



**UTN.BA** FACULTAD  
REGIONAL  
BUENOS AIRES  
SECRETARÍA DE EXTENSIÓN UNIVERSITARIA FRBA UTN

**Centro de  
e-Learning**

# **EXPERTO UNIVERISTARIO EN MySQL Y PHP NIVEL INTERMEDIO**



[www.sceu.frba.utn.edu.ar/e-learning](http://www.sceu.frba.utn.edu.ar/e-learning)



## Módulo 1

# FUNCIONES PARA EL PROCESAMIENTO DE CADENAS



## Funciones para el preprocesamiento de cadenas.



## Presentación de la Unidad:

**En esta unidad veremos algunas de las funciones que brinda PHP para el procesamiento de cadenas.**



## Objetivos:

- ❖ **Aprender a utilizar algunas funciones que nos serán de utilidad a la hora de manejar variables del tipo string.**



## Temario:

### *Introducción*

*Funciones: echo y print()*

*Función: printf()*

*Función: sprintf()*

*Función: strtoupper()*

*Función: strtolower()*

*Funcion: ucwords()*

*Función: trim()*

*Funciones: ltrim() y chop()*

*Función: strlen()*

*Función: chr()*

*Función: ord()*

*Función: number\_format()*

*Función: substr()*

*Función: explode()*

*Función: nl2br()*

*Función: ucfirst()*

*Función: ucwords()*

*Función: strpos()*

*Función str\_replace()*



## CONSIGNAS PARA EL APRENDIZAJE COLABORATIVO

En esta Unidad los participantes se encontrarán con diferentes tipos de consignas que, en el marco de los fundamentos del MEC\*, los referenciarán a tres comunidades de aprendizaje, que pondremos en funcionamiento en esta instancia de formación, a los efectos de aprovecharlas pedagógicamente:



1. Los foros asociados a cada una de las unidades.
2. La Web 2.0.
3. Los contextos de desempeño de los participantes.

Es importante que todos los participantes realicen las actividades sugeridas y compartan en los foros los resultados obtenidos.

## FUNCIONES PARA EL PROCESAMIENTO DE CADENAS.

### Introducción

El lenguaje PHP ofrece una serie de herramientas para el procesamiento de cadenas, que es muy utilizado en mostrar datos con un cierto formato o en almacenar en una base de datos información de una manera determinada.

A continuación se detallan algunas funciones para el manejo de cadenas:

### Funciones: *echo* y *print()*

Estas dos instrucciones, que ya hemos visto anteriormente, se utilizan para mostrar una o varias cadenas en la pantalla. Generalmente se ha venido utilizando la instrucción *echo* en los ejemplos realizados, pero también es posible utilizar la función *print()*, con la diferencia de que en esta función, la cadena a imprimir debe estar entre paréntesis, cosa que con la instrucción *echo*, al no tratarse en realidad de una función, no es necesario.

Ejemplo:

```
echo $nombre; print ($domicilio);
```

### Función: *printf()*

Esta función es similar a la función *print()*, pero permite dar un cierto formato a lo que se desea imprimir. La Sintaxis de esta función es:

```
printf ("formato", $cadena);
```

formato debe ser una letra permitida.

\$cadena es la variable a imprimir.

Los tipos de formatos permitidos, son los siguientes:

Letra de Formato	Especificación de lo que realiza
b	Expresa a la variable como número binario
c	Expresa a la variable como carácter ASCII
d	Expresa a la variable como número decimal
f	Expresa a la variable como número de punto flotante



o	Expresa a la variable como número octal
s	Expresa a la variable como un string (cadena)
x	Expresa a la variable como número hexadecimal en minúsculas
X	Expresa a la variable como número hexadecimal en Mayúsculas

Para establecer que la letra es un formato de impresión, se debe colocar delante de ella el signo de porcentaje (%).

En resumen, la cadena que contiene las instrucciones de formato permite usar tanto patrones como caracteres normales. Los patrones van precedidos del signo %, y sus valores se incluirán en la cadena formateada, o indicarán como debe ser presentada.

