



**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 MATEMATICAS



## Funciones matemáticas.



## Presentación de la Unidad:

**En esta unidad veremos algunas de las funciones matemáticas que brinda PHP.**



## Objetivos:

- ❖ **Aprender a utilizar algunas funciones que nos serán de utilidad a la hora de trabajar con variables numéricas para poder operar con ellas.**



## Temario:

*Introducción*

*Función ceil*

*Función floor*

*Función round*

*Función fmod*

*Función is\_nan*

*Función rand*

*Función getrandmax*

*Función max*

*Función min*



## 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 MATEMATICAS.**

### *Introducción*

- `abs` — Valor absoluto
- `acos` — Arco coseno
- `acosh` — Arco coseno hiperbólico
- `asin` — Arco seno
- `asinh` — Arco seno hiperbólico
- `atan2` — Arco tangente de dos variables
- `atan` — Arco tangente
- `atanh` — Arco tangente hiperbólica
- `base_convert` — Convertir un número entre bases arbitrarias
- `bindec` — Binario a decimal
- `ceil` — Redondear fracciones hacia arriba
- `cos` — Coseno
- `cosh` — Coseno hiperbólico
- `decbin` — Decimal a binario
- `dechex` — Decimal a hexadecimal
- `decoct` — Decimal a octal
- `deg2rad` — Convierte el número en grados a su equivalente en radianes
- `exp` — Calcula la exponencial de e
- `expm1` — Devuelve  $\exp(\text{numero})-1$ , calculado de tal forma que no pierde precisión incluso cuando el valor del numero se aproxima a cero.
- `floor` — Redondear fracciones hacia abajo
- `fmod` — Devuelve el residuo de punto flotante (módulo) de la división de los argumentos
- `getrandmax` — Mostrar el mayor valor aleatorio posible
- `hexdec` — Hexadecimal a decimal
- `hypot` — Calcula la longitud de la hipotenusa de un triángulo de ángulo recto
- `is_finite` — Encuentra si un valor es un número finito legal
- `is_infinite` — Encuentra si un valor es infinito
- `is_nan` — Encuentra si un valor no es un número
- `lcg_value` — Generador lineal congruente combinado
- `log10` — Logaritmo en base 10
- `log1p` — Devuelve  $\log(1 + \text{numero})$ , calculado de tal forma que no pierde precisión incluso cuando el valor del numero se aproxima a cero.
- `log` — Logaritmo natural
- `max` — Encontrar el valor más alto
- `min` — Encontrar el valor más bajo
- `mt_getrandmax` — Mostrar el mayor valor aleatorio posible
- `mt_rand` — Genera un mejor número entero aleatorio
- `mt_srand` — Genera un mejor número entero aleatorio a partir de una semilla

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)





- octdec — Octal a decimal
- pi — Obtener valor de pi
- pow — Expresión exponencial
- rad2deg — Convierte el número en radianes a su equivalente en grados
- rand — Genera un número entero aleatorio
- round — Redondea un float
- sin — Seno
- sinh — Seno hiperbólico
- sqrt — Raíz cuadrada
- srand — Genera un número entero aleatorio a partir de una semilla
- tan — Tangente
- tanh — Tangente hiperbólica

Esta es la lista de todas las funciones matemáticas que PHP tiene disponibles para nosotros. Pueden encontrar información sobre las mismas en el manual de PHP: <http://php.net/manual/es/ref.math.php>

Nosotros veremos algunas de ellas que consideramos importantes, pero en realidad todas lo son, lo importante es saber que existen y que podemos recurrir a ellas cuando tengamos la necesidad. Cuando tengamos que resolver algo entonces lo que tenemos que hacer es recurrir a la lista de funciones y ver cual de ellas es la apropiada para ayudarme a resolver el problema. Allí encontrarán la información necesaria como parámetros de entrada, que es lo que devuelve y como se utilizan, en la mayoría de los casos hay ejemplos de su uso.

### Función ceil

Redondear fracciones hacia arriba.

Descripción

float ceil ( float \$value )

Devuelve el siguiente valor entero mayor redondeando hacia arriba value si fuera necesario.

Parámetros

Value: El valor a redondear

Valores devueltos

value redondeado al siguiente entero más alto. El valor de retorno de ceil() sigue siendo de tipo float ya que el rango de valores de float es usualmente mayor que el del tipo integer.

Ejemplos

<?php

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)



```
echo ceil(4.3); // 5  
echo ceil(9.999); // 10  
echo ceil(-3.14); // -3  
?>
```

### Función floor

Redondear fracciones hacia abajo.

