

# Laboratorio II / Laboratorio de Software II 2024

---

## Clase Teórica 03: Bucles. Arreglos. Funciones para Arreglos



**Docente: Myriam Ruiz**  
Licenciada en Informática  
Profesora en Computación y Matemática  
Programadora Universitaria



# Repeticiones

---

- Hay dos tipos de repeticiones:
  - 1) **Determinadas**: Las que **conocemos** desde el principio **cuántas veces se repetirá**
  - 2) **Indeterminadas**: Las que **no conocemos** desde el principio **cuántas veces se repetirá**



# Repeticiones – Ejemplos cotidianos

---

- En qué situación sabemos desde el principio la cantidad de repeticiones que habrá?

***No sabemos cantidad***

**Comer un Yogurt**



***Si sabemos cantidad***

**Hacer una serie de ejercicios en el Gym**



# Repeticiones: Elegir HACER o MIENTRAS?

---

*No sabemos cantidad*

**Comer un Yogurt**



**Usar MIENTRAS  
o HACER MIENTRAS**

*Si sabemos cantidad*

**Hacer una serie de ejercicios en el Gym**



**Usar HACER**

# Algoritmo HACER n VECES = for en PHP

---

```
HACER N VECES (i=1, N, paso)
    A1;
(fin_HACER)
```

```
for (inicializar; condición; paso) {
    A1;
}
```

El paso puede ser positivo o negativo, incrementarse de a 1 o en cualquier cantidad.

# Algoritmo HACER n VECES = for en PHP

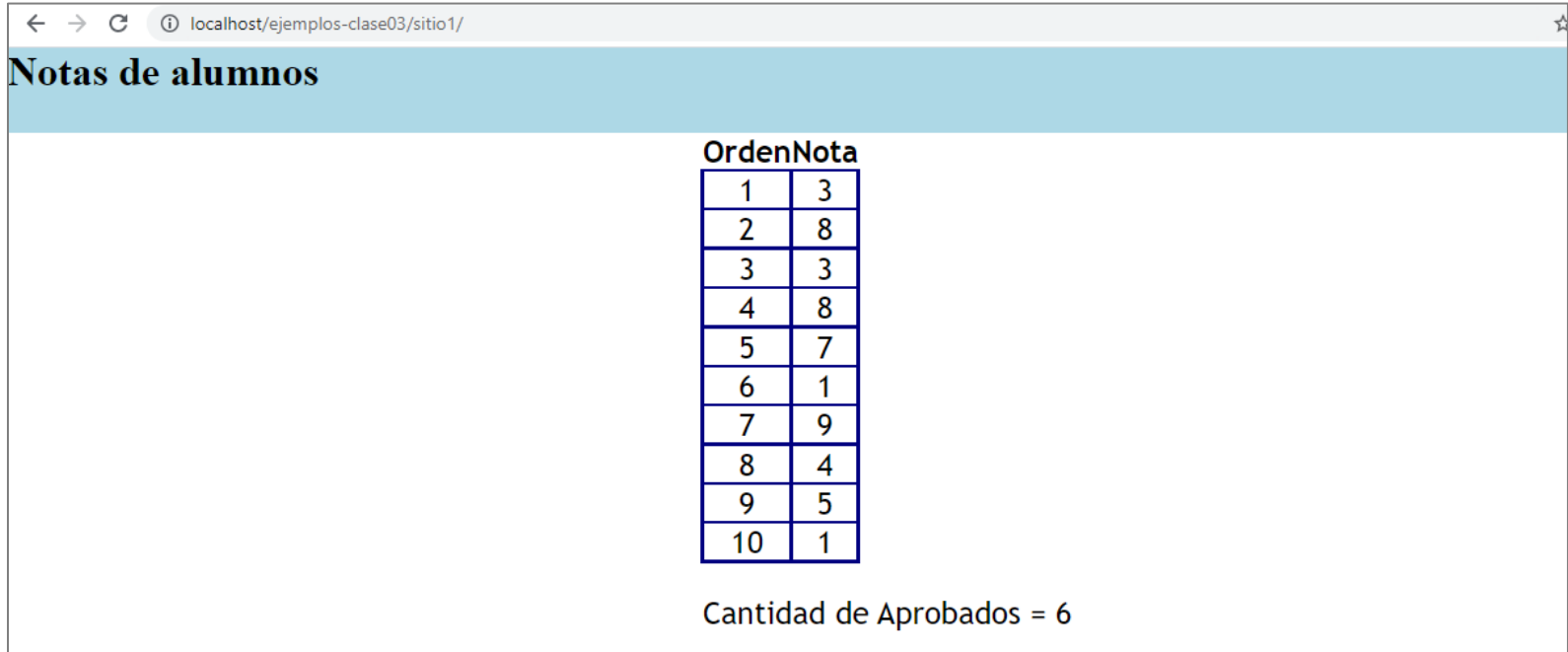
---

<pre>for (\$i = 1; \$i &lt;= \$n; \$i++) {     A1; }</pre> <p>1</p>	<pre>for (\$j = \$n; \$j &gt;= 0; \$j--) {     A1; }</pre> <p>2</p>
<pre>for (\$x = 2; \$x &lt;= \$n; \$x += 2) {     A1; }</pre> <p>3</p>	<pre>for (\$y = 81; \$y &gt;= 1; \$y -= 3) {     A1; }</pre> <p>4</p>

- 1 El paso es positivo, avanza de a uno
- 2 El paso es negativo, retrocede de a uno
- 3 El paso es positivo, avanza de a 2
- 4 El paso es negativo, retrocede de a 3

# Ejemplo: for

- Leer la nota de 10 alumnos, contar cuantos estuvieron aprobados (con nota mayor o igual a 4), mostrar en una tabla el orden y la nota. Al final mostrar un mensaje con la cantidad contada. Se verá así:



The screenshot shows a web browser window with the address bar displaying "localhost/ejemplos-clase03/sitio1/". The page has a light blue header with the text "Notas de alumnos". Below the header, there is a table with two columns: "Orden" and "Nota". The table contains 10 rows of data. Below the table, the text "Cantidad de Aprobados = 6" is displayed.