Entre el signo % y el caracter especificador podemos incluir uno o dos números, ambos opcionales. Si existen, el primer número indica el número de caracteres (anchura) que debe ocupar la cadena formateada. printf devolverá como mínimo ese número de caracteres, o más si el resultado excede. Si no indicamos este número, printf devolverá los caracteres que sean necesarios.

Para rellenar los caracteres necesarios usualmente se usan espacios en blanco, aunque podemos indicar que alternativamente se usen ceros o cualquier otro carácter. Estos caracteres de relleno normalmente preceden al resultado, aunque con el - podemos hacer que vayan al final.

Si el primer número es un cero, la anchura extra será llenada con ceros y no con espacios en blanco. En este caso se ignora el signo - aunque lo hayamos puesto. Y podemos especificar cualquier otro carácter de relleno si le precede una comilla simple '.

Ejemplo:

Este ejemplo almacena el valor del sueldo en la variable \$sueldo, y luego lo muestra de diferente forma:

```
<html>
<head>
<title> ejemplo1-printf.php </title>
</head>
<body>
<?php
    $edad="38 años";
    printf ("%d", $edad);
    printf ("<br/>");
    printf ("%4.2f", $edad);
?>
</body>
</html>
```

Centro de Formación, Investigación y Desarrollo de Soluciones de e-Learning.

UTN - FRBA. Secretaría de Cultura y Extensión Universitaria

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148 // e-learning@sceu.frba.utn.edu.ar

Ejemplo:

Asignar a una variable, una determinada cadena y posteriormente mostrarla en todos los formatos descriptos.

```
<html>
<head>
<title> ejemplo2-formatos.php </title>
</head>
<body>
<?php
$cadena = "76.9 minutos"; printf("resultado en formato binario: "); printf("%b",
$cadena);
printf("<br/>");
printf("resultado según la tabla ascii: "); printf("%c", $cadena);
printf("<br/>");
printf("resultado en formato decimal: "); printf("%d", $cadena);
printf("<br/>");
printf("resultado en formato float: "); printf("%f", $cadena); printf("<br/>");
printf("resultado en formato octal: "); printf("%o", $cadena); printf("<br/>");
printf("resultado en formato string: "); printf("%s", $cadena);
printf("<br/>");
printf("resultado en formato hexadecimal (minúscula: "); printf("%x", $cadena);
printf("<br/>");
printf("resultado en formato hexadecimal (mayúscula: "); printf("%X", $cadena);
?>
</body>
</html>
```

### ***Función: sprintf()***

Es similar a la función printf, con la diferencia de que no imprime el resultado en el navegador, sino que se utiliza para almacenar una cadena en una variable.

La sintaxis es la siguiente:

```
variable = sprintf ("Formato", cadena);
```

En este caso, sprintf, devuelve en la variable el valor de la cadena transformado de acuerdo al Formato indicado. El Formato puede ser cualquier letra de las descriptas

Centro de Formación, Investigación y Desarrollo de Soluciones de e-Learning.

UTN - FRBA. Secretaría de Cultura y Extensión Universitaria

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148 // e-learning@sceu.frba.utn.edu.ar



para la función printf.

Ejemplo:

IDEM al ejemplo 2 utilizando sprintf.

```
<html>
<head>
<title> ejemplo3-sprintf.php </title>
</head>
<body>
<?php
$cadena = "76.9 minutos";
// transformación a binario
$variable = sprintf("%b<br/>", $cadena);
// transformación a caracteres
$variable .= sprintf("%c<br/>", $cadena);
// transformación a decimal
$variable .= sprintf("%d<br/>", $cadena);
// transformación a float

$variable .= sprintf("%f<br/>", $cadena);
// transformación a octal: ";
$variable .= sprintf("%o<br/>", $cadena);
// transformación a string
$variable .= sprintf("%s<br/>", $cadena);
// transformación a hexadecimal (minúscula)
$variable .= sprintf("%x<br/>", $cadena);
// transformación a hexadecimal (mayúscula)
$variable .= sprintf("%X<br/>", $cadena); echo $variable;
?>
</body>
</html>
```

### ***Función: strtoupper()***

Esta función permite transformar una cadena a mayúsculas.