Descripción

float floor ( float \$value )

Devuelve el máximo de los enteros menores o iguales, redondeando value si es necesario.

Parámetros

Value: El valor a redondear

Valores devueltos

value redondeado al anterior entero más bajo. El valor de retorno de floor() sigue siendo de tipo float ya que el rango de valores de float es usualmente mayor que el del tipo integer.

Ejemplos

```
<?php  
echo floor(4.3); // 4  
echo floor(9.999); // 9  
echo floor(-3.14); // -4  
?>
```

### Función round

Redondea un float.

Descripción

float round ( float \$val [, int \$precision = 0 [, int \$mode = PHP\_ROUND\_HALF\_UP ]] )

Devuelve el valor redondeado de val con la precision especificada (número de dígitos después del punto decimal). precision puede también ser negativo o cero (valor predeterminado).

Parámetros

val: El valor a redondear

precision: Opcionalmente, el número de dígitos decimales a redondear.

mode :

Use una de las siguientes constantes para especificar el modo de redondeo.

Constante

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

### Descripción

**PHP\_ROUND\_HALF\_UP** Redondea val hacia arriba a precision lugares decimales alejándose de cero, cuando está a medio camino. Hace que 1.5 sea 2, y -1.5 sea -2.

**PHP\_ROUND\_HALF\_DOWN** Redondea val hacia abajo a precision lugares decimales hacia cero, cuando está a medio camino. Hace que 1.5 sea 1, y -1.5 sea -1.

**PHP\_ROUND\_HALF\_EVEN** Redondea val a precision lugares decimales hacia el siguiente valor par.

**PHP\_ROUND\_HALF\_ODD** Redondea val a precision lugares decimales hacia el siguiente valor impar.

### Valores devueltos

El valor redondeado

### Ejemplos

#### Ejemplo #1 Ejemplos de round()

```
<?php
echo round(3.4); // 3
echo round(3.5); // 4
echo round(3.6); // 4
echo round(3.6, 0); // 4
echo round(1.95583, 2); // 1.96
echo round(1241757, -3); // 1242000
echo round(5.045, 2); // 5.05
echo round(5.055, 2); // 5.06
?>
```

#### Ejemplo #2 Ejemplos de mode

```
<?php
echo round(9.5, 0, PHP_ROUND_HALF_UP); // 10
echo round(9.5, 0, PHP_ROUND_HALF_DOWN); // 9
echo round(9.5, 0, PHP_ROUND_HALF_EVEN); // 10
echo round(9.5, 0, PHP_ROUND_HALF_ODD); // 9
```

```
echo round(8.5, 0, PHP_ROUND_HALF_UP); // 9
echo round(8.5, 0, PHP_ROUND_HALF_DOWN); // 8
echo round(8.5, 0, PHP_ROUND_HALF_EVEN); // 8
echo round(8.5, 0, PHP_ROUND_HALF_ODD); // 9
?>
```

#### Ejemplo #3 Ejemplos de mode con precisión

```
<?php
/* Usar PHP_ROUND_HALF_UP com precisión de 1 dígito decimal */
echo round( 1.55, 1, PHP_ROUND_HALF_UP); // 1.6
echo round( 1.54, 1, PHP_ROUND_HALF_UP); // 1.5
echo round(-1.55, 1, PHP_ROUND_HALF_UP); // -1.6
echo round(-1.54, 1, PHP_ROUND_HALF_UP); // -1.5

/* Usar PHP_ROUND_HALF_DOWN com precisión de 1 dígito decimal */
```

```
echo round( 1.55, 1, PHP_ROUND_HALF_DOWN); // 1.5
echo round( 1.54, 1, PHP_ROUND_HALF_DOWN); // 1.5
echo round(-1.55, 1, PHP_ROUND_HALF_DOWN); // -1.5
echo round(-1.54, 1, PHP_ROUND_HALF_DOWN); // -1.5

/* Usar PHP_ROUND_HALF_EVEN com precisión de 1 dígito decimal */
echo round( 1.55, 1, PHP_ROUND_HALF_EVEN); // 1.6
echo round( 1.54, 1, PHP_ROUND_HALF_EVEN); // 1.5
echo round(-1.55, 1, PHP_ROUND_HALF_EVEN); // -1.6
echo round(-1.54, 1, PHP_ROUND_HALF_EVEN); // -1.5

/* Usar PHP_ROUND_HALF_ODD com precisión de 1 dígito decimal */
echo round( 1.55, 1, PHP_ROUND_HALF_ODD); // 1.5
echo round( 1.54, 1, PHP_ROUND_HALF_ODD); // 1.5
echo round(-1.55, 1, PHP_ROUND_HALF_ODD); // -1.5
echo round(-1.54, 1, PHP_ROUND_HALF_ODD); // -1.5
?>
```

### Función fmod

Devuelve el resto en punto flotante (módulo) de la división de los argumentos.

