

2º ASIR

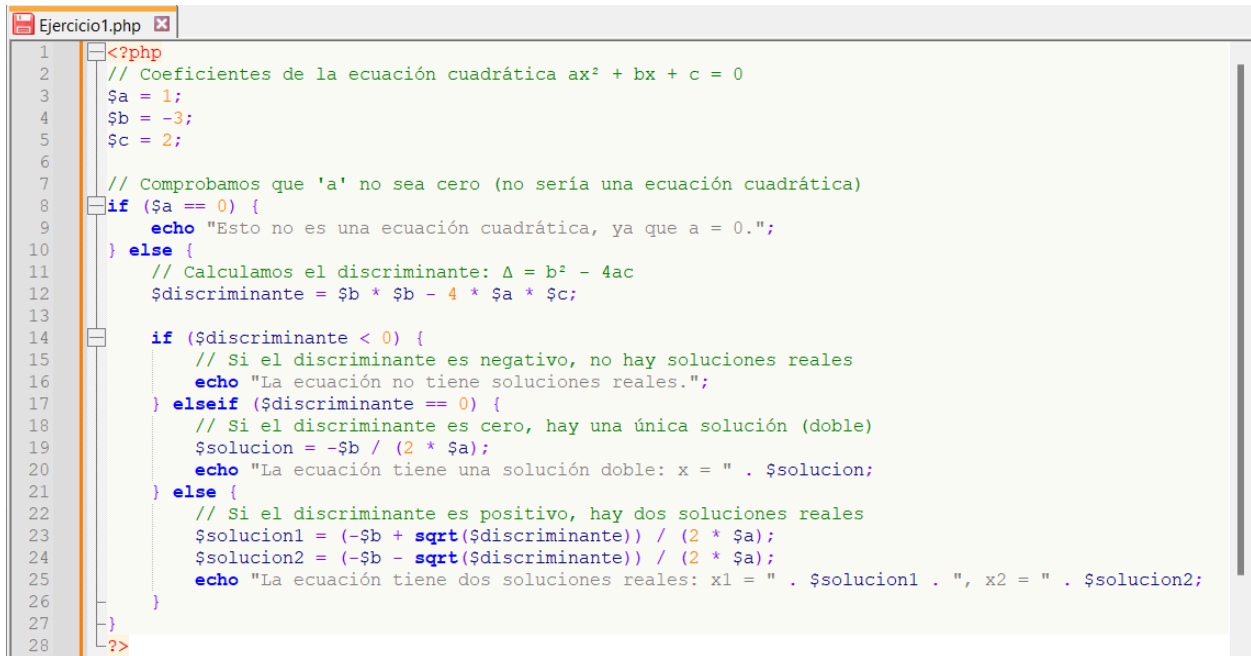
Implantación de aplicaciones Web

Lara Meneses Hernández

Práctica - Capítulo 2 Sintaxis



1. Escribe un script para resolver ecuaciones de segundo grado, $ax^2 + bx + c = 0$. Si la ecuación no tiene soluciones reales hay que mostrar un mensaje de error:

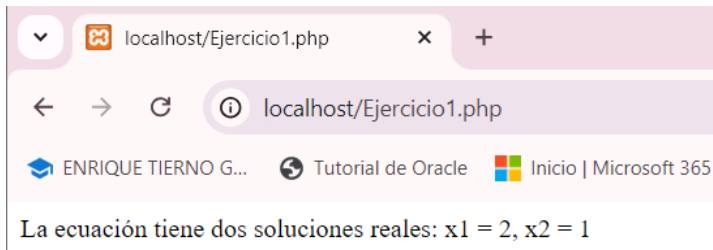


```
1 <?php
2 // Coeficientes de la ecuación cuadrática ax² + bx + c = 0
3 $a = 1;
4 $b = -3;
5 $c = 2;
6
7 // Comprobamos que 'a' no sea cero (no sería una ecuación cuadrática)
8 if ($a == 0) {
9     echo "Esto no es una ecuación cuadrática, ya que a = 0.";
10 } else {
11     // Calculamos el discriminante: Δ = b² - 4ac
12     $discriminante = $b * $b - 4 * $a * $c;
13
14     if ($discriminante < 0) {
15         // Si el discriminante es negativo, no hay soluciones reales
16         echo "La ecuación no tiene soluciones reales.";
17     } elseif ($discriminante == 0) {
18         // Si el discriminante es cero, hay una única solución (doble)
19         $solucion = -$b / (2 * $a);
20         echo "La ecuación tiene una solución doble: x = " . $solucion;
21     } else {
22         // Si el discriminante es positivo, hay dos soluciones reales
23         $solucion1 = (-$b + sqrt($discriminante)) / (2 * $a);
24         $solucion2 = (-$b - sqrt($discriminante)) / (2 * $a);
25         echo "La ecuación tiene dos soluciones reales: x1 = " . $solucion1 . ", x2 = " . $solucion2;
26     }
27 }
28 ?>
```

