

CLASES DE ARREGLOS EN PHP

Los arreglos se pueden definir como colecciones de datos que presentan características similares entre ellos.

Los arreglos pueden ser de tipo vectorial, matriciales o asociativos.

Los arreglos se crean por medio de la palabra reservada `array()`, dentro de los paréntesis se ubica la colección de datos que conformaran el arreglo.

En el siguiente ejemplo podemos observar la creación de un arreglo.

```
<?php
```

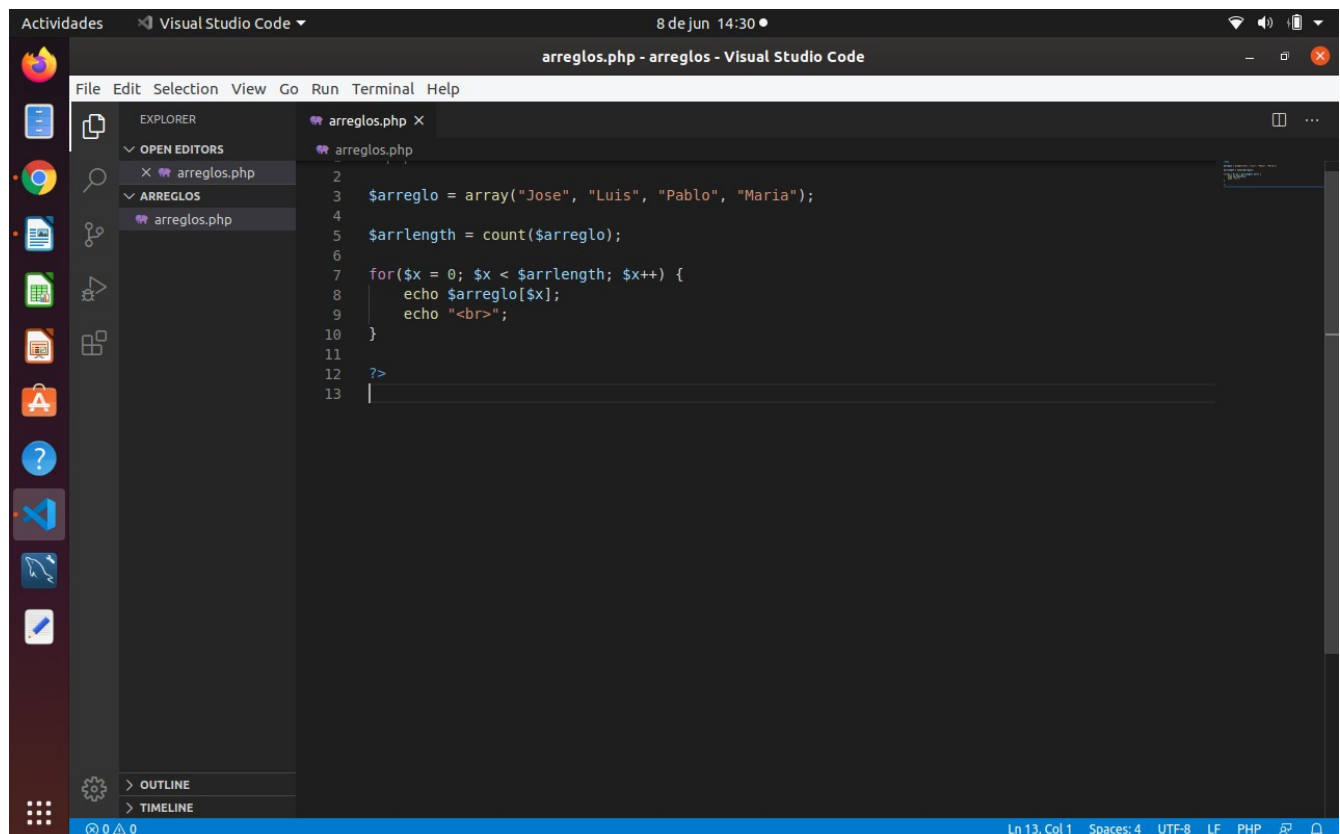
```
$arreglo = array("Jose", "Luis", "Pablo", "Maria");
```

```
$arlength = count($arreglo);
```

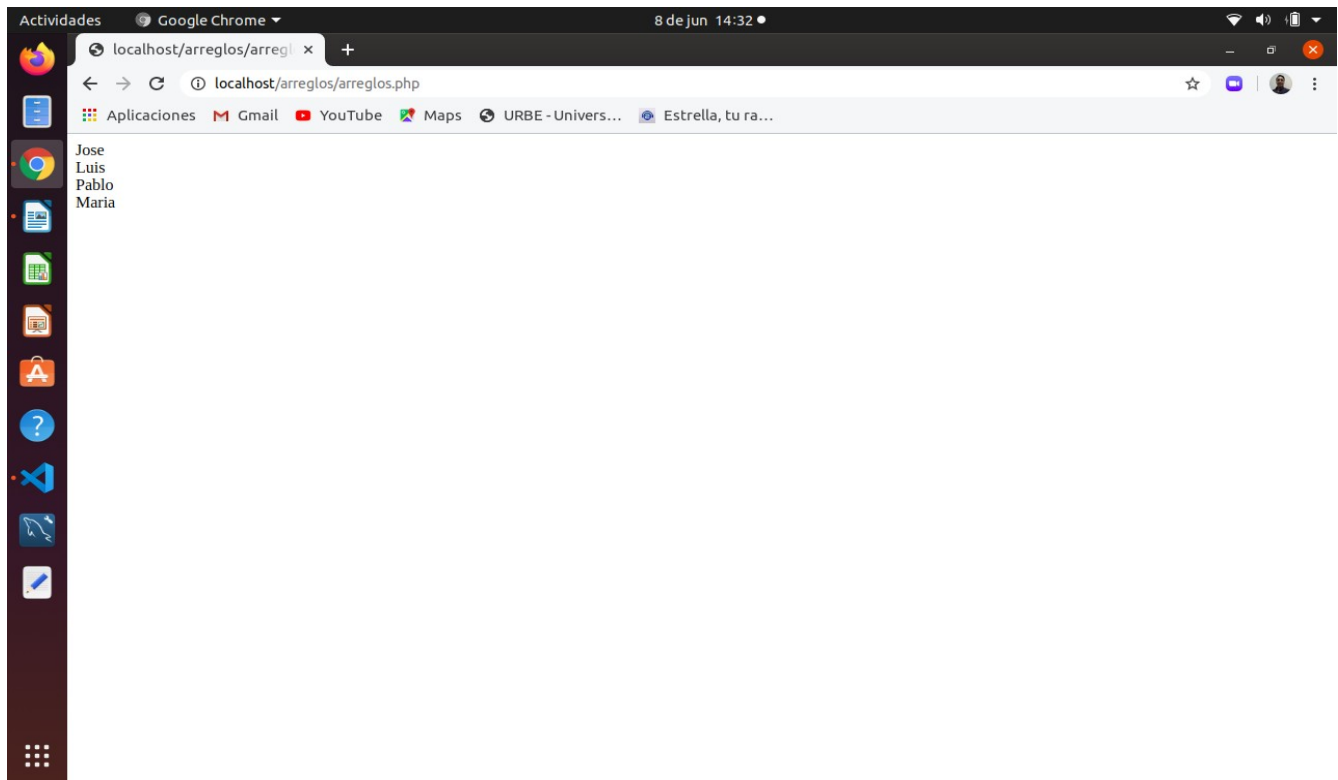
```
for($x = 0; $x < $arlength; $x++) {  
    echo $arreglo[$x];  
    echo "<br>";  
}
```

```
?>
```