Descripción

float fmod ( float \$x , float \$y )

Devuelve el resto de punto flotante de dividir el dividendo (x) por el divisor (y). El resto (r) es definido como:  $x = i * y + r$ , para algún entero i. Si y es diferente de cero, r tiene el mismo signo que x y una magnitud menor que la magnitud de y.

Parámetros

x : El dividendo

y : El divisor

Valores devueltos

El resto en punto flotante de x/y

Ejemplos

Ejemplo #1 Uso de fmod()

```
<?php
$x = 5.7;
$y = 1.3;
$r = fmod($x, $y);
// $r es igual a 0.5, ya que  $4 * 1.3 + 0.5 = 5.7$ 
?>
```

### Función is\_nan

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



Encuentra si un valor no es un número.

Descripción

bool is\_nan ( float \$val )

Verifica si val no es un numero, como por ejemplo el resultado de acos(1.01).

Parámetros

val : El valor a revisar

Valores devueltos

Devuelve TRUE si val no es un número, en caso contrario devuelve FALSE.

Ejemplos

Ejemplo #1 Ejemplo de is\_nan()

```
<?php
```

```
// Invalid calculation, will return a
```

```
// NaN value
```

```
$nan = acos(8);
```

```
var_dump($nan, is_nan($nan));
```

```
?>
```

El resultado del ejemplo sería:

float(NAN)

bool(true)

### Función rand

Genera un número entero aleatorio

Descripción

int rand ( void )

int rand ( int \$min , int \$max )

Si se invoca sin los argumentos opcionales min y max, rand() devuelve un entero pseudoaleatorio entre 0 y getrandmax(). Para obtener un número aleatorio entre 5 y 15 (incluidos), por ejemplo, use rand(5, 15).

Nota: En algunas plataformas (como en Windows), getrandmax() sólo alcanza hasta 32767. En caso de necesitar un valor mayor de 32767, se deberá especificar min y max para crear un valor mayor que este, o considere emplear mt\_rand() en su lugar.

Parámetros

min: El menor valor a devolver (por defecto: 0)

max: El mayor valor a devolver (por defecto: getrandmax())

Valores devueltos

Un valor pseudoaleatorio entre min (o 0) y max (o getrandmax()), incluidos).

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



## Ejemplos

### Ejemplo #1 Ejemplo de rand()

```
<?php  
echo rand() . "\n";  
echo rand() . "\n";
```

```
echo rand(5, 15);  
?>
```

## Función getrandmax

Mostrar el mayor valor aleatorio posible

Descripción

int getrandmax ( void )

Devuelve el valor máximo que puede ser devuelto por una llamada a rand().

Valores devueltos

El valor aleatorio más grande posible devuelto por rand()

## Función max

Encontrar el valor más alto

Descripción

mixed max ( array \$values )

mixed max ( mixed \$value1 , mixed \$value2 [, mixed \$... ] )

Si el primer y único parámetro es una matriz, max() devuelve el valor más alto en esa matriz. Si al menos dos parámetros son entregados, max() devuelve el mayor de estos valores.

Nota:

PHP evaluará un valor string no-numérico como 0 si se compara con un integer, pero aun devuelve la cadena si ésta es vista como el valor numérico más alto. Si varios argumentos evalúan a 0, max() devolverá el valor de cadena alfanumérica más alto si se da alguna cadena, o de lo contrario se devuelve un 0 numérico.

Parámetros

values

Un array que contiene los valores.

value1

Cualquier valor comparable.

value2

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

Cualquier valor comparable.

...

Cualquier valor comparable.

Valores devueltos

max() devuelve el mayor valor numérico de los parámetros. Si varios valores pueden considerarse del mismo tamaño, se devolverá el listado en primer lugar.

Cuando max() debe evaluar varios arrays, devuelve el array de mayor tamaño. Si todos los arrays tienen el mismo tamaño, max() empleará la ordenación lexicográfica para encontrar el valor a devolver.

Cuando se debe comparar un string, se le considerará como un integer en la comparación.