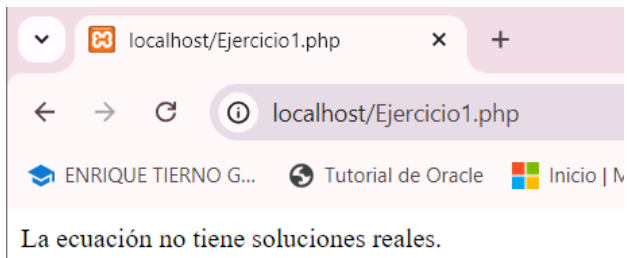
Explicación:

1. **Variables:** Los coeficientes a , b , y c se asignan directamente en el código.
2. **Discriminante:** Se calcula el discriminante $\Delta = b^2 - 4ac$.
3. **Soluciones:**
 - Si el discriminante es negativo, se muestra un mensaje indicando que no hay soluciones reales.
 - Si es cero, se calcula la solución doble.
 - Si es positivo, se calculan las dos soluciones reales.
4. **Condicionales:** El script usa condicionales `if-else` para determinar qué tipo de solución mostrar.

Este script resolverá la ecuación directamente cuando se ejecute. Puedes cambiar los valores de a , b , y c según la ecuación que quieras resolver.



Modificamos el valor de las variables para que nos muestre el mensaje “La ecuación no tiene números reales”.



2. Crea una función para resolver la ecuación de segundo grado. Esta función recibe los coeficientes de la ecuación y devuelve un array con las soluciones. Si no hay soluciones reales, devuelve FALSE.

```

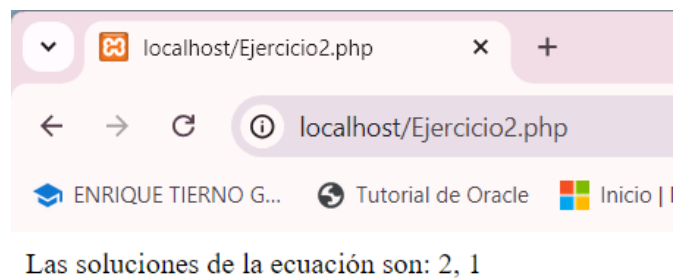
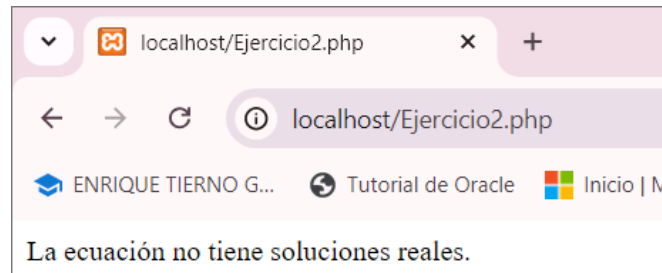
1  <?php
2  function resolverEcuacionCuadratica($a, $b, $c) {
3      // Comprobamos que 'a' no sea cero (no sería una ecuación cuadrática)
4      if ($a == 0) {
5          return FALSE; // No es una ecuación cuadrática
6      }
7
8      // Calculamos el discriminante:  $\Delta = b^2 - 4ac$ 
9      $discriminante = $b * $b - 4 * $a * $c;
10
11     if ($discriminante < 0) {
12         // Si el discriminante es negativo, no hay soluciones reales
13         return FALSE;
14     } elseif ($discriminante == 0) {
15         // Si el discriminante es cero, hay una única solución (doble)
16         $solucion = -$b / (2 * $a);
17         return [$solucion]; // Devolvemos un array con una solución
18     } else {
19         // Si el discriminante es positivo, hay dos soluciones reales
20         $solucion1 = (-$b + sqrt($discriminante)) / (2 * $a);
21         $solucion2 = (-$b - sqrt($discriminante)) / (2 * $a);
22         return [$solucion1, $solucion2]; // Devolvemos un array con dos soluciones
23     }
24 }
25
26 // Ejemplo de uso:
27 $a = 1;
28 $b = -3;
29 $c = 2;
30 $resultado = resolverEcuacionCuadratica($a, $b, $c);
31
32 if ($resultado === FALSE) {
33     echo "La ecuación no tiene soluciones reales.";
34 } else {
35     echo "Las soluciones de la ecuación son: " . implode(" ", $resultado);
36 }
37
38