Vista del código en el Visual Studio Code



Vista del resultado en el navegador



Impresión de la información del arreglo por medio de `var_dump()` y `print_r()`

Por medio de estas 2 palabras reservadas por php se puede imprimir en pantalla el contenido del arreglo mostrando sus respectivos atributos de diferente forma.

`var_dump()`

Con esta palabra reservada no muestra las características del contenido del arreglo como lo son el índice del dato, así como también el tipo de dato y su valor.

Código

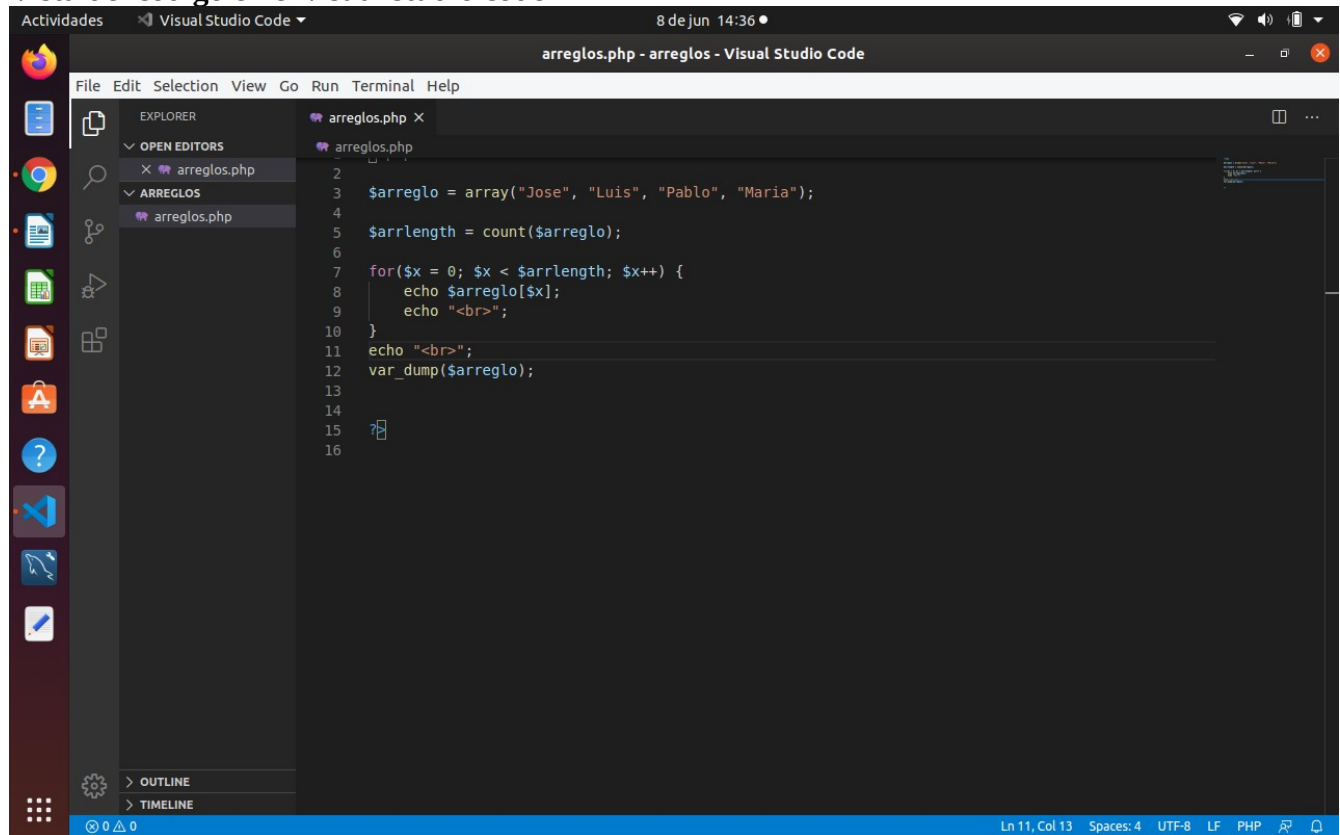
```
<?php
```

```
$arreglo = array("Jose", "Luis", "Pablo", "Maria");  
$arreglo = count($arreglo);
```