Ejemplos

Ejemplo #1 Ejemplos de uso de max()

```
<?php  
echo max(1, 3, 5, 6, 7); // 7  
echo max(array(2, 4, 5)); // 5
```

```
// Cuando 'hola' es convertido en integer se convierte en 0. Ambos parámetros son  
// del mismo tamaño por lo que el orden en el que son colocados determina el  
resultado
```

```
echo max(0, 'hola'); // 0  
echo max('hola', 0); // hola
```

```
echo max('42', 3); // '42'
```

```
// En este caso 0 > -1, así que 'hola' es el valor devuelto.
```

```
echo max(-1, 'hola'); // hola
```

```
?>
```

### **Función min**

Encontrar el valor más bajo

Descripción

mixed min ( array \$values )

mixed min ( mixed \$value1 , mixed \$value2 [, mixed \$... ] )

Si el primer y único parámetro es un array, min() devuelve el valor más bajo de ese array. Si se proporcionan al menos dos parámetros, min() devuelve el menor de estos valores.

Nota:

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

PHP evaluará un valor string no numérico como 0 si se compara con un integer, pero aún devuelve la cadena si ésta es vista como el valor numérico más bajo. Si varios argumentos se evalúan a 0, min() devolverá el valor de cadena alfanumérica más bajo si se da alguna cadena, o de lo contrario se devuelve un 0 numérico.

Parámetros

valores

Un array que contiene los valores.

value1

Cualquier valor comparable.

value2

Cualquier valor comparable.

...

Cualquier valor comparable.

Valores devueltos

min() devuelve el valor numéricamente más bajo de los parámetros.

Ejemplos

Ejemplo #1 Ejemplos de uso de min()

```
<?php
```

```
echo min(2, 3, 1, 6, 7); // 1
```

```
echo min(array(2, 4, 5)); // 2
```

```
echo min(0, 'hola'); // 0
```

```
echo min('hola', 0); // hola
```

```
echo min('hola', -1); // -1
```

```
?>
```

### *Algunos ejemplos prácticos en donde usamos algunas funciones*

Los ejemplos están en la carpeta ejemplos de la presente unidad.

Vamos a repasar el código de los mismos para entender que hacen.

Los archivos genera\_contasena.php y genera\_contrasena\_1.php son dos maneras diferentes de generar una contraseña aleatoriamente.

En el primer caso lo que hacemos es almacenar en la variable del tipo array alfabeto todos los caracteres que queremos que formen parte de nuestra contraseña, las letras de la a a la z. Y luego con la función rand() lo que hacemos es calcular un número al azar que será utilizado como subíndice del array para ir concatenando las letras en forma aleatoria.

Luego concatenamos un número aleatorio entre 1000 y 9999. De esta forma nos queda una contraseña generada al azar compuesta de 4 letras y cuatro dígitos.

**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



Como es sabido, en programación no hay una única manera de hacer las cosas, dos personas podemos llegar al mismo resultado por caminos diferentes. Es decir que esto mismo podríamos haberlo conseguido con una lógica diferente.

En el script `genera_contraseña_1.php` tenemos otra alternativa de generar la misma contraseña. En este caso combinamos el uso de la función `rand()` definida en las funciones matemáticas de PHP con la función `chr` definida en las funciones de Sting que lo que hace es devolver el carácter correspondiente al valor ASCII del mismo.

Como podemos ver entre el nro 65 y 90 se encuentran las letras de la A a la Z y entre el 97 y 122 se encuentran las letras minúsculas.

