

# Ejercicios PHP 2

## 1. Contador de Palabras

Crea un programa que cuente el número de palabras en una cadena de texto ingresada por el usuario.

- Pedir al usuario que ingrese una frase.
  - Inicializar un contador de palabras en 0.
  - Recorrer cada carácter de la frase.
  - Si el carácter es un espacio, incrementar el contador de palabras.
  - Al finalizar, sumar 1 al contador de palabras (para contar la última palabra).
- Imprimir el número total de palabras.

```
<?php
error_reporting(E_ALL);
//error_reporting(0);

$contador = 0;
$frase = readline("Introduzca la  cadena de texto: ");

for ($i = 0; $i < strlen($frase); $i++) { // Cuenta cuantos espacios hay
en la cadena de texto
    switch($frase[$i]) {

        case " ": $contador++;
        break;

        default:
        break;
    }
}

$contador++;

echo ("La cadena de texto tiene $contador palabras") // Indica cuantas
palabras hay en la cadena de texto

?>
```

```
PS C:\Users\CampusFP> cd c:\xampp\htdocs\programacion\Ejercicio2
PS C:\xampp\htdocs\programacion\Ejercicio2> php 1.php
Introduzca la  cadena de texto: Hola mundo
La cadena de texto tiene 2 palabras
PS C:\xampp\htdocs\programacion\Ejercicio2> 
```

## 2. Ordenamiento de array

Crea un programa que ordene un array de números de forma ascendente.

- Crear un array de números aleatorios.
- Utilizar un doble bucle para comparar cada elemento con los demás y realizar intercambios si están en el orden incorrecto (algoritmo de burbuja).
- Imprimir el array ordenado.

```
<?php
error_reporting(E_ALL);
//error_reporting(0);

$array = array();

$numero_valores = 5; // Indica la cantidad de valores que tendrá el array

for ($x = 0; $x < $numero_valores; $x++) { // Introduce valores
    aleatorios en el array

        $numero_aleatorio = rand(1,100);
        array_push($array,$numero_aleatorio);
}

echo("-Array desordenado")."\n"; // Muestra el array desordenado
for ($x = 0; $x < count($array); $x++) {
    echo $array[$x]."\n";
}

for ($y = 0; $y < count($array); $y++) { // Ordena el array

    for ($z = 0; $z < count($array); $z++) {

        if ($array[$z] > $array[$y]) { // Si uno de los números es mayor
que otro intercambian sus posiciones
            $puente = $array[$z];
            $array[$z] = $array[$y];
            $array[$y] = $puente;
        }
    }
}

echo("-Array ordenado")."\n"; // Muestra el array ordenado
for ($x = 0; $x < count($array); $x++) {

    echo $array[$x]."\n";
}
```

?>

```
▼ TERMINAL
PS C:\xampp\htdocs\programacion\Ejercicio2> php 2.php
-Array desordenado
97
97
83
29
10
50
-Array ordenado
10
29
50
83
97
PS C:\xampp\htdocs\programacion\Ejercicio2> 
```

### 3. Validación de contraseña

Crea un programa que valide si una contraseña cumple con ciertos criterios.

Pedir al usuario que ingrese una contraseña.

Verificar si la contraseña tiene al menos 8 caracteres.

Verificar si la contraseña contiene al menos una letra mayúscula, una minúscula y un número.

Utilizar expresiones regulares para realizar estas verificaciones.

Si la contraseña cumple con todos los criterios, mostrar un mensaje de éxito.

```
<?php
error_reporting(E_ALL);
//error_reporting(0);

$contador = 0;
$patron1 = "/[[:alnum:]]{8,}/";
$patron2 = "/[[:digit:]]/";
$patron3 = "/[a-z]/";
$patron4 = "/[A-Z]/";
$clave = readline("Introduzca la contraseña: ");

if (preg_match($patron1, $clave)) { // Comprueba el número de caracteres
    de la contraseña

    echo("La contraseña tiene 8 o más caracteres")."\n";
    $contador++;
}
```

```
} else {  
  
    echo("La contraseña tiene menos de 8 caracteres")."\n";  
  
}  
  
if (preg_match($patron2, $clave)) { // Comprueba si la contraseña incluye  
números  
  
    echo("La contraseña contiene números")."\n";  
    $contador++;  
  
} else {  
  
    echo("La contraseña no contiene números")."\n";  
  
}  
  
if (preg_match($patron3, $clave)) { // Comprueba si la contraseña incluye  
minúsculas  
  
    echo("La contraseña contiene minúsculas")."\n";  
    $contador++;  
  
} else {  
  
    echo("La contraseña no contiene minúsculas")."\n";  
  
}  
  
if (preg_match($patron4, $clave)) { // Comprueba si la contraseña incluye  
mayúsculas  
  
    echo("La contraseña contiene mayúsculas")."\n";  
    $contador++;  
  
} else {  
  
    echo("La contraseña no contiene mayúsculas")."\n";  
  
}  
  
if ($contador == 4) { // Indica si la contraseña cumple todas las  
condiciones  
  
    echo ("La contraseña es válida");  
  
} else {
```

```

    echo ("La contraseña no es válida");
}

?>