Ejemplo:

Almacenar una frase en una variable y posteriormente mostrarla en mayúsculas.

```
<html>
<head>
```

```
<title> ejemplo4-strtoupper.php </title>
</head>
<body>
<?php
$frase = "hoy es un día soleado."; echo "<h3>";
echo "frase ingresada: " . $frase; echo "</h3><br/><br/>";
echo "frase en mayúsculas: " . strtoupper($frase);
?>
</body>
</html>
```

### **Función: strtolower()**

Es la inversa de la función anterior, es decir que transforma una cadena en minúsculas.

Ejemplo:

Almacenar una frase en una variable y posteriormente mostrarla en minúsculas.

```
<html>
<head>
<title> ejejemplo5-strolower.php </title>
</head>
<body>
<?php
$frase = "ESTE APUNTE ES DE PHP";
echo "<h3>";
echo "frase ingresada: " . $frase; echo "</h3><br/><br/>";
echo "frase en minúsculas: " . strtolower($frase);
?>
</body>
</html>
```

### **Funcion: ucwords()**

Convierte el primer carácter de cada palabra en mayúsculas.



Ejemplo:

Inicializar una variable con el nombre y apellido de una persona y posteriormente mostrarla con sus primeros caracteres en mayúsculas.

```
<html>
<head>
<title> ejemplo6-ucwords.php </title>
</head>
<body>
<?php
$nombre = "marcelo rodríguez"; echo "<h3>";
echo "cadena original: " . $nombre; echo "</h3><br/><br/>";
echo "cadena convertida: " . ucwords($nombre);
?>
</body>
</html>
```

### ***Función: trim()***

Mediante esta función, se eliminan los espacios en blanco al inicio y al final de una cadena. También elimina los caracteres de control: “\n”, “\r”, “\t” y “\0”.

Cuando trabajamos con formularios, debemos asegurar que los datos solicitados llegan a la base de datos o al fichero con una cierta integridad, a fin de tener un sistema homogéneo y bien ordenado. Una de las funciones de manejo de cadenas que nos ayudarán en esta tarea será la función trim(). Con ella, eliminaremos los espacios en blanco existentes al principio y al final de una cadena, quedando aislada la palabra o frase introducida.

Ejemplo:

```
<html>
<head>
<title> ejemplo7-trim.php </title>
</head>
<body>
<?php
$profesion = " técnico constructor "; echo "<h3>";
echo "cadena original: " . $profesion;
echo "</h3><br/><br/>";
echo "cadena convertida: " . trim($profesion);
?>
```

Centro de Formación, Investigación y Desarrollo de Soluciones de e-Learning.

UTN - FRBA. Secretaría de Cultura y Extensión Universitaria

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148 // e-learning@sceu.frba.utn.edu.ar

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

Al ejecutar este ejemplo, no se observarán diferencias al visualizarlo en el navegador, pero si se observarán diferencias si se visualiza el código generado por el programa.

Existe además una funcionalidad casi desconocida de la función trim(), ya que no solamente descarta los espacios en blanco al principio y al final de la cadena de texto que le pasamos por parámetro, sino que en PHP dicha funcionalidad puede ser ampliada y le podemos indicar cuál o cuáles caracteres queremos que elimine al principio y al final de la cadena de texto.

La función se definiría de la siguiente manera:

```
trim ( string cadena [, string lista_caracteres] )
```

```
<html>  
<head>  
<head><title> ejemplo7-trim.php </title></head>  
</head>  
<body>  
<?php  
$texto = "--- Texto con guiones primero y sin guines luego de la función trim() ---"; echo  
"<h3>";  
echo "cadena original: " . $texto;  
echo "<br/><br/>";  
echo "cadena convertida: " . trim($texto, "-"); echo "</h3>";  
?>  
</body>  
</html>
```

### **Funciones: ltrim() y chop()**

Funcionan de forma similar a trim(), con la diferencia de que ltrim(), elimina espacios en blanco al principio de la cadena, mientras que chop(), elimina espacios en blanco al final de la cadena.