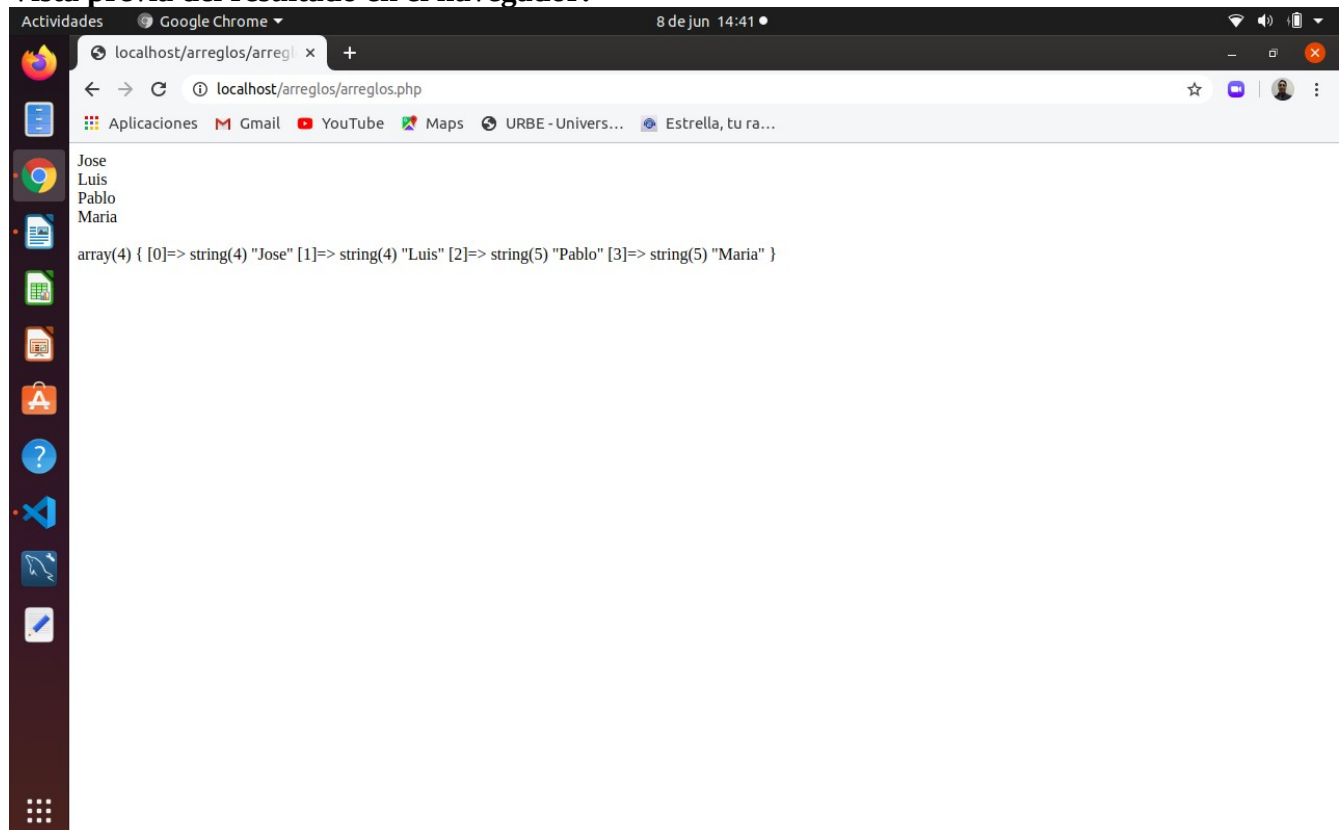
```
for($x = 0; $x < $arreglo; $x++) {  
    echo $arreglo[$x];  
    echo "<br>";  
}  
echo "<br>";  
var_dump($arreglo);
```

```
?>
```

Vista del código en el visual studio code



Vista previa del resultado en el navegador:



print_r()

Con esta palabra reservada no muestra las características del contenido del arreglo como lo son el índice del dato y su valor.

Codigo:

```
<?php
```

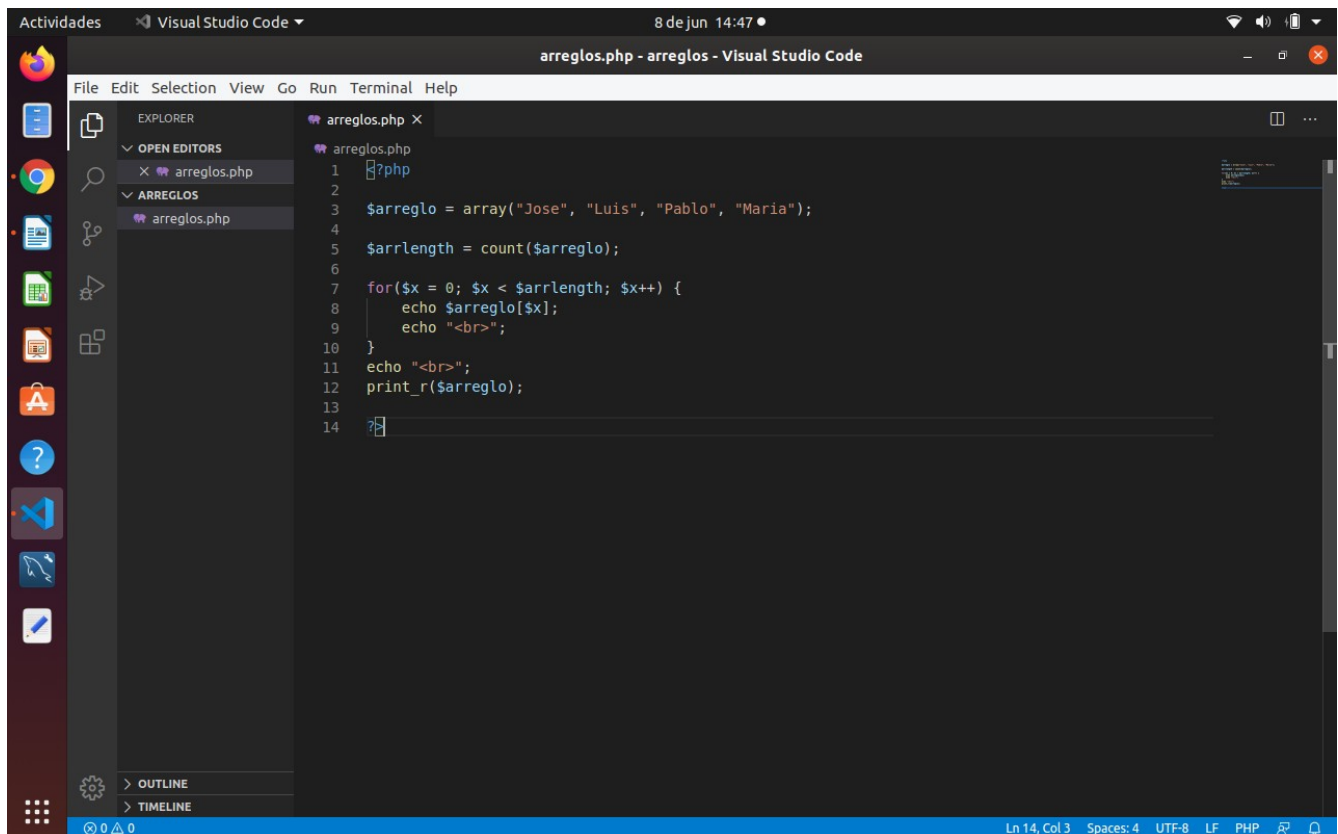
```
$arreglo = array("Jose", "Luis", "Pablo", "Maria");
```

```
$arlength = count($arreglo);
```