Orden	Nota
1	3
2	8
3	3
4	8
5	7
6	1
7	9
8	4
9	5
10	1

Cantidad de Aprobados = 6

# Ejemplo: for

- Código:

```
> xampp21 > htdocs > ejemplos-clase03 > sitio1 > 🐘 index.php
1  <?php
2  |      require_once('html/header.html');
3  ?>
4  |      |      |      |      <table>
5  |      |      |      |      |      <thead>
6  |      |      |      |      |      |      <tr>
7  |      |      |      |      |      |      |      <th>Orden</th>
8  |      |      |      |      |      |      |      <th>Nota</th>
9  |      |      |      |      |      |      |      </tr>
10 |      |      |      |      |      |      |      </thead>
11 |      |      |      |      |      |      |      <tbody>
12 <?php
13 |      $cont = 0;
14 |      for ($i=1; $i <= 10 ; $i++) {
```



# Ejemplo: for

- Código:

```
12  <?php
13      $cont = 0;
14      for ($i=1; $i <= 10 ; $i++) {
15          $nota = mt_rand(0, 10);
16          echo '<tr>';
17          echo '<td>' . $i.'</td><td>' . $nota . '</td>';
18          echo '</tr>';
19          if ($nota >= 4) {
20              $cont++;
21          }
22      }
23  ?>
24      </tbody>
25  </table>
```

# Ejemplo: for

---

- Código:

```
19         if ($nota >= 4) {
20             $cont++;
21         }
22     }
23     ?>
24         </tbody>
25     </table>
26     <p>Cantidad de Aprobados = <?php echo $cont; ?> </p>
27
28     <?php
29         require_once('html/footer.html');
30     ?>
```

# Algoritmo MIENTRAS = while en PHP

---

```
MIENTRAS (condición se cumpla)  
    A1;  
(fin_MIENTRAS)
```

```
while (condición se cumpla) {  
    A1;  
}
```

## Ejemplo: while

---

- Simular las compras de un super, generando y Sumando los costos de productos Mientras el dinero disponible sea mayor al costo de un nuevo producto que se quiera comprar.
  - Los costos pueden variar entre 10 y 2500 pesos.
  - El dinero puede variar entre 100 y 15000.
  - Al finalizar, mostrar el dinero con el que se contaba, la cantidad de productos comprada y el monto total de la compra.

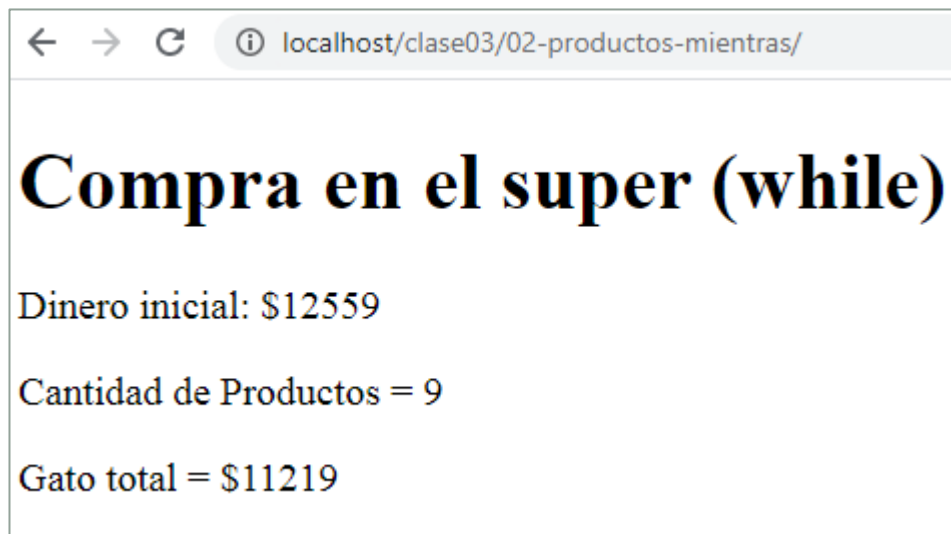
# Ejemplo: while

---

```
<article>
  <?php
    $total = 0;
    $cantidad = 0;
    $dinero = mt_rand(100, 15000);
    $dineroInicial = $dinero;
    $costo = mt_rand(10, 2500);
    while ($dinero >= $costo) {
      $dinero -= $costo; // $dinero = $dinero - $costo;
      $total += $costo; // $total = $total + $costo;
      $cantidad++;
      $costo = mt_rand(10, 2500);
    }
    echo '<p>Dinero inicial: $' . $dineroInicial . '</p>';
    echo '<p>Cantidad de Productos = ' . $cantidad . '</p>';
    echo '<p>Gato total = $' . $total . '</p>';
  ?>
</article>
```

# Ejemplo: while

---



## Ejemplo: Separar los dígitos de un número entero

- Una forma de poder separar los dígitos de un número, es utilizar operaciones de división entera y cociente, a fin de poder ver de derecha a izquierda cada dígito.
- Analicemos la siguiente división entera en 10.

$$\begin{array}{r} 123 \overline{) 10} \\ - 10 \phantom{00} \\ \hline 23 \phantom{00} \\ - 20 \phantom{00} \\ \hline 3 \end{array}$$

← Cociente entero entre Num y 10

← Resto de dividir Num en 10

# Ejemplo: Separar los dígitos de un número entero

$$\begin{array}{r} 123 \overline{) 10} \\ - 10 \\ \hline 23 \\ - 20 \\ \hline 3 \end{array}$$

intdiv(Num, 10) o (int) (num/10)

Num % 10

- Observe que si nos interesa saber con qué dígito termina un número, en este caso 3, usamos el resto (%) en 10
- Si quisiéramos volver a ver el último dígito, pero ahora que sea el 2, podemos quitar el 3 y volver a aplicar %. Para quitar un dígito desde la derecha, podemos hacer una división entera en 10



# Ejemplo: Separar los dígitos de un número entero

---

- Un algoritmo (existen otros, que pueden consultar en práctica o crear uno, es sólo una sugerencia) para separar los dígitos puede ser el siguiente:

P1. LEER(num)

P2.  $\text{aux} \leftarrow \text{num}$

P3. MIENTRAS ( $\text{aux} > 0$ ) HACER // <- por qué compara con 0?

$\text{ultimodig} \leftarrow \text{Resto}(\text{aux}, 10)$

    ESCRIBIR ultimoDig // se puede hacer cualquier operación aquí

$\text{aux} \leftarrow \text{DivisionEntera}(\text{aux}, 10)$  // pierde último dígito

Fin-Mientras

P4. FIN.