### Función: *strlen()*

Esta función permite obtener la longitud de una cadena de caracteres pasada como parámetro.

Ejemplo:

Almacenar en una variable una cadena de caracteres y luego mostrar la cantidad de caracteres que la constituyen.

```
<html>
<head>
<title> ejemplo8-strlen.php </title>
</head>
<body>
<?php
$cadena = "php es un lenguaje de programación."; echo "<h3>";
echo "cadena : " . $cadena; echo "</h3><br/><br/>";
echo "cantidad de caracteres: " . strlen($cadena);
?>
</body>
</html>
```

### Función: *chr()*

La función *chr()* recibe un valor numérico y devuelve el carácter de la tabla ASCII que le corresponde.

Ejemplo:

Imprimir la tabla ASCII desde el código 30 en adelante.

```
<html>
<head>
<title> ejemplo9-chr.php </title>
</head>
<body>
<?php
echo "<h3>";
echo "<table border=\"2\" align=\"center\" width=\"100%\">";
```

```
$c=30;
```

Centro de Formación, Investigación y Desarrollo de Soluciones de e-Learning.

UTN - FRBA. Secretaría de Cultura y Extensión Universitaria

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148 // e-learning@sceu.frba.utn.edu.ar



```
for ($i=1; $i<24; $i++)
{
    echo "<tr>";
    for ($j=1; $j<=10; $j++)
    {
        echo "<th align=\"center\">" . $c . "</th>"; echo "<td align=\"center\">". chr($c) .
        "</td>";
        $c++;
    }
    echo "</tr>";
}
echo "</table>";
?>
</body>
</html>
```

#### **Función: ord()**

Es la función inversa a la chr(), es decir que dado un carácter como parámetro, la función ord(), devuelve el valor del código que le corresponde.

Ejemplo:

Asignar a una variable un carácter y posteriormente mostrar el código que le corresponde:

```
<html>
<head>
<title> ejemplo10-ord.php </title>
</head>
<body>
<?php
echo "<h3>";
$c = "l";
echo "el caracter: $c tiene como código: " .ord($c); echo "</h3>"; ?>
</body>
</html>
```





### *Función: number\_format()*

La función `number_format` permite dar un formato deseado a un valor numérico determinado.

La sintaxis es la siguiente:

`number_format(variable Numerica, Cantidad De Decimales, "Separador de decimales", "Separador de Miles") ;`

Ejemplo:

Dado un cierto valor numérico, imprimirlo con dos decimales separados por una coma, mostrando también los miles separados por puntos.

```
<html>
<head>
<title> ejercicio11-numberformat.php </title>
</head>
<body>
<?php
echo "<h3>";
$valor = 8546921546.4875;
echo "el valor corresponde a: ";
echo number_format($valor,2,"","."); echo "</h3>";
?>
</body>
</html>
```

### *Función: substr()*

Permite extraer una porción de texto de la cadena pasada como parámetro. Su sintaxis es la siguiente:

`substr(cadenaOriginal, lugarInicial, cantidadDeCaracteres)`

**cadenaOriginal:** Es la cadena de donde se extraerá el texto.

**lugarInicial:** Representa la posición desde donde se comenzarán a extraer el texto.

**CantidadDeCaracteres:** Indica cuantos caracteres se extraerán. (Opcional).

Ejemplo:

Centro de Formación, Investigación y Desarrollo de Soluciones de e-Learning.

UTN - FRBA. Secretaría de Cultura y Extensión Universitaria

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148 // [e-learning@sceu.frba.utn.edu.ar](mailto:e-learning@sceu.frba.utn.edu.ar)



```
<html>
<head>
<title> ejemplo12-subsrt.php </title>
</head>
<body>
<?php
echo "<h3>";
$cadena = "alicia martinez";
echo "cadena original: " . $cadena; echo "<br/>";
echo "tres primeros caracteres: "; echo substr($cadena,0,3);
echo "<br/>";
echo "seis caracteres a partir del octavo: "; echo substr($cadena,7,6);
echo "<br/>";
echo "cuatro caracteres comenzando desde el último: "; echo substr($cadena,-4);
echo "</h3>";
?>
</body>
</html>
```

### ***Función: explode()***

Divide una cadena en varias cadenas.