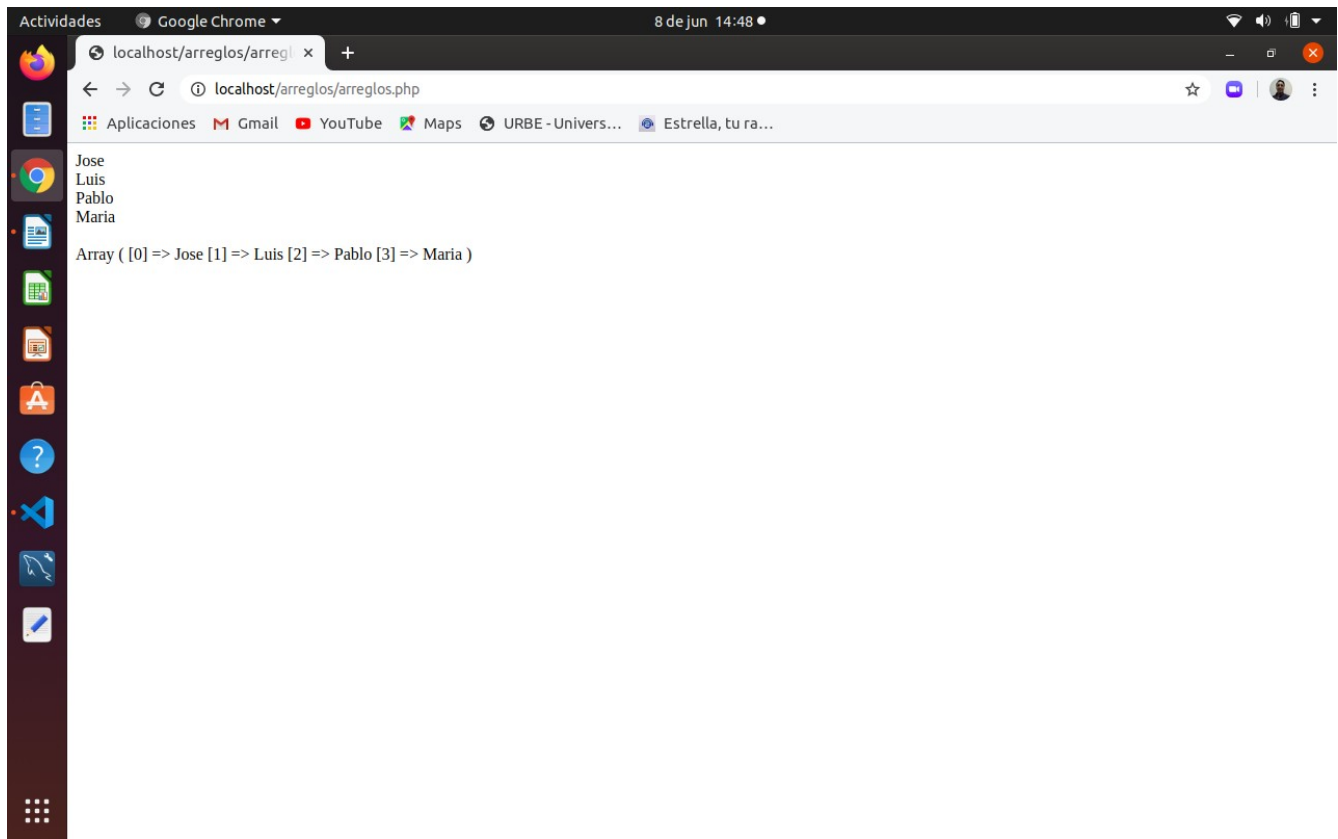
```
for($x = 0; $x < $arlength; $x++) {  
    echo $arreglo[$x];  
    echo "<br>";  
}  
echo "<br>";  
print_r($arreglo);
```

```
?>
```

Vista del código en el visual studio code



Vista previa del resultado en el navegador:



push_array()

Permite agregar elementos al arreglo, para ello necesita de dos parámetros, el primer parámetro es el nombre del arreglo y el segundo parámetro es el valor o valores a ingresar dentro del arreglo.