# Ejemplo: Separar los dígitos de un número entero

- Implementación del código. Generar números entre 0 y 99999, y mostrar sus dígitos por separado.

```
<article>
  <?php
    $num = mt_rand(0, 99999);
    $aux = $num;
    echo '<p>Número: ' . $num '</p>';
    while ($aux > 0) {
      $ultimoDig = $aux % 10;
      echo '<p>Díg: ' . $ultimoDig . '</p>';
      $aux = (int)($aux/10);
    }
  ?>
</article>
```

num	aux	ultiDig
41506	41506	6
	4150	0
	415	5
	41	1
	4	4
	0	

*Seguimiento con  
Prueba de Escritorio*

Observe que el último cociente en 10 dará siempre cero

# Ejemplo: Separar los dígitos de un número entero

- Implementación del código. Generar números entre 0 y 99999, y mostrar sus dígitos por separado.

← → ↻ ⓘ localhost/clase03/04-digitos-while/

## Separar dígitos para analizar (while)

Número: 41506

Díg: 6

Díg: 0

Díg: 5

Díg: 1

Díg: 4

num	aux	ultiDig
41506	41506	6
	4150	0
	415	5
	41	1
	4	4
	0	

*Seguimiento con  
Prueba de Escritorio*

Observe que el último cociente en 10 dará siempre cero

# Algoritmo HACER MIENTRAS = do while en PHP

---

<pre>HACER   A1; MIENTRAS(condición se cumpla)</pre>	<pre>do {   A1; } while(condición se cumpla);</pre>
--	---

# Ejemplo: do while

---

- Programar un juego que cumpla estas condiciones:
  - Debe tirar 2 dados mientras su suma sea distinta de 7.
  - Ir contando las veces que se tiró.
  - Al final mostrar el total de veces tiradas y cuáles fueron los valores de cada dado que dio suma 7.

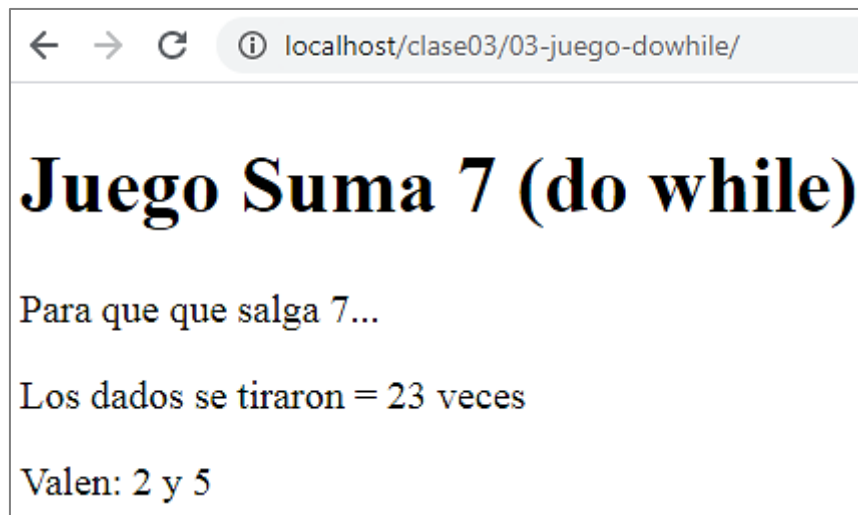
## Ejemplo: do while

---

```
<article>
  <?php
    $cant = 0; // inicializar variable
    do {
      $dado1 = mt_rand(1, 6); // el dado tiene 6 caras
      $dado2 = mt_rand(1, 6);
      $suma = $dado1 + $dado2;
      $cant++; // cuenta las veces que va tirando
    } while($suma != 7);
    echo '<p>Para que que salga 7...</p>';
    echo '<p>Los dados se tiraron = ' . $cant . ' veces</p>';
    echo '<p>Valen: ' . $dado1 . ' y ' . $dado2 . '</p>';
  ?>
</article>
```

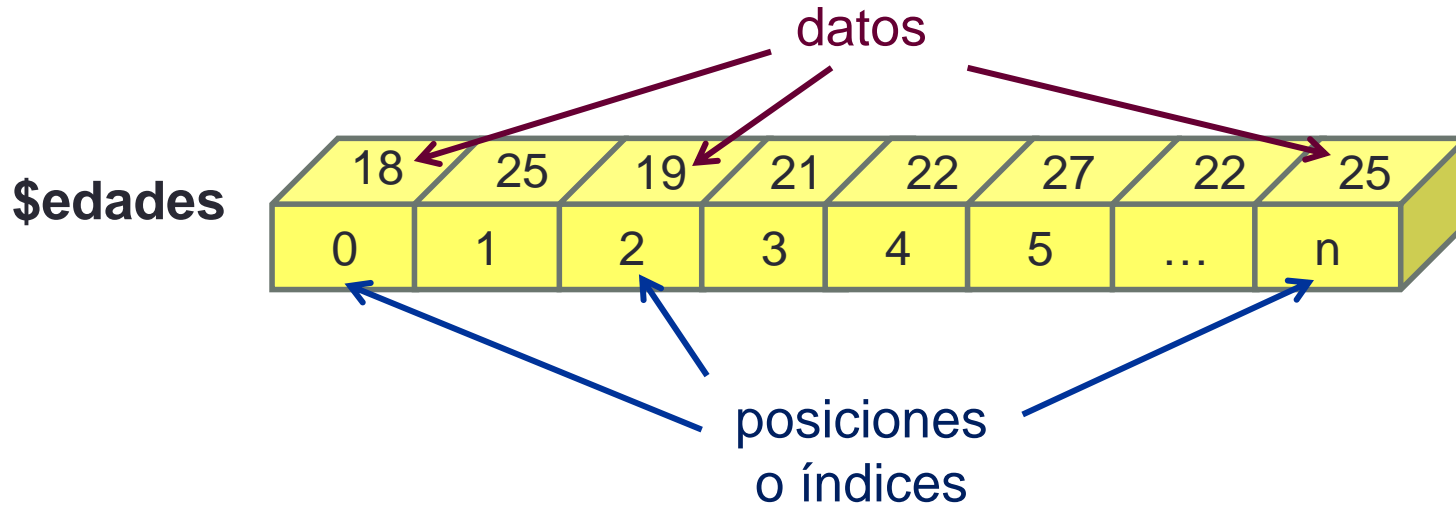
# Ejemplo: do while

---



# Arreglos (array)

- Un array es una colección de datos almacenados bajo el mismo nombre de variable, identificables mediante un índice o posición. Ejemplo: array \$edades