```

Explicación:

- Función:** `resolverEcuacionCuadratica($a, $b, $c)`:
 - Calcula el discriminante.
 - Si no hay soluciones reales (discriminante negativo), retorna FALSE.
 - Si hay una solución doble (discriminante cero), devuelve un array con una única solución.
 - Si hay dos soluciones reales (discriminante positivo), devuelve un array con ambas soluciones.
- Uso:**
 - La variable `$resultado` recibe el valor devuelto por la función.
 - Si la función devuelve FALSE, se imprime el mensaje indicando que no hay soluciones reales.
 - Si la función devuelve un array, se imprimen las soluciones con `implode()`.

Este código es un ejemplo genérico y puede adaptarse fácilmente a otros valores de los coeficientes.



3. Almacena la función anterior en el fichero matemáticas. Crea un fichero que la incluya y la utilice.

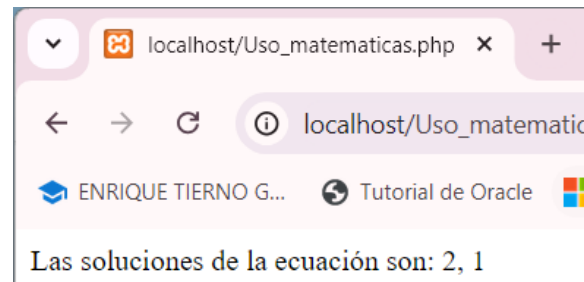
Archivo matematicas.php: contiene la función.

```
matematicas.php
1 <?php
2 // Archivo matematicas.php
3
4 function resolverEcuacionCuadratica($a, $b, $c) {
5     // Comprobamos que 'a' no sea cero (no sería una ecuación cuadrática)
6     if ($a == 0) {
7         return FALSE; // No es una ecuación cuadrática
8     }
9
10    // Calculamos el discriminante:  $\Delta = b^2 - 4ac$ 
11    $discriminante = $b * $b - 4 * $a * $c;
12
13    if ($discriminante < 0) {
14        // Si el discriminante es negativo, no hay soluciones reales
15        return FALSE;
16    } elseif ($discriminante == 0) {
17        // Si el discriminante es cero, hay una única solución (doble)
18        $solucion = -$b / (2 * $a);
19        return [$solucion]; // Devolvemos un array con una solución
20    } else {
21        // Si el discriminante es positivo, hay dos soluciones reales
22        $solucion1 = (-$b + sqrt($discriminante)) / (2 * $a);
23        $solucion2 = (-$b - sqrt($discriminante)) / (2 * $a);
24        return [$solucion1, $solucion2]; // Devolvemos un array con dos soluciones
25    }
26 }
27
28 ?>
```