Código:

```
<?php

$arreglo = array("Jose", "Luis", "Pablo", "Maria");

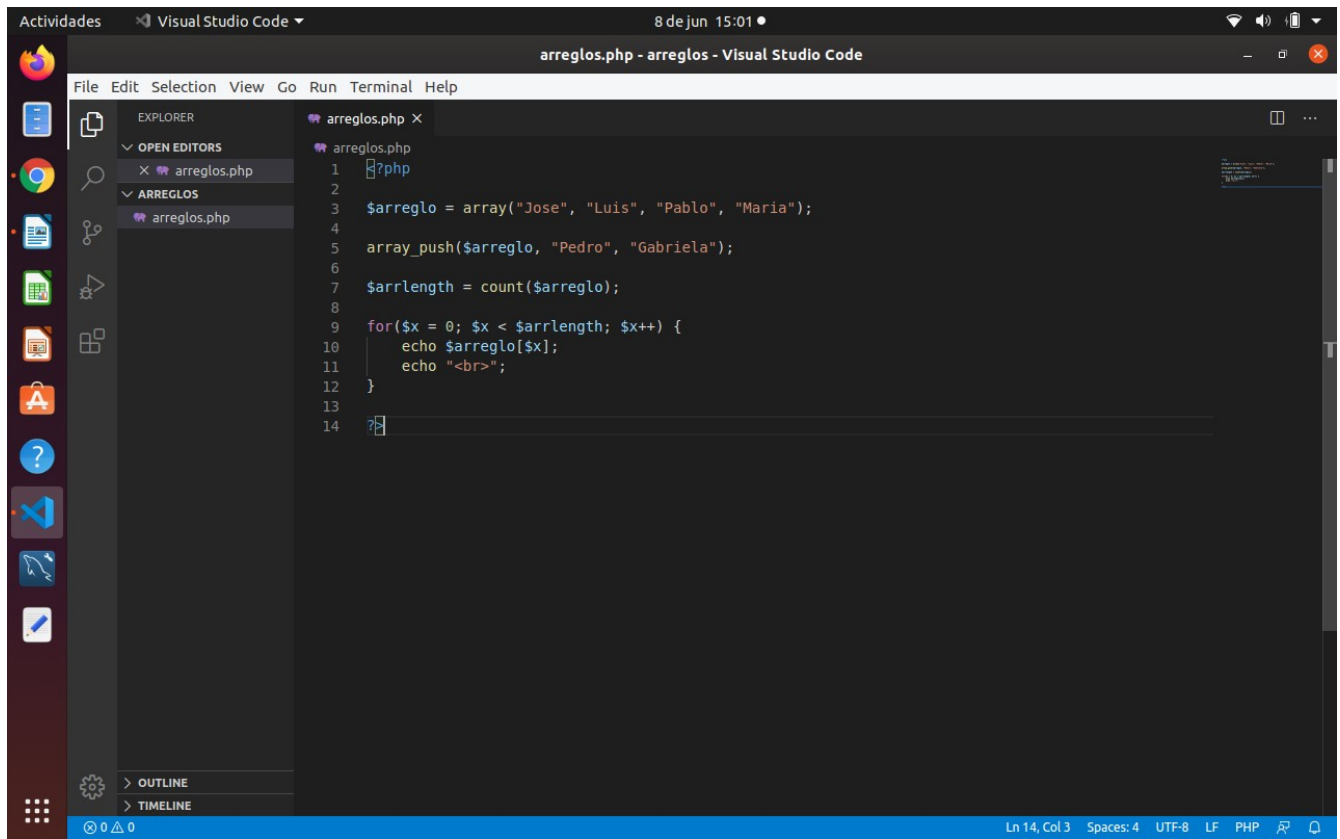
array_push($arreglo, "Pedro", "Gabriela");

$arlength = count($arreglo);

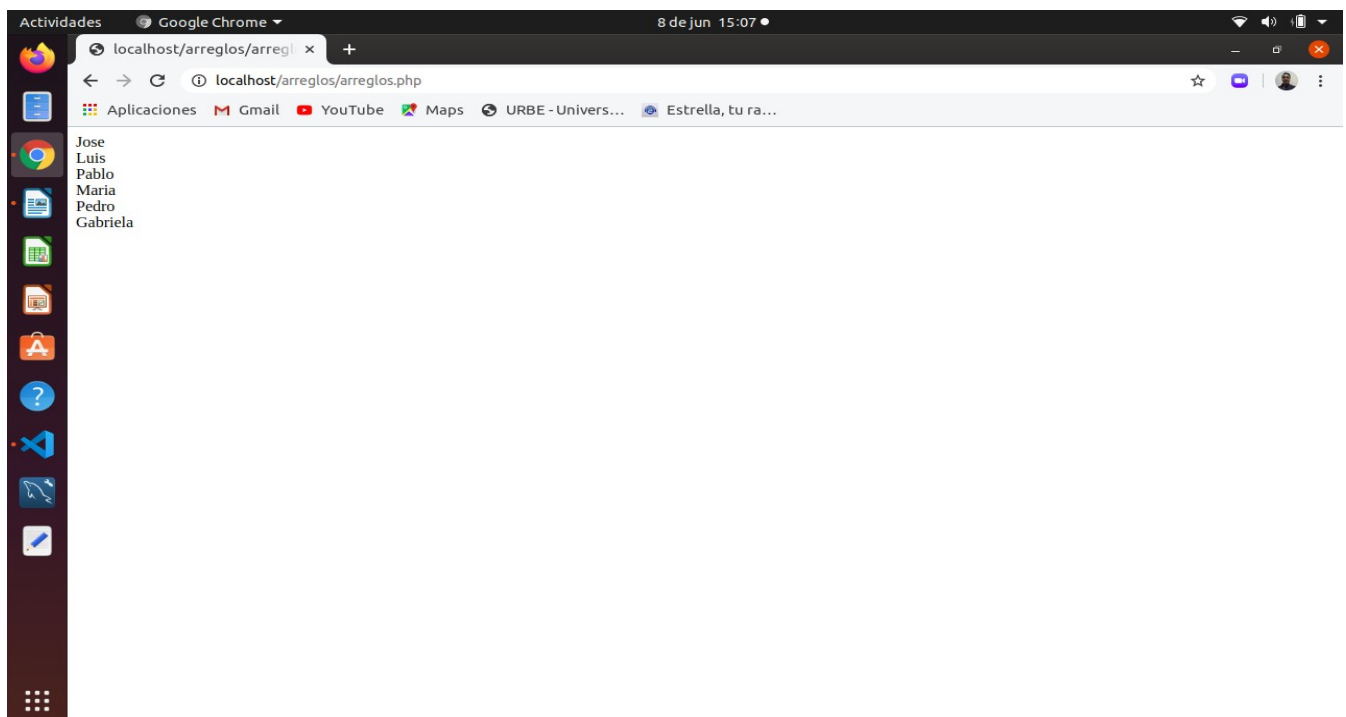
for($x = 0; $x < $arlength; $x++) {
    echo $arreglo[$x];
    echo "<br>";
}

?>
```

Vista del código en el visual studio code



Vista previa del resultado en el navegador:



unset()

Permite eliminar o borrar un elemento del arreglo, indicando para ello el nombre del arreglo con el respectivo índice del elemento a borrar.

Código:

```
<?php
```

```
$arreglo = array("Jose", "Luis", "Pablo", "Maria");
```

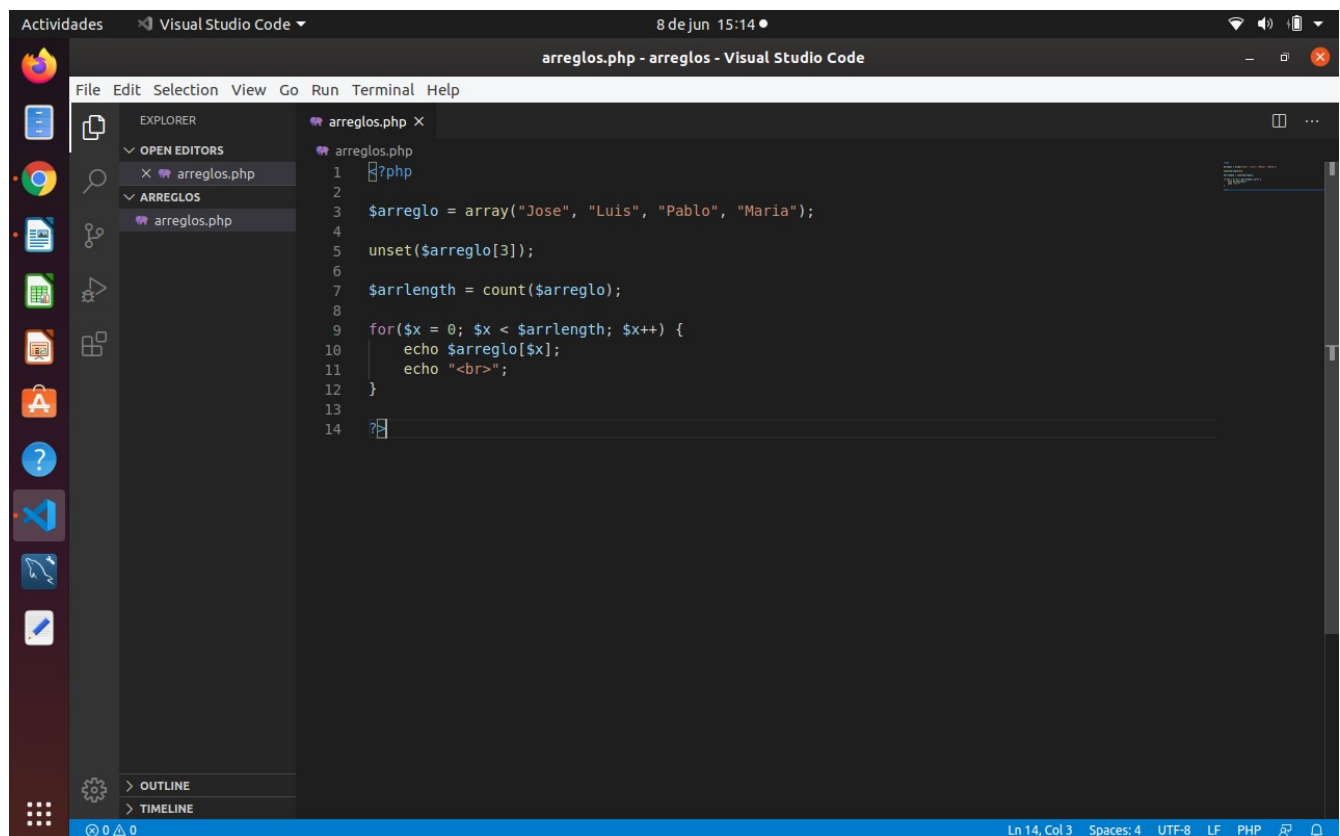
```
unset($arreglo[3]);
```

```
$arlength = count($arreglo);
```