# Arreglos (array)

---

- Un array se pueden ver como una Matriz.
- En el caso anterior es una matriz fila, es decir de orden  $1 \times n$ , 1 fila y n columnas, conocida en programación como **vector**.
- En notación de Matrices sería:

$$(a_1 \ a_2 \ \dots \ a_n) \in \mathbb{R}^{1 \times n}$$

# Arreglos (array)

---

- También veremos arreglos de tipo Matriz rectangular, o sea de orden  $m \times n$ , que son, por ejemplo, la forma en que un **formulario** almacena los datos.
- En notación de Matrices sería:

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1j} & \dots & a_{1n} \\ \vdots & \vdots & & \vdots & & \vdots \\ a_{i1} & a_{i2} & \dots & a_{ij} & \dots & a_{in} \\ \vdots & \vdots & & \vdots & & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mj} & \dots & a_{mn} \end{pmatrix}$$

*fila i*

*columna j*

# Tipos de Arreglos

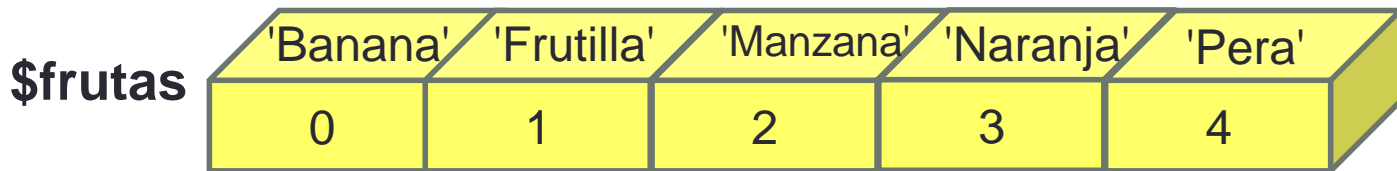
---

- Hay dos tipos de arreglos:
  - **Array Indexado**: se accede por índice **numérico**
  - **Array Asociativo**: se accede mediante **claves alfanuméricas**

# Array Indexado

---

- Los array pueden contener números (como se vio), letras o palabras



- Para **crear el array**

```
$frutas = array('Banana', 'Frutilla', 'Manzana', 'Naranja', 'Pera');
```

// desde PHP 5.4

```
$frutas = ['Banana', 'Frutilla', 'Manzana', 'Naranja', 'Pera'];
```

# Array Indexado

---

- **Mostrar el contenido de a una componente** de un array **matriz fila** usando corchete `[]` e índice:

echo \$frutas[0]; // mostrará Banana

echo \$frutas[3]; // mostrará Naranja

\$frutas	'Banana'	'Frutilla'	'Manzana'	'Naranja'	'Pera'
	0	1	2	3	4

# Array Indexado

---

- **Modificar el contenido o cargarle datos** a un array:

```
$frutas[0] = 'Mandarina';
```

```
$frutas[1] = 'Uva';
```

```
print_r($frutas)
```

\$frutas	'Mandarina'	'Uva'	'Manzana'	'Naranja'	'Pera'
	0	1	2	3	4

```
Array ( [0] => Mandarina [1] => Uva [2] => Manzana [3] => Naranja [4] => Pera )
```

# Array Indexado

---

- **Mostrar todas las componentes de un array:**

```
print_r($arreglo);
```

```
print_r($frutas);
```

Array ( [0] => Banana [1] => Frutilla [2] => Manzana [3] => Naranja [4] => Pera )



Los índices se crearon de manera automática

# Array Indexado

---

- **Mostrar todas las componentes de un array y el tipo de dato:**

```
var_dump($arreglo);
```

```
var_dump($frutas);
```

```
array(5) { [0]=> string(9) "Mandarina" [1]=> string(3) "Uva" [2]=> string(7) "Manzana" [3]=> string(7) "Naranja" [4]=> string(4) "Pera" }
```



Diagram illustrating the components of the array dump output:

- índice**: Points to the index `[0]`.
- Tipo de dato**: Points to the data type `string(9)`.
- dato almacenado**: Points to the stored value `"Mandarina"`.



# Array Indexado – Acceso a componente que no existe

---

- **Error cuando accedemos a una componente que no existe**

echo \$frutas[5]; // el arreglo sólo llega hasta 4

**Notice:** Undefined offset: 5 in C:\xampp19\htdocs\ejemplosClase02\08-arregloFrutas\index.php on line 14

\$frutas	'Banana'	'Frutilla'	'Manzana'	'Naranja'	'Pera'
	0	1	2	3	4

# Array Indexado – Se puede agregar componentes

---

- **Si asignamos nuevos valores sin indicar una posición, estos se agregan al arreglo**

`$frutas[] = 'Arándano';`

\$frutas	'Banana'	'Frutilla'	'Manzana'	'Naranja'	'Pera'	'Arándano'
	0	1	2	3	4	5

# Array Indexado – Se puede agregar componentes

---

- **Si asignamos nuevos valores indicando posiciones no contiguas**

Array ( [0] => Banana [1] => Frutilla [2] => Manzana [3] => Naranja [4] => Pera [5] => Uva )

\$frutas[20] = 'Ananá';

Array ( [0] => Banana [1] => Frutilla [2] => Manzana [3] => Naranja [4] => Pera [5] => Uva [20] => Ananá )

# Array Indexado – Se puede agregar componentes

---

- **Si asignamos nuevos valores indicando posiciones no contiguas**

Array ( [0] => Banana [1] => Frutilla [2] => Manzana [3] => Naranja [4] => Pera [5] => Uva [20] => Ananá )

\$frutas[10] = 'Tomate';

Array ( [0] => Banana [1] => Frutilla [2] => Manzana [3] => Naranja [4] => Pera [5] => Uva [20] => Ananá [10] => Tomate )

\* Observe que el índice 10 no se ubica antes del 20

# Array Indexado – Se puede agregar componentes

---

- **Si asignamos nuevos valores indicando posiciones no contiguas**

Array ( [0] => Banana [1] => Frutilla [2] => Manzana [3] => Naranja [4] => Pera [5] => Uva [20] => Ananá [10] => Tomate )

```
$frutas[] = 'Kiwi';
```

Array ( [0] => Banana [1] => Frutilla [2] => Manzana [3] => Naranja [4] => Pera [5] => Uva [20] => Ananá [10] => Tomate [21] => Kiwi )

\* Observe que el nuevo valor se agrega automáticamente en un índice que le sigue al más alto que había

