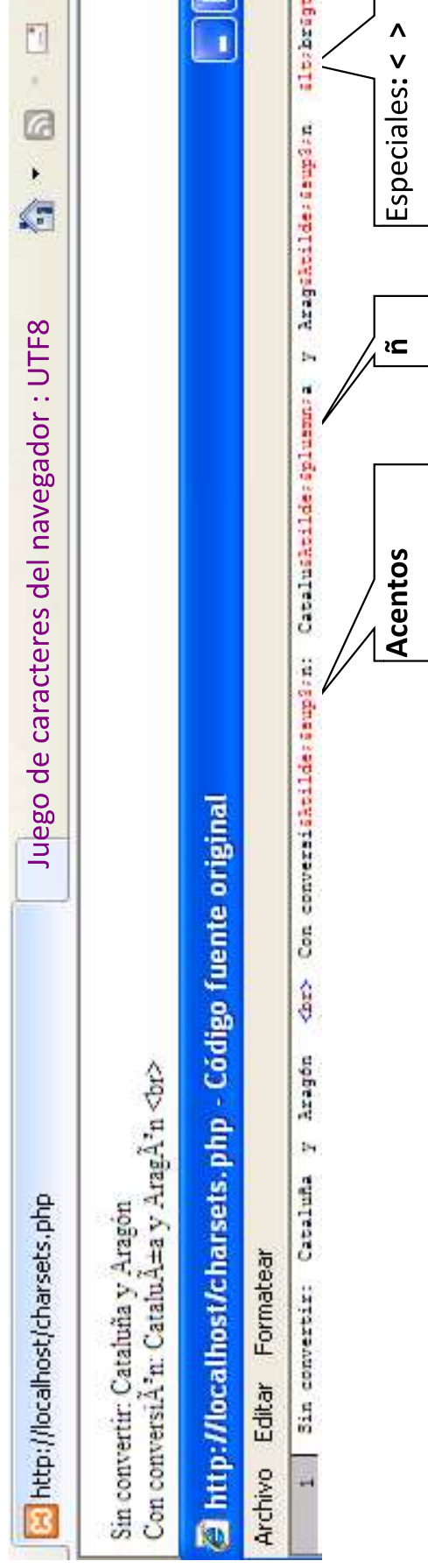
❑ Marcas y caracteres en HTML

htmlspecialchars(cadena [, flags])

En versiones anteriores a PHP 5.4 por defecto la codificación usada en la conversión es ISO-8859-1; A partir de PHP 5.4 por defecto es UTF-8

Usa conversión por defecto a ISO-8859-1

```
<?php // Guardado como TEST.php en codificación UTF-8
echo " Sin convertir: Cataluña y Aragón <br> ";
echo htmlspecialchars( "Con conversión: Cataluña y Aragón <br> " );
?>
```



Funciones de CADENAS

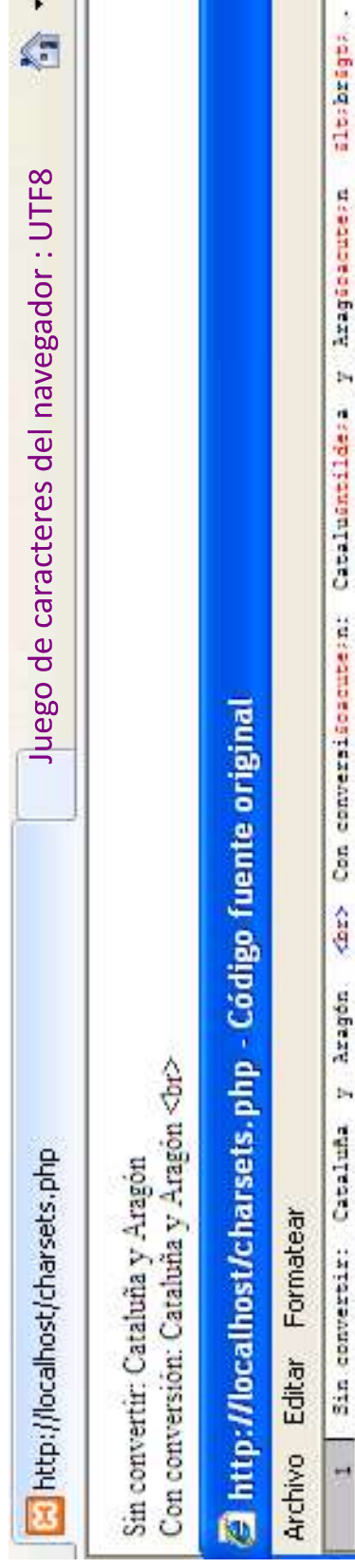


❑ Marcas y caracteres en HTML

htmlspecialchars(cadena [, flags])

```
<?php // Guardado como TEST.php en codificación UTF-8
echo " Sin convertir: Cataluña y Aragón <br> ";
echo htmlspecialchars( "Con conversión: Cataluña y Aragón <br> ", ENT_QUOTES, "UTF-8" );
?>
```