La sintaxis de la función es la siguiente:

array explode ( string \$delimiter , string \$string [, int \$limit ] )

Devuelve un array de cadenas, siendo cada una de ellas una subcadena del parámetro string formado por la división según los delimitadores indicados en el parámetro delimiter.

Parámetros:

Delimiter: La cadena delimitadora.

String: La cadena de entrada.

Limit:

Si el parámetro limit es positivo, el array devuelto contendrá el máximo de elementos en el limit y el último elemento contendrá el resto del string.

Si el parámetro limit es negativo, se devolverán todos los componentes a excepción el último -limit.

Si el parámetro limit es cero, actuará como si su valor fuera 1.

Ejemplos:



```
<?php
// Ejemplo 1
$pizza = "piece1 piece2 piece3 piece4 piece5 piece6";
$pieces = explode(" ", $pizza);
echo $pieces[0]; // piece1
echo $pieces[1]; // piece2
```

```
// Ejemplo 2
$data = "foo*:1023:1000::/home/foo:/bin/sh";
list($user, $pass, $uid, $gid, $gecos, $home, $shell) = explode(":", $data);
echo $user; // foo
echo $pass; // *
```

?>

```
<?php
// Ejemplo 3
/* Un string que no contiene el delimitador simplemente devolverá un array de un
elemento con el string original. */
$input1 = "hello";
$input2 = "hello,there";
var_dump( explode( ',', $input1 ) );
var_dump( explode( ',', $input2 ) );
```

?>

El resultado del ejemplo sería:

```
array(1)
(
    [0] => string(5) "hello"
)
array(2)
(
    [0] => string(5) "hello"
    [1] => string(5) "there"
)
```

```
<?php
/* Ejemplo del parámetro limit. */
$str = 'one|two|three|four';

// positive limit
print_r(explode('|', $str, 2));
```



```
// negative limit (since PHP 5.1)
print_r(explode('|', $str, -1));
?>
```

El resultado del ejemplo sería:

Array

```
(
  [0] => one
  [1] => two|three|four
)
```

Array

```
(
  [0] => one
  [1] => two
  [2] => three
)
```

### **Función: nl2br()**

Inserta saltos de línea HTML antes de todas las nuevas líneas de un string.

La sintaxis es la siguiente:

```
string nl2br ( string $string [, bool $is_xhtml = true ] );
```

Devuelve el parámetro string con un '<br />' o '<br>' insertado antes de cada nueva línea. (\r\n, \n\r, \n y \r).

Ejemplo:

```
<?php
echo nl2br("Bienvenido\r\nEste es mi documento HTML", false);
?>
```

El resultado del ejemplo sería:

Bienvenido<br>

Este es mi documento HTML

### **Función: ucfirst()**

Permite pasar a mayúscula el primer carácter de una cadena. La función ucfirst() es útil cuando se desea imprimir nombres propios.

Ejemplo:

```
<html>
<head><title> ejemplo17-ucfirst.php </title></head>
<body>
<?php
echo "<h3>";
$cadena = "alejandra";
echo "cadena original: " . $cadena; echo "<br/><br/>";
echo "cadena procesada: " . ucfirst($cadena);
echo "<br/><br/>"; echo "</h3>";
?>
</body>
</html>
```

### **Función: ucwords()**

Permite cambiar a mayúsculas el primer carácter de todas las palabras de una cadena.