# Una cadena puede ser tratada como un array

La palabra Muercielago tiene 6 vocales

C: > xampp20 > htdocs > clase03 > 06-cadena-array > index.php

```
14      <article>
15      <?php
16          $palabra = 'Muercielago';
17          $longitud = strlen($palabra); // cantidad de letras
18          $cantidad = 0;
19          for($i = 0; $i < $longitud; $i++){
20              if($palabra[$i] == 'a' || $palabra[$i] == 'e'
21              || $palabra[$i] == 'i' || $palabra[$i] == 'o'
22              || $palabra[$i] == 'u'){
23                  $cantidad++;
24              }
25          }
26          echo '<p>La palabra ' . $palabra . ' tiene ' . $cantidad . ' vocales</p>';
27      ?>
28  </article>
29  </section>
```

# Array Asociativo

---

- **Creación:**

`$variable = array(clave1 => dato1, clave2 => dato2, ...);`

`$variable = [clave1 => dato1, clave2 => dato2, ...];`

**Ejemplo:**

**\$materias**

'Laboratorio I'	'Laboratorio II'	'Elementos'	'Programación'
P03	P07	PU1	P06

`$materias = array('P03' => 'Laboratorio I', 'P07' => 'Laboratorio II',  
'PU1' => 'Elementos', 'P06' => 'Programación');`

# Array Asociativo

---

- **Modificación o cargado de datos:**

```
$arreglo['clave'] = dato;
```

Ejemplo:

```
print_r($materias);
```

```
$materias['PU1'] = 'Elementos de Comp. y Lógica';
```

```
echo '<hr>';
```





```
print_r($materias);
```



# Array Asociativo

---

- **Modificación o cargado de datos:**

    localhost/clase03/07-arreglo-asociativo/

## Ejemplo modificación de array

Array ( [P03] => Laboratorio I [P07] => Laboratorio II [PU1] => Elementos [P06] => Programación )

---

Array ( [P03] => Laboratorio I [P07] => Laboratorio II [PU1] => Elementos de Comp. y Lógica [P06] => Programación )

## Agregar en un array asociativo, un elemento sin poner un índice

- Al agregar un nuevo elemento sin indicar un índice, lo agregará igual que en los array indexados:

```
<?php
    $materias = [
        'P03' => 'Laboratorio I',
        'P07' => 'Laboratorio II',
        'PU1' => 'Elementos de Comp. y Lógica',
        'P06' => 'Programación'
    ];

    print_r($materias);
    echo '<hr>';
    $materias[] = 'Álgebra y Geom. Analítica';
    print_r($materias);
?>
```

## Agregar en un array asociativo, un elemento sin poner un índice

---

- Observe que le coloca un índice, empezando por el valor numérico más bajo que corresponda:

```
Array ( [P03] => Laboratorio I [P07] => Laboratorio II [PU1] => Elementos de Comp. y Lógica  
[P06] => Programación )
```

```
Array ( [P03] => Laboratorio I [P07] => Laboratorio II [PU1] => Elementos de Comp. y Lógica  
[P06] => Programación [0] => Álgebra y Geom. Analítica )
```



# Se pueden guardar distintos tipos de datos

---

```
$alumno = array('edad' => 19, 'Carrera' => 'Lic. en  
Informática', 'Sueldo' => 15490.45);  
var_dump($alumno);
```

← → ↻ ⓘ localhost/clase03/08-arregloVarDump/

```
array(3) { ["edad"]=> int(19) ["Carrera"]=> string(20) "Lic. en Informática" ["Sueldo"]=> float(15490.45) }
```

# Se podrán duplicar los índices?

- Intentaremos usar más de una vez el mismo índice en los arreglos indexados y asociativos

```
<article>
  <?php
    // arreglo indexado, índice automático
    $datos0 = array('Banana', 'Manzana', 'Mango');
    print_r($datos0);
    echo '<p>Tiene ' . count($datos0) . ' componentes</p>';
    echo '<hr>';
    // arreglo indexado, índice hecho a mano
    // se intenta duplicar índice 1
    $datos1 = array(0=>'Banana', 1=>'Manzana', 1=>'Mango');
    print_r($datos1);
    echo '<p>Tiene ' . count($datos1) . ' componentes</p>';
    echo '<hr>';
    //arreglo asociativo, se intenta duplicar índice 'nombre'
    $datos2 = array('nombre' => 'Myriam', 'nombre' => 'Juan');
    print_r($datos2);
    echo '<p>Tiene ' . count($datos2) . ' componentes</p>';
    echo '<hr>';
  ?>
```

# Se podrán duplicar sin querer los índices?

- Se pudo usar el mismo índice más de una vez? Veamos:

<p>← → ↻ ⓘ localhost/clase03/09-indice-array/</p>
<p><b>Ejemplo: intento de duplicar índice en array</b></p>
<p>Array ( [0] =&gt; Banana [1] =&gt; Manzana [2] =&gt; Mango )</p> <p>Tiene 3 componentes</p>
<p>Array ( [0] =&gt; Banana [1] =&gt; Mango )</p> <p>Tiene 2 componentes</p>
<p>Array ( [nombre] =&gt; Juan )</p> <p>Tiene 1 componentes</p>

Observe que en el 2do y 3er ejemplo, al duplicar el índice, sólo prevalece el último uso y se pierde el dato anterior que tenía el mismo índice

# Array Matriz Rectangular

---

- Cuando tenemos un arreglo en el que cada componente es un arreglo, se puede pensar como una matriz de dimensión  $m \times n$ .
- De la misma manera que nos referimos en matrices a la componente con 2 subíndices, por ejemplo  $a_{12}$ , tendremos que usar dos pares de corchetes `[][]` para indicar qué componente queremos seleccionar

# Array Matriz Rectangular

---

- Si la matriz se llama A, la componente de la fila 1, columna 2 se escribe:
  - Notación de Álgebra:  $a_{12}$
  - Notación de php: `A[1][2]`

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}$$



# Array Matriz Rectangular

s > clase03 > 10-arreglo-de-arreglos > index.php

```
<article>
```

```
<h2>Cada clave (país) tiene asociado un array</h2>
```

```
<?php
```

```
    $pais = array (
```

```
        'Argentina' => array (
```

```
            'Idioma' => 'Español',
```

```
            'Moneda' => 'Peso',
```

```
            'Huso Horario' => 'UTC-3'
```

```
        ),
```

```
        'Brasil' => array (
```

```
            'Idioma' => 'Portugues',
```

```
            'Moneda' => 'Real',
```

```
            'Huso Horario' => 'UTC-3'
```

```
        )
```

```
    );
```

```
    print_r($pais);
```

```
    echo '<hr>';
```