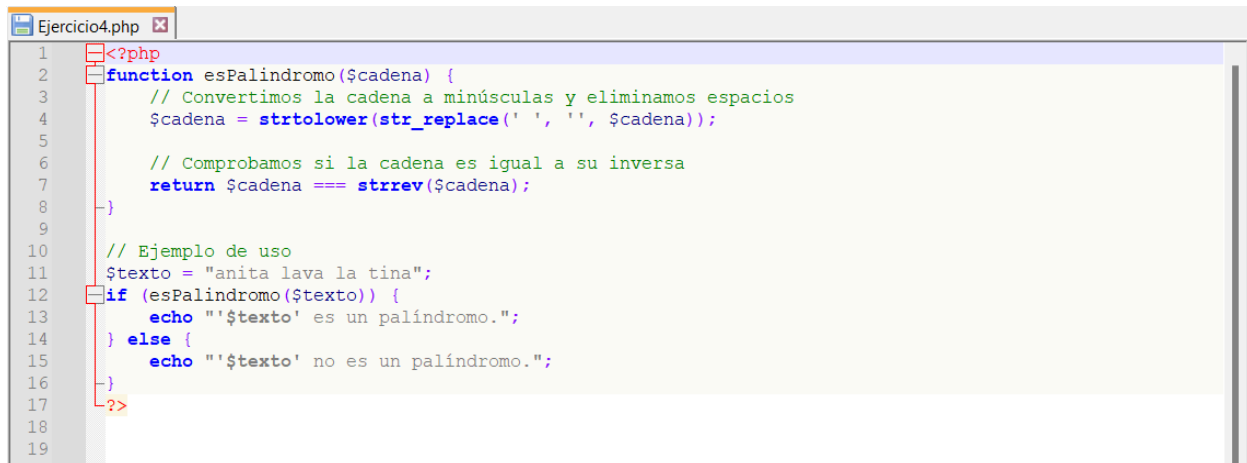
Archivo Uso_matematicas.php: incluye el archivo matematicas.php y utiliza la función definida

```
Uso_matematicas.php
1 <?php
2 // Incluimos el archivo que contiene la función
3 include 'matematicas.php';
4
5 // Ejemplo de uso de la función:
6 $a = 1;
7 $b = -3;
8 $c = 2;
9 $resultado = resolverEcuacionCuadratica($a, $b, $c);
10
11 if ($resultado === FALSE) {
12     echo "La ecuación no tiene soluciones reales.";
13 } else {
14     echo "Las soluciones de la ecuación son: " . implode(" ", $resultado);
15 }
16
17 ?>
```

Comprobamos que funciona:



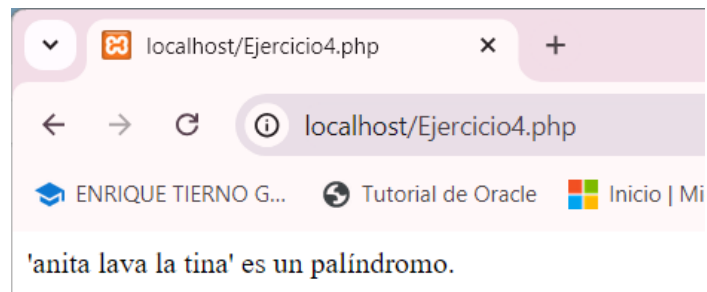
4. Escribe una función que reciba una cadena y compruebe si es un palíndromo.



```
1 <?php
2 function esPalindromo($cadena) {
3     // Convertimos la cadena a minúsculas y eliminamos espacios
4     $cadena = strtolower(str_replace(' ', '', $cadena));
5
6     // Comprobamos si la cadena es igual a su inversa
7     return $cadena === strrev($cadena);
8 }
9
10 // Ejemplo de uso
11 $texto = "anita lava la tina";
12 if (esPalindromo($texto)) {
13     echo "'$texto' es un palíndromo.";
14 } else {
15     echo "'$texto' no es un palíndromo.";
16 }
17 ?>
```

Explicación:

1. **Conversión a minúsculas y eliminación de espacios:** Se usa `strtolower()` para convertir a minúsculas y `str_replace(' ', '', $cadena)` para eliminar los espacios.
2. **Comparación:** Comparamos la cadena con su versión invertida usando `strrev()`.
3. **Retorno:** Devuelve `true` si es un palíndromo y `false` en caso contrario.



localhost/Ejercicio4.php

localhost/Ejercicio4.php

ENRIQUE TIERNO G... Tutorial de Oracle Inicio | Mi

'anita lava la tina' es un palíndromo.

5. Escribe una función que reciba un array de números, y un número, el límite. La función tiene que devolver un nuevo array que contenga solo los elementos del array original menores que el límite.

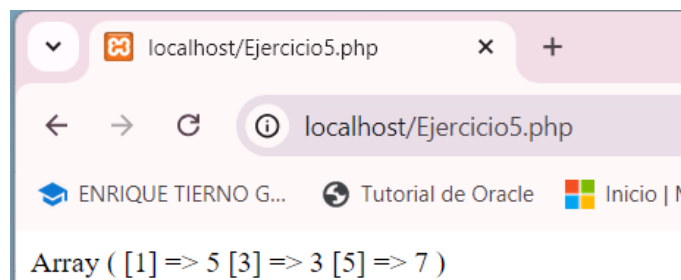
```
Ejercicio5.php
1  <?php
2  function filtrarMenoresQue($array, $limite) {
3      // Usamos array_filter para filtrar los elementos menores que el límite
4      return array_filter($array, function($numero) use ($limite) {
5          return $numero < $limite;
6      });
7  }
8
9  // Ejemplo de uso:
10 $numeros = [10, 5, 20, 3, 15, 7];
11 $limite = 10;
12 $resultado = filtrarMenoresQue($numeros, $limite);
13
14 print_r($resultado); // Muestra los números menores que el límite
15 ?>
16
17
18
```