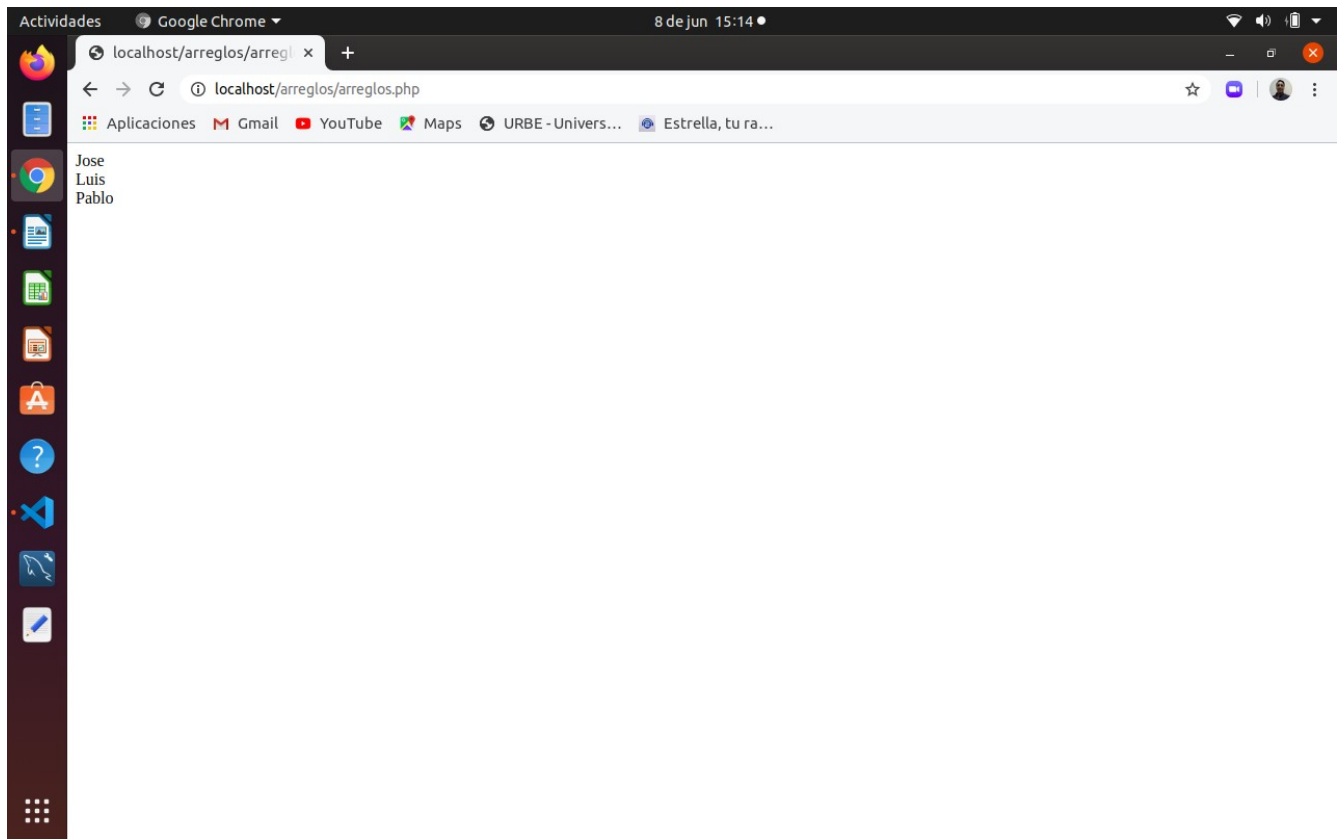
```
for($x = 0; $x < $arlength; $x++) {  
    echo $arreglo[$x];  
    echo "<br>";  
}
```

```
?>
```

Vista del código en el visual studio code



Vista previa del resultado en el navegador:



Otras palabras reservadas utilizadas en los arreglos

count()

Permite contar la cantidad de elementos que conforman el largo del arreglo.

sort()

ordena el array de manera ascendente, cogiendo los elementos de menor a mayor. Los parámetros son estos:

codigo:

```
sort(array, ordenarpor);
```

array (obligatorio): Es el array que queremos ordenar.

Ordenar por (opcional): Es el método de ordenación, dependiendo de los valores que haya en el array. Sus posibles valores son los siguientes:

0 = Orden por defecto comparando los valores sin importar de su tipo de valor.

1 = Orden numérico de los valores del array.

2 = Orden comparando los valores como si fuesen cadenas de texto.

3 = Orden por cadenas de texto usando la configuración regional del servidor.

4 = Orden de los elementos como cadenas de texto usando el «orden natural», es decir, ordenando el array por clave y valor.

Existe la función opuesta **rsort()** con la misma sintaxis que ordena el array de manera descendente, de mayor a menor.

asort()

ordena el array en orden ascendente, es decir, de menor a mayor ordenándolo por su valor. Vamos a ver los parámetros:

Código:

```
asort(array, ordenarpor);
```

array (obligatorio): Es el array que queremos ordenar.

Ordenar por (opcional): Es el método de ordenación, dependiendo de los valores que haya en el array. Sus valores son los mismos expuestos en la sintaxis anterior de **sort()**.

También existe la función opuesta, se llama **arsort()** y tiene la misma sintaxis, esta función ordena los arrays de forma descendente, de mayor a menor por el valor de los elementos.

ksort()

ordena el array asociativo en orden ascendente, es decir, de menor a mayor ordenándolo por su clave. En la reordenación no se pierde la relación clave->valor. Vamos a ver los parámetros:

Código:

```
ksort(array, ordenarpor);
```

array (obligatorio): Es el array que queremos ordenar.

ordenarpor (opcional): Es el método de ordenación, dependiendo de los valores que haya en el array. Sus valores son los mismos expuestos en la sintaxis anterior de **sort()**.

También existe la función opuesta, se llama **krsort()** y tiene la misma sintaxis, esta función ordena los arrays de forma descendente, de mayor a menor por las claves del array.