**TABLA DE CARACTERES DEL CÓDIGO ASCII**

|    |    |    |    |     |     |     |     |     |     |                   |
|----|----|----|----|-----|-----|-----|-----|-----|-----|-------------------|
| 1  | 25 | 49 | 73 | 97  | 121 | 145 | 169 | 193 | 217 | 241               |
| 2  | 26 | 50 | 74 | 98  | 122 | 146 | 170 | 194 | 218 | 242               |
| 3  | 27 | 51 | 75 | 99  | 123 | 147 | 171 | 195 | 219 | 243               |
| 4  | 28 | 52 | 76 | 100 | 124 | 148 | 172 | 196 | 220 | 244               |
| 5  | 29 | 53 | 77 | 101 | 125 | 149 | 173 | 197 | 221 | 245               |
| 6  | 30 | 54 | 78 | 102 | 126 | 150 | 174 | 198 | 222 | 246               |
| 7  | 31 | 55 | 79 | 103 | 127 | 151 | 175 | 199 | 223 | 247               |
| 8  | 32 | 56 | 80 | 104 | 128 | 152 | 176 | 200 | 224 | 248               |
| 9  | 33 | 57 | 81 | 105 | 129 | 153 | 177 | 201 | 225 | 249               |
| 10 | 34 | 58 | 82 | 106 | 130 | 154 | 178 | 202 | 226 | 250               |
| 11 | 35 | 59 | 83 | 107 | 131 | 155 | 179 | 203 | 227 | 251               |
| 12 | 36 | 60 | 84 | 108 | 132 | 156 | 180 | 204 | 228 | 252               |
| 13 | 37 | 61 | 85 | 109 | 133 | 157 | 181 | 205 | 229 | 253               |
| 14 | 38 | 62 | 86 | 110 | 134 | 158 | 182 | 206 | 230 | 254               |
| 15 | 39 | 63 | 87 | 111 | 135 | 159 | 183 | 207 | 231 | 255               |
| 16 | 40 | 64 | 88 | 112 | 136 | 160 | 184 | 208 | 232 | PRESIONA LA TECLA |
| 17 | 41 | 65 | 89 | 113 | 137 | 161 | 185 | 209 | 233 | Alt               |
| 18 | 42 | 66 | 90 | 114 | 138 | 162 | 186 | 210 | 234 | MÁS EL NÚMERO     |
| 19 | 43 | 67 | 91 | 115 | 139 | 163 | 187 | 211 | 235 | CORTESÍA DE:      |
| 20 | 44 | 68 | 92 | 116 | 140 | 164 | 188 | 212 | 236 | REDEC             |
| 21 | 45 | 69 | 93 | 117 | 141 | 165 | 189 | 213 | 237 | desde 1978        |
| 22 | 46 | 70 | 94 | 118 | 142 | 166 | 190 | 214 | 238 |                   |
| 23 | 47 | 71 | 95 | 119 | 143 | 167 | 191 | 215 | 239 |                   |
| 24 | 48 | 72 | 96 | 120 | 144 | 168 | 192 | 216 | 240 |                   |

Entonces este sería otra manera de llegar al mismo resultado por un camino diferente y como estas se les pueden ocurrir mil variantes. Pueden pensar en la que les resulte mas conveniente.

Incluso podrían modificar las condiciones de generación de la contraseña e ir modificando el código según sus necesidades.

El otro ejemplo práctico de uso de funciones lo tenemos en el archivo `paginado.php` y lo que estamos intentando hacer es un paginado de resultados.

Supongamos que queremos mostrar las características de nuestros productos que tenemos almacenados en la base de datos y que disponemos de 84 productos pero no queremos mostrar los 84 en la misma página sino que los vamos a mostrar de a 10 por

página, en ese caso necesitaremos 9 páginas para mostrarlos, las primeras 8 páginas contendrán 10 productos y la última contendrá 4 productos.

Para resolver esto podemos contar la cantidad total de productos que tenemos en la tabla y dividirla por la cantidad de productos que queremos mostrar por página. En nuestro ejemplo  $84/10$  nos da 8.4, pero como no podemos representar 8.4 páginas, aplicamos la función `ceil` a ese valor que nos devuelve 9. Tenemos entonces lo que necesitamos, 9 páginas, de las cuales las primeras 8 contendrán 10 productos y la última 4.

El código del ejercicio es este:

```
<?php
$total_productos = 84;
$total_prod_x_pag = 10;
$total_pags = ceil($total_productos / $total_prod_x_pag);

for ($i=0;$i<$total_pags;$i++){
    $proddesde = $i * $total_prod_x_pag;
    $nro_pag = $i + 1;
    echo '<a href="paginado.php?desde='.$proddesde.'">['.$nro_pag.']/> - ';
}

if (isset($_GET['desde'])){
    $pag = ($_GET['desde'] + $total_prod_x_pag) / $total_prod_x_pag;
    echo 'Página '.$pag.' de '.$total_pags;
}
else{
    echo 'Página 1 de '.$total_pags;
}

?>
```

Aquí estamos moviendo fijos los valores del total de productos, si lo trabajáramos con una tabla sql, en esa variable tendríamos que almacenar el valor resultado de ejecutar la sentencia: “SELECT COUNT(\*) FROM productos”.

La consulta de productos se tendría que hacer dependiendo la página con una cláusula `LIMIT prod_desde, cantidad_productos`. En nuestro ejemplo `cantidad_productos` sería 10, la cantidad de productos a mostrar por página.

Ya vamos a ver mas adelante como realizarlo en forma completa con el uso de las tablas sql. Por ahora nos centramos en la lógica y en el uso de la función `ceil`.

Para darle un poco de diseño a nuestro paginado vamos a utilizar Bootstrap tomando

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



uno de los componentes que nos ofrece este framework para hacer paginación de resultados. Les dejo el link al sitio oficial de Bootstrap donde habla sobre este componente: <http://getbootstrap.com/components/#pagination>

Utilizando este diseño adaptamos el ejercicio anterior. El código lo encuentran comentado en paginadoBootstrap.php