Explicación:

1. **array_filter()**: Se utiliza para filtrar el array original. Recorre cada elemento y aplica una función de callback.
2. **Función de callback**: La función anónima compara cada número con el límite, devolviendo `true` si el número es menor que el límite y `false` si no lo es. Solo los números que cumplen esta condición se incluyen en el nuevo array.
3. **Uso de use (\$limite)**: Esto permite que la variable `$limite` sea accesible dentro de la función de callback.

Ejemplo:

Con el array [10, 5, 20, 3, 15, 7] y el límite 10, el código devolverá:

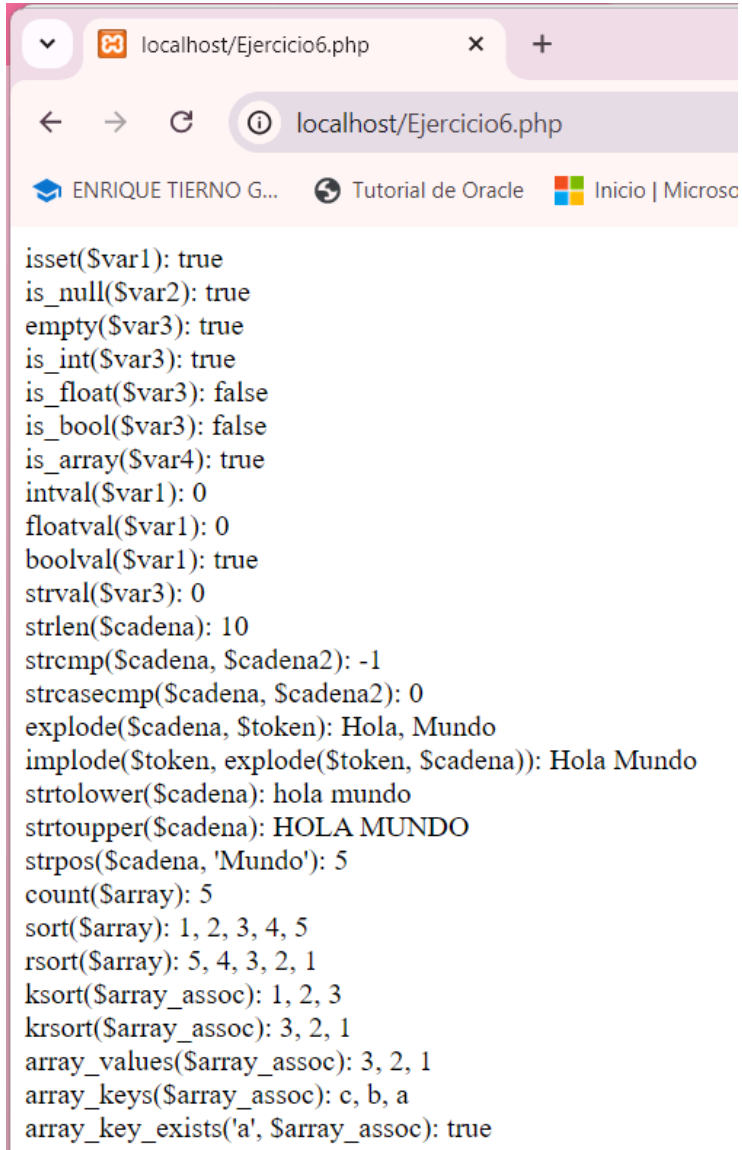


```
localhost/Ejercicio5.php
localhost/Ejercicio5.php
ENRIQUE TIERNO G... Tutorial de Oracle Inicio | M
Array ( [1] => 5 [3] => 3 [5] => 7 )
```