Arreglos Asociativos

Este es un array cuyos valores se asignan mediante claves

```
$variable = array(clave1=>valor1, clave2=>valor2, clave3=>valor3...);
```

\$variable: Elemento donde se quedará almacenado el array.

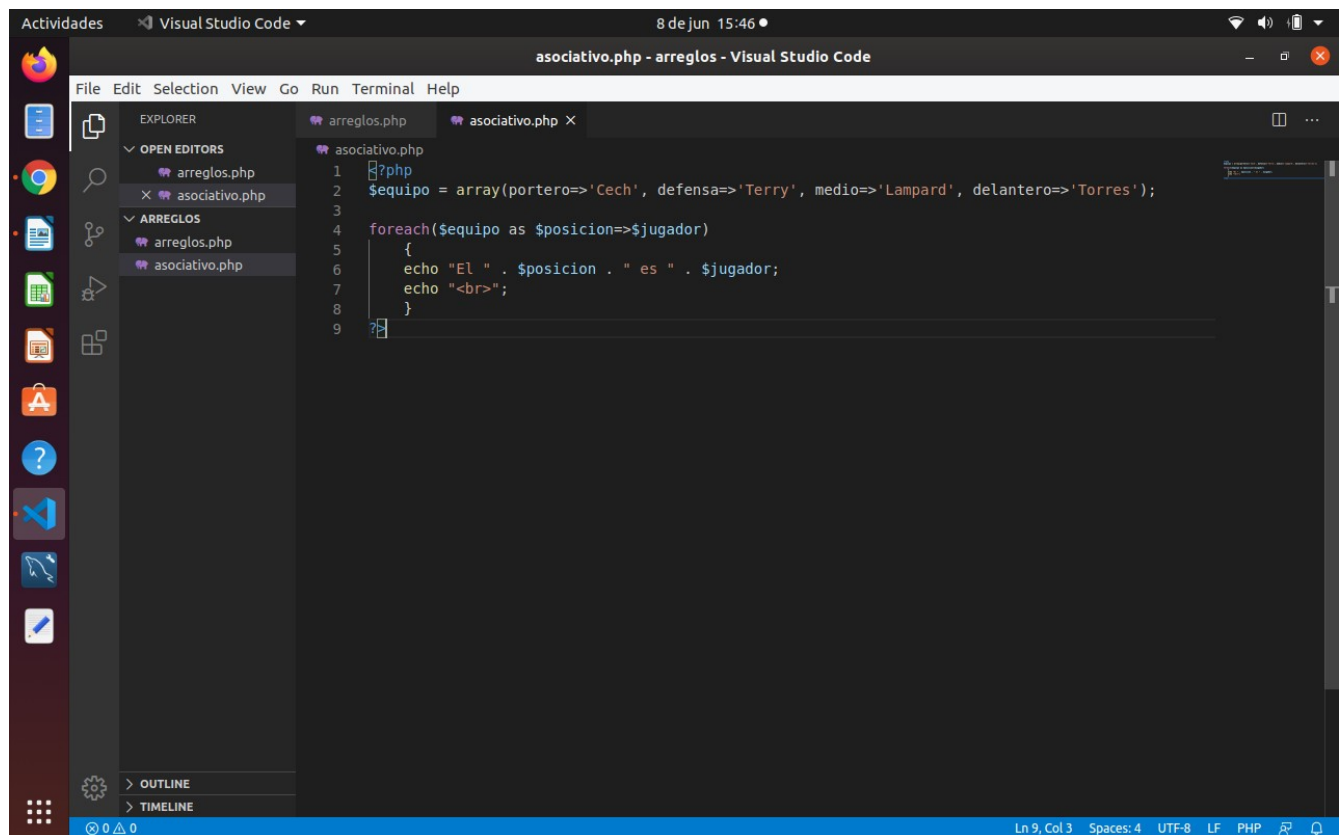
\$clave1, \$clave2 y \$clave3: Claves por las cuales se asignarán y asociarán los valores 1, 2 y 3.

Código:

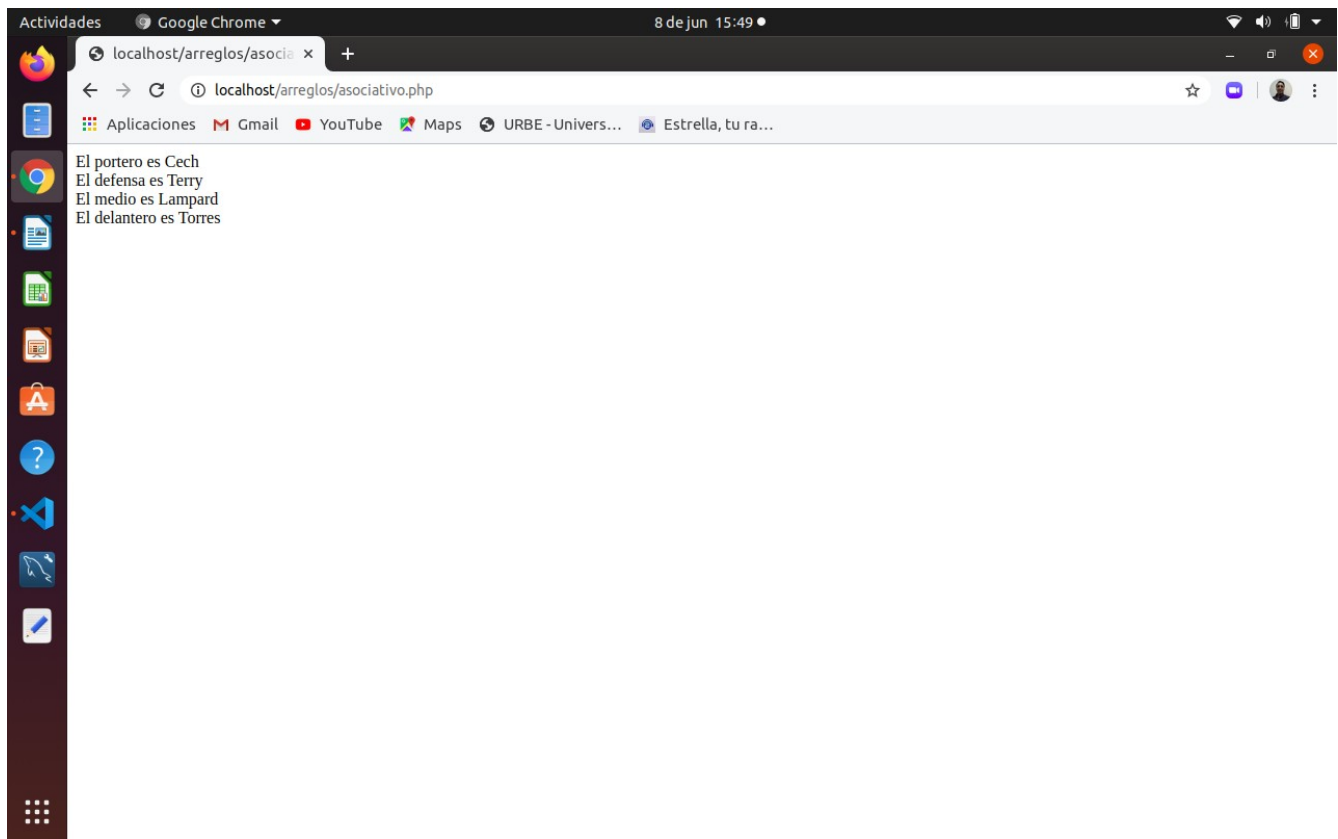
```
<?php
$equipo = array(portero=>'Cech', defensa=>'Terry', medio=>'Lampard',
delantero=>'Torres');

foreach($equipo as $posicion=>$jugador)
{
    echo "El " . $posicion . " es " . $jugador;
    echo "<br>";
}
?>
```