```

▼ TERMINAL

```

PS C:\xampp\htdocs\programacion\Ejercicio2> php 3.php
Introduzca la contraseña: 446DYTfjrtej3W
La contraseña tiene 8 o más dígitos
La contraseña contiene números
La contraseña contiene minúsculas
La contraseña contiene mayúsculas
La contraseña es válida
PS C:\xampp\htdocs\programacion\Ejercicio2> php 3.php
Introduzca la contraseña: gehg
La contraseña tiene menos de 8 dígitos
La contraseña no contiene números
La contraseña contiene minúsculas
La contraseña no contiene mayúsculas
La contraseña no es válida
PS C:\xampp\htdocs\programacion\Ejercicio2> 

```

## 4. Generador de nombres aleatorios

Crea un programa que genere nombres aleatorios a partir de listas de nombres y apellidos.

Crear dos arrays, uno con nombres y otro con apellidos.

Generar un número aleatorio para seleccionar un nombre y otro para seleccionar un apellido.

Concatenar el nombre y el apellido para formar un nombre completo aleatorio.

```

<?php
error_reporting(E_ALL);
//error_reporting(0);

$nombres = array( //Posibles nombres
    1 => "Juan ",
    2 => "Raúl ",
    3 => "María ",
    4 => "David ",
    5 => "Luna ",
);

$apellidos = array( //Posibles apellidos
    1 => "Pérez ",
    2 => "Morales ",
    3 => "Casas ",
    4 => "García ",

```

```

        5 => "Guijo ",
    );

    $nombre_random = rand(1, count($nombres)); //Elije uno de los nombres al
    azar
    $apellido_random = rand(1, count($apellidos)); //Elije uno de los
    apellidos al azar

    echo("-Nombres: ")."\n";
    for ($x = 1; $x <= count($nombres); $x++) { //Imprime los nombres

        echo $nombres[$x]."\n";

    }

    echo("-Apellidos: ")."\n";
    for ($x = 1; $x <= count($apellidos); $x++) { //Imprime los apellidos

        echo $apellidos[$x]."\n";

    }

    echo("-Nombre aleatorio:")."\n";
    $nombre_completo = $nombres[$nombre_random] .
    $apellidos[$apellido_random]; // Crea el nombre completo

    echo $nombre_completo //Imprime el nombre completo
    ?>

```

#### ▼ TERMINAL

```

PS C:\xampp\htdocs\programacion\Ejercicio2> php 4.php
-Nombres:
Juan
Raúl
María
David
Luna
-Apellidos:
Pérez
Morales
Casas
García
Guijo
-Nombre aleatorio:
Raúl Casas
PS C:\xampp\htdocs\programacion\Ejercicio2> 

```

## 5. Simulador de dado

Crea un programa que simule el lanzamiento de un dado de seis caras.

Utilizar la función `rand()` para generar un número aleatorio entre 1 y 6.

Imprimir el número obtenido, simulando el resultado del lanzamiento del dado.

```
<?php
error_reporting(E_ALL);
//error_reporting(0);

echo("Lanzas un dado")."\n";

$numero = rand(1, 6); //Escoge un número entre 1 y 6

echo("Ha salido el número: $numero"); //Muestra el número

?>
```

▼ TERMINAL

```
PS C:\xampp\htdocs\programacion\Ejercicio2> php 5.php
Lanzas un dado
Ha salido el número: 6
PS C:\xampp\htdocs\programacion\Ejercicio2> █
```

<https://github.com/Rodrigo-Garcia-Ortiz/Programacion.git>