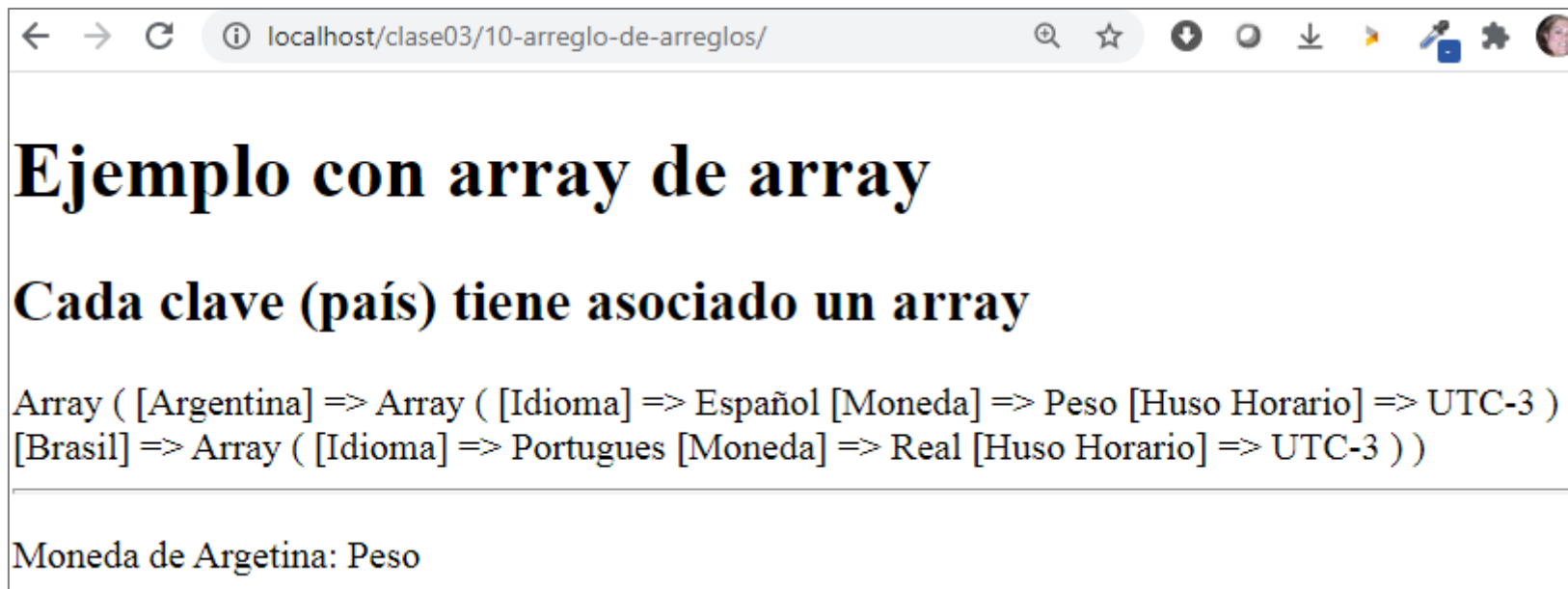
```
    echo '<p>Moneda de Argetina: ' . $pais['Argentina']['Moneda'] . '</p>';
```

```
?>
```

fila Argentina,  
columna Moneda



# Array Matriz Rectangular



The screenshot shows a web browser window with the address bar displaying 'localhost/clase03/10-arreglo-de-arreglos/'. The page content includes a main heading, a sub-heading, and a code example for a 2D array representing country data. The code uses nested arrays to map country names to their respective language, currency, and time zone. Below the code, there is a text label for the currency of Argentina.

← → ↻ ⓘ localhost/clase03/10-arreglo-de-arreglos/ 🔍 ☆ ⬇️ ⬅️ ⬇️ ⬆️ ⚙️ 👤

## Ejemplo con array de array

### Cada clave (país) tiene asociado un array

```
Array ( [Argentina] => Array ( [Idioma] => Español [Moneda] => Peso [Huso Horario] => UTC-3 )  
[Brasil] => Array ( [Idioma] => Portugues [Moneda] => Real [Huso Horario] => UTC-3 ) )
```

Moneda de Argetina: Peso

# Array Constante

---

- **Se pueden crear arreglos Constantes**

// Válido desde PHP 5.6.0

```
const DOCENTES = array('Flores', 'Pepe', 'Ruiz', 'Singh');
```

// Válido desde PHP 7

```
define('DOCENTES2', array('Flores', 'Pepe', 'Ruiz', 'Singh'));
```

```
echo DOCENTES2[1]; // Mostrará Pepe
```

# foreach

---

- Con la finalidad de recorrer todas las componentes de un array se puede utilizar el bucle foreach. Nos ahorramos tener que definir un índice.

- **Para array indexado**

```
foreach ($arreglo as $valor) {  
}
```

- **Para array asociativo**

```
foreach ($arreglo as $clave => $valor) {  
}
```

# foreach para array indexado

---

- Recorrer un arreglo con edades de alumnos y encontrar la edad más alta. Mostrar al final dicha edad.
- Se lee para cada \$edad que se encuentra en \$edadesAlumnos

```
<section>
  <article>
    <?php
      $edadesAlumnos = array(25,33, 18, 41, 20, 31);
      $mayor = 0; // inicializamos en cero
      foreach ($edadesAlumnos as $edad) {
        if($edad > $mayor) {
          $mayor = $edad;
        }
      }
      echo '<p>Edad más alta: ' . $mayor . '</p>';
    ?>
  </article>
```

# foreach para array indexado

---

- Recorrer un arreglo con edades de alumnos y encontrar la edad más alta. Mostrar al final dicha edad.



# foreach para array asociativo

---

- Recorrer un arreglo asociativo de películas, que contiene título y actor principal. Mostrar el arreglo en una tabla.
- Debe lucir así:



A screenshot of a web browser window. The address bar shows 'localhost/clase03/13-foreach-asociativo/'. The page content features a large heading 'Ejemplo foreach con arreglo asociativo' followed by a table with two columns: 'Título' and 'Protagonista'. The table lists four movies: 'Bad Boys Life' with 'Will Smith', 'Aves de Presa' with 'Margot Robbie', 'Viuda Negra' with 'Scarlett Johansson', and 'Rapido y Furioso 9' with 'Vin Diesel'.