6. Escribe un script para probar las funciones del cuadro 2.6

```
Ejercicio6.php
1 <?php
2 // Funciones de variables
3 $var1 = "Hola";
4 $var2 = null;
5 $var3 = 0;
6 $var4 = array(1, 2, 3);
7
8 echo "isset(\$var1): " . (isset($var1) ? 'true' : 'false') . "<br>\n";
9 echo "is_null(\$var2): " . (is_null($var2) ? 'true' : 'false') . "<br>\n";
10 echo "empty(\$var3): " . (empty($var3) ? 'true' : 'false') . "<br>\n";
11 echo "is_int(\$var3): " . (is_int($var3) ? 'true' : 'false') . "<br>\n";
12 echo "is_float(\$var3): " . (is_float($var3) ? 'true' : 'false') . "<br>\n";
13 echo "is_bool(\$var3): " . (is_bool($var3) ? 'true' : 'false') . "<br>\n";
14 echo "is_array(\$var4): " . (is_array($var4) ? 'true' : 'false') . "<br>\n";
15 echo "intval(\$var1): " . intval($var1) . "<br>\n";
16 echo "floatval(\$var1): " . floatval($var1) . "<br>\n";
17 echo "boolval(\$var1): " . (boolval($var1) ? 'true' : 'false') . "<br>\n";
18 echo "strval(\$var3): " . strval($var3) . "<br>\n";
19
20 // Funciones de cadenas
21 $cadena = "Hola Mundo";
22 $cadena2 = "hola mundo";
23 $token = " ";
24
25 echo "strlen(\$cadena): " . strlen($cadena) . "<br>\n";
26 echo "strcmp(\$cadena, \$cadena2): " . strcmp($cadena, $cadena2) . "<br>\n";
27 echo "strcasecmp(\$cadena, \$cadena2): " . strcasecmp($cadena, $cadena2) . "<br>\n";
28 echo "explode(\$cadena, \$token): " . implode(" ", explode($token, $cadena)) . "<br>\n";
29 echo "implode(\$token, explode(\$token, \$cadena)): " . implode($token, explode($token, $cadena)) . "<br>\n";
30 echo "strtolower(\$cadena): " . strtolower($cadena) . "<br>\n";
31 echo "strtoupper(\$cadena): " . strtoupper($cadena) . "<br>\n";
32 echo "strpos(\$cadena, 'Mundo'): " . strpos($cadena, 'Mundo') . "<br>\n";
33
34 // Funciones de arrays
35 $array = array(1, 2, 3, 4, 5);
36 $array_assoc = array("a" => 1, "b" => 2, "c" => 3);
37
38 echo "count(\$array): " . count($array) . "<br>\n";
39 sort($array);
40 echo "sort(\$array): " . implode(" ", $array) . "<br>\n";
41 rsort($array);
42 echo "rsort(\$array): " . implode(" ", $array) . "<br>\n";
43 ksort($array_assoc);
44 echo "ksort(\$array_assoc): " . implode(" ", $array_assoc) . "<br>\n";
45 krsort($array_assoc);
46 echo "krsort(\$array_assoc): " . implode(" ", $array_assoc) . "<br>\n";
47 echo "array_values(\$array_assoc): " . implode(" ", array_values($array_assoc)) . "<br>\n";
48 echo "array_keys(\$array_assoc): " . implode(" ", array_keys($array_assoc)) . "<br>\n";
49 echo "array_key_exists('a', \$array_assoc): " . (array_key_exists('a', $array_assoc) ? 'true' : 'false') . "<br>\n";
50 ?>
```

Ejecución:



```
isset($var1): true
is_null($var2): true
empty($var3): true
is_int($var3): true
is_float($var3): false
is_bool($var3): false
is_array($var4): true
intval($var1): 0
floatval($var1): 0
boolval($var1): true
strval($var3): 0
strlen($cadena): 10
strcmp($cadena, $cadena2): -1
strcasecmp($cadena, $cadena2): 0
explode($cadena, $token): Hola, Mundo
implode($token, explode($token, $cadena)): Hola Mundo
strtolower($cadena): hola mundo
strtoupper($cadena): HOLA MUNDO
strpos($cadena, 'Mundo'): 5
count($array): 5
sort($array): 1, 2, 3, 4, 5
rsort($array): 5, 4, 3, 2, 1
ksort($array_assoc): 1, 2, 3
krsort($array_assoc): 3, 2, 1
array_values($array_assoc): 3, 2, 1
array_keys($array_assoc): c, b, a
array_key_exists('a', $array_assoc): true
```