Ejemplo:

```
<html>
<head><title> ejemplo18-ucwords.php </title></head>
<body>
<?php
echo "<h3>";
$cadena = "javier es buen alumno"; echo "cadena original: " . $cadena; echo
"<br/><br/>";
echo "cadena procesada: " . ucwords($cadena); echo "<br/><br/>";
echo "</h3>";
?>
</body>
</html>
```

### *Función: strpos()*

Esta función devuelve la posición que se encuentra en una cadena fuente una cadena clave. La sintaxis es la siguiente:

\$Pos = strpos(cadenaFuente, cadenaClave, comienzo);

cadenaFuente: Es la cadena que será recorrida.

cadenaClave: Es la cadena buscada.

comienzo: Es opcional, y determina de que posición comenzará a explorar. Si se ignora, comienza desde el principio, posición cero.

Ejemplo:

Almacenar en una cadena, una frase y luego una palabra clave a buscar su posición. Imprimir la posición que le corresponde.

```
<html>
<head>
<title> ejemplo19-strpos.php </title>
</head>
<body>
<?php
echo "<h3>";
$cadena = "en un ecosistema conviven diversos tipos de organismos";
$clave = "eco";
echo "cadena original: " . $cadena; echo "<br/><br/>";
echo "clave buscada: " . $clave;

echo "<br/><br/>";
$p = strpos($cadena, $clave); if ($p!="")
{
echo "la clave se encuentra en la posición: $p" ;
}
else
{
echo "no existe la clave en la cadena fuente";
}
echo "</h3>";
?>
</body>
</html>
```

Tener presente que la primera posición es Cero.

### *Función str\_replace()*

En PHP existe una función llamada str\_replace(); que permite reemplazar caracteres de una cadena.

La sintaxis de la función str\_replace(); tiene 3 parámetros, el primero es la cadena a buscar, el segundo es la cadena por la que se va a reemplazar y por último se pasa como parámetro la cadena original.

Las cadenas de texto en PHP funcionan como un arreglo de caracteres, por lo que podemos acceder a partes de él como si de un arreglo se tratara. Eso sería una forma, sin embargo la función str\_replace() nos simplifica ese trabajo:

```
$cadena = str_replace("a","@",$cadena);
```

Solo con la función anterior podemos reemplazar la cantidad de código que hemos utilizado anteriormente para prácticamente lo mismo.

Podemos utilizar entonces, str\_replace (String Replace) en PHP para buscar y reemplazar datos con datos alternativos. Por ejemplo, para filtrar los mensajes del foro y encontrar palabras ofensivas y reemplazarlas con asteriscos.

Utilizando la instrucción:

```
str_replace (buscar, reemplazar_con,en_el_contenido_de);
```

El primer parámetro es lo que está buscando, el segundo parámetro es por lo que se va a reemplazar, y el tercer parámetro es la cadena donde se desea buscar. En los tres parámetros se puede utilizar una matriz como entrada si se desea.

Ejemplo:

```
<?php
$swear = array("palabra1", "palabra2", "palabra3", "palabra4", "palabra5",
"palabra6");
$text="Hola palabra1 palabra2 quien eres, palabra3 palabra4 palabra5 palabra6";
echo "$text <br/><br/>"; foreach($swear as $key=> $value){
```

Centro de Formación, Investigación y Desarrollo de Soluciones de e-Learning.

UTN - FRBA. Secretaría de Cultura y Extensión Universitaria

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148 // e-learning@sceu.frba.utn.edu.ar



```
$text = str_replace("$value", "*****", $text);  
}  
echo "$text";  
?>
```

Otro ejemplo:

```
<?php
```

```
$cadena = 'HTML5 es la quinta revision del lenguaje basico de la www. Las  
especificaciones de HTML5 estan dadas por la W3C';  
$buscar = 'HTML5';  
$reemplazo = '____';  
  
echo str_replace($buscar, $reemplazo, $cadena);  
?>
```

Este ejemplo producirá el siguiente resultado:

\_\_\_\_ es la quinta revision del lenguaje basico de la www. Las especificaciones de \_\_\_\_  
estan dadas por la W3C

Estas son algunas de las funciones de string que PHP nos proporciona para trabajar con cadenas, y que consideramos útiles, hay algunas otras. Para conocer todas las funciones existentes o para consultar mas ejemplos pueden consultar <http://www.php.net/manual/es/ref.strings.php>