Título	Protagonista
Bad Boys Life	Will Smith
Aves de Presa	Margot Robbie
Viuda Negra	Scarlett Johansson
Rapido y Furioso 9	Vin Diesel

# foreach para array asociativo

```
<article>
  <?php
    $peliculas = array(
      'Bad Boys Life' => 'Will Smith',
      'Aves de Presa' => 'Margot Robbie',
      'Viuda Negra' => 'Scarlett Johansson',
      'Rapido y Furioso 9' => 'Vin Diesel'
    );
    echo '<table><thead><tr><th>Título</th><th>Protagonista</th></tr></thead>';
    echo '<tbody>';
    foreach($peliculas as $titulo => $protag){
      echo '<tr><td>' . $titulo . '</td>'
        . '<td>' . $protag . '</td></tr>';
    }
    echo '</tbody></table>';
  ?>
</article>
```

Para cada elemento del arreglo, tomar el índice como \$titulo y el dato como \$protag



# Funciones Predefinidas para Arrays

---

- **Ordenar por los datos**

- `sort($arreglo)` // Ordena de manera ascendente
- `rsort($arreglo)` // Ordena de manera descendente

- **Ordenar por claves**

- `ksort($arreglo)` // Ordena de manera ascendente
- `krsort($arreglo)` // Ordena de manera descendente

- **Desordenar los datos**

- `shuffle($arreglo)` // Desordena las componentes

# Funciones para Arrays

---

- **Verificar si es un arreglo**
  - `is_array($arreglo)` // devuelve true o false
- **Verificar si clave existe en el arreglo**
  - `array_key_exist($clave, $arreglo)` // existencia de clave, devuelve true o false
  - `in_array($valor, $arreglo)` // existencia de valor, devuelve true o false
- **Buscar un dato en un arreglo (retorna clave o falso)**
  - `array_search($valor, $arreglo)` // \$valor es lo que se busca. Devuelve la clave del valor encontrado o false

# Funciones para Arrays

---

- **Obtener al azar la clave de una componente**  
`array_rand ( array $array [, int $num = 1 ] )`
- **Sumar todas las componentes de un array**  
`array_sum ( array $array ) : number`
- **Eliminar componentes de un array**  
`unset($arreglo[indice]) : void`

Hay más funciones para arreglos:

<https://www.php.net/manual/es/ref.array.php>

# Funciones para Arrays – Ejemplo array\_search

---

Dado un arreglo con nombres de profe, encontrar la posición del profe Daniel.

```
<article>
  <?php
    $profes = array('Daniel', 'Emmanuel', 'Juan Pablo', 'Myriam');
    print_r($profes);
    $buscado = 'Daniel';
    $resultado = array_search($buscado, $profes);
    echo '<p>Daniel está en la posición: ' . $resultado . '</p>';
  ?>
</article>
```

# Funciones para Arrays – Ejemplo array\_search

---

Dado un arreglo con nombres de profe, encontrar la posición del profe Daniel.

A screenshot of a web browser window. The address bar shows the URL 'localhost/clase03/14-search-indexado/'. The main content area has a large heading 'Ejemplo foreach con arreglo indexado' in bold black font. Below the heading, there is a line of text: 'Array ( [0] => Daniel [1] => Emmanuel [2] => Juan Pablo [3] => Myriam )'. At the bottom, it says 'Daniel está en la posición: 0'.

# Funciones para Arrays – Ejemplo array\_search

---

Dado el nombre de un producto almacenado en un array asociativo encontrar su código (que es una clave)

```
<article>
  <?php
    $productos = array(
      'A001' => 'Azucar',
      'A002' => 'Yerba',
      'A003' => 'Fernet'
    );
    $nombre = 'Fernet';
    $codigo = array_search($nombre, $productos);
    echo '<p>Código: ' . $codigo . '</p>';
  ?>
```

# Funciones para Arrays – Ejemplo array\_search

---

Dado el nombre de un producto almacenado en un array asociativo encontrar su código (que es una clave)



# Funciones para Arrays – Ejemplo in\_array y sort

---

- Llenar un arreglo de 5 componentes con números aleatorios entre 0 y 36. Mostrar el arreglo original, ordenarlo de manera ascendente y mostrarlo de nuevo. Nota: Un valor no podrá guardarse más de una vez. Es decir, es sin repetición




# Funciones para Arrays – Ejemplo in\_array y sort

---

```
<article>
  <?php
    $sorteo = array(); // arreglo vacío
    for($i= 0; $i <= 4; $i++){
      do {
        $num = mt_rand(0, 36);
      } while (in_array($num, $sorteo)); // impide duplicados
      $sorteo[$i] = $num; // almacena un valor en la posición $i
    }
    print_r ($sorteo);
    sort($sorteo); // ordena el arreglo
    echo '<hr>';
    print_r ($sorteo);
  ?>
```

# Funciones para Arrays – Ejemplo in\_array y sort

---

 localhost/clase03/16-odernarArregloIndexado/

## Uso de in\_array y sort

Array ( [0] => 7 [1] => 4 [2] => 36 [3] => 8 [4] => 24 )

---

Array ( [0] => 4 [1] => 7 [2] => 8 [3] => 24 [4] => 36 )

# Funciones para Arrays – Ejemplo array\_rand

---

- `array_rand ( array $array [, int $num = 1 ] )`
  - Selecciona uno o más valores aleatorios de un array y devuelve la clave (o claves) de dichos valores aleatorios
  - \$num representa la cantidad de claves que queremos que nos devuelva. Si no ponemos un número, toma por defecto 1
- Ejemplo: Almacenando las Marcas de los Celulares más vendidos en Argentina, simule la compra de 1000 equipos , seleccionando al azar una componente de dicho arreglo. Contar cuantos equipos se compraron de cada marca, contándola según que clave tiene.

# Funciones para Arrays – Ejemplo array\_rand

---

```
<article>
  <?php
    $marcas = array('Apple', 'Huawei', 'LG','Motorola', 'Samsung', 'Xiaomi');
    $cantApple = 0; // inicializamos todos los contadores
    $cantHuawei = 0;
    $cantLG = 0;
    $cantMotorola = 0;
    $cantSamsung = 0;
    $cantXiaomi = 0;
    for ($i=0; $i < 1000 ; $i++) { // simulamos la compra de 1000 equipos
      $clave = array_rand($marcas); // selecciona una componente
      switch ($clave) {
```

# Funciones para Arrays – Ejemplo array\_rand

---

```
$cantXiaomi = 0;  
for ($i=0; $i < 1000 ; $i++) { // simulamos la compra de 1000 equipos  
    $clave = array_rand($marcas); // selecciona una componente  
    switch ($clave) {  
        case 0:  
            $cantApple++;  
            break;  
        case 1:  
            $cantHuawei++;  
            break;  
        case 2:  
            $cantLG++;  
            break;  
        case 3:  
            $cantMotorola++;  
            break;  
        case 4:  
            $cantSamsung++;  
            break;  
    }  
}
```