Se indica que use conversión UTF-8



Juego de caracteres del navegador : UTF8

Sin convertir: Cataluña y Aragón
Con conversión: Cataluña y Aragón

http://localhost/charset.php - Código fuente original

Archivo Editar Formatear

1 Sin convertir: Cataluña y Aragón.
 Con conversión: Cataluña y Aragón.

Acentos

ñ

Especiales: < >

Funciones de CADENAS



❑ Marcas y caracteres en HTML

html_entity_decode(*cadena* [, *flags*])

Devuelve una cadena con las entidades HTML de \$cadena convertidas en sus caracteres correspondientes.

```
<?php // Guardado como TEST.php en codificacion UTF-8
$a = htmlentities(' Cataluña y Aragón <br> ' , ENT_QUOTES, "UTF-8" );
echo html_entity_decode($a);
?>
```

Usa conversión
por defecto a
ISO-8859-1



❑ Marcas y caracteres en HTML

`htmlspecialchars(cadena)`

Devuelve una copia de cadena con los caracteres especiales de HTML (<, >, ', " y &) convertidos en entidades HTML.

Muy útil para mostrar código en una página.

```
<?php
echo htmlspecialchars('<?php echo \'Hola\' ?>') . '<br />';
?>
```

`htmlspecialchars_decode(cadena)`

Devuelve una copia de la cadena con las entidades html descodificadas, es decir convierte **<** en **<** y **>** en **>**.

```
<?php
echo htmlspecialchars_decode('&lt;strong&gt;HOLA&lt;/strong&gt;') ;
?>
```

❑ Marcas y caracteres en HTML **nl2br(*cadena*)**

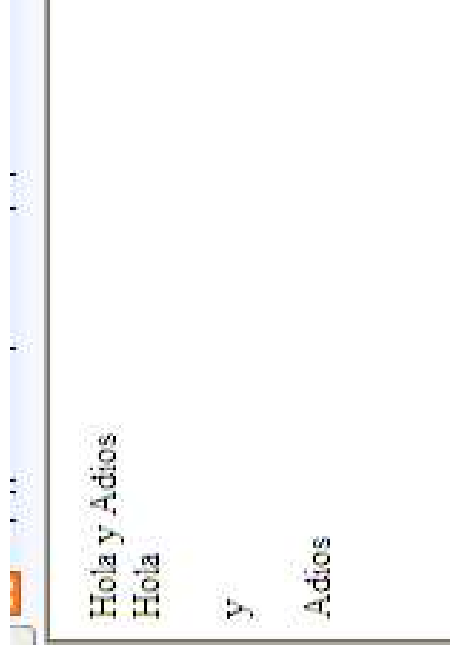
Devuelve una copia de la cadena con todos los saltos de línea (`\r\n`, o `\n\r`, o `\n` o `\r`) convertidos en saltos de línea HTML (`
`).

```
<?php
$cadena=<<'Marca'
Hola

y

Adios
Marca;
echo $cadena . '<br>';
echo nl2br($cadena);
```

?>



Funciones de CADENAS



❑ Encriptación

bin2hex(*cadena*)

Devuelve una cadena ASCII que contiene la representación hexadecimal de la cadena. La conversión se realiza byte a byte

crypt(*cadena*)

Devuelve la cadena encriptada utilizando una semilla aleatoria de dos caracteres. Por su carácter aleatorio, si se ejecuta dos veces seguidas dará resultados diferentes.

crypt(*cadena* , 'xx')

Devuelve la cadena encriptada usando como semilla los dos caracteres dados en el segundo parámetro.

El resultado de las funciones crypt comienzan con los 2 caracteres usados como semilla.

```
<?php
$a = 'Encriptame';
echo crypt($a) . '<br>';
echo crypt($a) . '<br>';
echo crypt($a,'z@') . '<br>';
echo crypt($a,'z@') . '<br>';
echo crypt($a,'z@') . '<br>';
echo crypt($a,'a3') . '<br>';
?>
```

```
$1$Rt3.GD1.SkZChIqN2GUA19czsAK2Ry1
$1$Xm1.Ui0.SvjrombXlr4zLAGdoQAIHS1
z@02A1Wb1TUA
z@02A1Wb1TUA
a3CyGDAmwNOp.
```

Funciones de CADENAS



❑ Encriptación

md5(*cadena*)

Aplica el algoritmo md5 y devuelve la huella digital generada por él.
Este algoritmo genera una cadena de 32 caracteres –huella digital – sea cual sea la longitud de la cadena a la que se aplique.
No dispone de mecanismo inverso

crc32(*cadena*)

Aplica el algoritmo crc32 de comprobación de integridad y devuelve el valor del mismo.
Se usa mucho en programas de compresión / descompresión de ficheros. Se aplica al comprimir y se incluye el valor obtenido dentro del fichero comprimido. Después de la descompresión se vuelve a aplicar el mismo algoritmo y se comparan ambos valores. La coincidencia será la garantía de que el fichero obtenido es idéntico al original.

```
<?php
$a = 'Encriptame';
echo md5($a) . '<br>';
echo md5($a) . '<br>';
echo crc32($a) . '<br>';
?>
```

```
67c3ca0aefda2595138168a85e7b33a0
67c3ca0aefda2595138168a85e7b33a0
-1128189886
```