# Funciones para Arrays – Ejemplo array\_rand

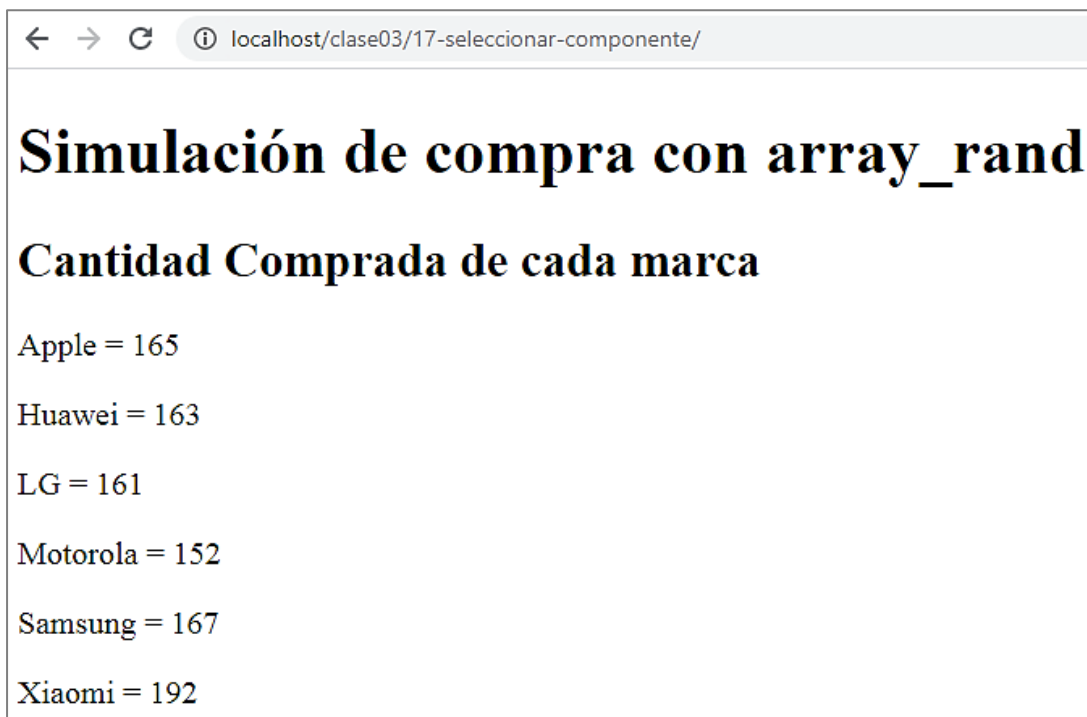
---

```
        case 4:
            $cantSamsung++;
            break;
        case 5:
            $cantXiaomi++;
    }
}

echo '<h2>Cantidad Comprada de cada marca</h2>';
echo '<p>Apple = ' . $cantApple . '</p>';
echo '<p>Huawei = ' . $cantHuawei . '</p>';
echo '<p>LG = ' . $cantLG . '</p>';
echo '<p>Motorola = ' . $cantMotorola . '</p>';
echo '<p>Samsung = ' . $cantSamsung . '</p>';
echo '<p>Xiaomi = ' . $cantXiaomi . '</p>';
```

# Funciones para Arrays – Ejemplo array\_rand

---



A screenshot of a web browser window. The address bar shows the URL `localhost/clase03/17-seleccionar-componente/`. The page content includes a main title, a subtitle, and a list of brand names with their corresponding purchase counts.

**Simulación de compra con array\_rand**

**Cantidad Comprada de cada marca**

Apple = 165

Huawei = 163

LG = 161

Motorola = 152

Samsung = 167

Xiaomi = 192

# Funciones para Arrays – Ejemplo unset

```
<?php
    require_once('html/header.html');
    $frutas = ['Banana', 'Frutilla', 'Manzana', 'Naranja', 'Pera'];
    print_r($frutas);
    echo '<hr>';
    unset($frutas[1]);
    print_r($frutas);
    require_once('html/footer.html');
?>
```

```
Array ( [0] => Banana [1] => Frutilla [2] => Manzana [3] => Naranja [4] => Pera )
Array ( [0] => Banana [2] => Manzana [3] => Naranja [4] => Pera )
```



# Declaración de arreglos en archivo separado

- Ejemplo: en el archivo datos.php, crear dos arreglos que tengan:

```
xampp21 > htdocs > ejemplos-clase03 > sitio > php > 🐘 datos.php
<?php
    $fotosAfiliados =
        [ '11111/0' => 'ruizm.jpg',
          '11112/1' => 'pepejp.jpg',
          '11113/0' => 'corralesm.jpg',
          '11114/0' => 'singhd.jpg'
        ];

    $nombresAfiliados =
        [ '11111/0' => 'Ruiz Myriam',
          '11112/1' => 'Pepe Juan Pablo',
          '11113/0' => 'Corrales Malena',
          '11114/0' => 'Singh Daniel'
        ];
?>
```

# Declaración de arreglos en archivo separado

- Ejemplo: archivo index.php:

```
mpp21 > htdocs > ejemplos-clase03 > sitio > 🐘 index.php
```

```
<?php
    require_once 'html/header.html';
    require_once 'php/datos.php'; /* archivo con arrays */

    /* seleccionar un afiliado al azar */
    $clave = array_rand($nombresAfiliados);
?>

<article>
    <figure>
        
        <figcaption><?php echo $nombresAfiliados[$clave]; ?></figcaption>
    </figure>
</article>
```

# Declaración de arreglos en archivo separado

---

- Ejemplo: archivo index.php:

## Afiliados Acción Social



Pepe Juan Pablo

# Declaración de arreglos en archivo separado

---

- Ejemplo: Podríamos mostrar ambos arreglos en una tabla:

```
<table>
  <thead>
    <tr><td>Código</td><td>Nombre</td><td>foto</td></tr>
  </thead>
  <tbody>
    <?php
      foreach($fotosAfiliados as $codigo => $foto) {
        echo '<tr><td>' . $codigo . '</td>';
        echo '<td>' . $nombresAfiliados[$codigo] . '</td>';
        echo '<td></tr>';
      }
    ?>
  </tbody>
</table>
```

# Declaración de arreglos en archivo separado

- Ejemplo: Podríamos mostrar ambos arreglos en una tabla:

## Afiliados Acción Social

11112/1	Pepe Juan Pablo	
11113/0	Corrales Malena	
11114/0	Singh Daniel	