Vista del código en el visual studio code



Vista previa del resultado en el navegador:



Arreglo bidimensional (multidimensional)

Se puede resumir como un array de arrays, su sintaxis y un ejemplo de array bidimensional y cómo recorrerlo:

```
$equipo_futbol = array (
    array("Rooney","Chicharito","Gigs"),
    array("Suarez"),
    array("Torres","Terry","Etoo")
);
```

```
foreach($equipo_futbol as $equipo)
{
    echo "En este equipo juegan: ";
    foreach($equipo as $jugador)
    {
        echo $jugador ." ";
    }
    echo "<br>";
}
```

`$equipo_futbol`: Es un array contenedor de otros 3 arrays con jugadores de futbol. Luego recorro el array con PHP `foreach()` `$equipo_futbol` y a su vez recorro cada array que encuentro para sacar los jugadores.

En caso de necesitar acceder a un elemento en cuestión, este es el código:

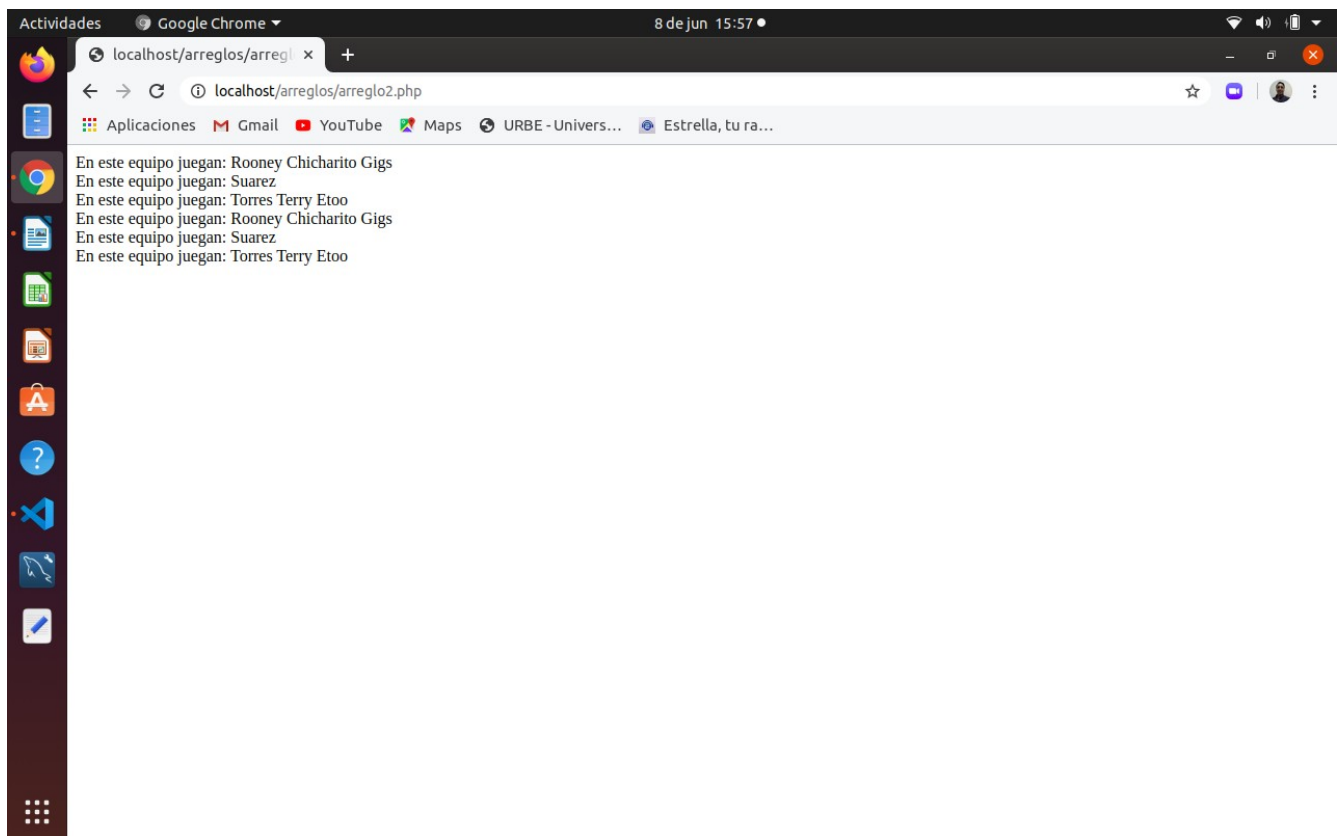
De esta manera hemos accedido al jugador 2 del primer equipo.

The image shows a screenshot of the Visual Studio Code editor interface. At the top, the title bar reads "arreglo2.php - arreglos - Visual Studio Code". The main editor area displays a PHP file named "arreglo2.php" with the following code:

```
1 <?php
2
3 $equipo_futbol = array
4     (
5         array("Rooney", "Chicharito", "Gigs"),
6         array("Suarez"),
7         array("Torres", "Terry", "Etoo")
8     );
9
10 foreach($equipo_futbol as $equipo)
11 {
12     echo "En este equipo juegan: ";
13     foreach($equipo as $jugador)
14     {
15         echo $jugador . " ";
16     }
17     echo "<br>";
18 }
19 $equipo_futbol = array
20     (
21         array("Rooney", "Chicharito", "Gigs"),
22         array("Suarez"),
23         array("Torres", "Terry", "Etoo")
24     );
25
26 foreach($equipo_futbol as $equipo)
27 {
28     echo "En este equipo juegan: ";
29     foreach($equipo as $jugador)
30     {
31         echo $jugador . " ";
32     }
33     echo "<br>";
34 }
35
36 ?>
```

The left sidebar contains the Explorer view, which shows the file structure with "arreglo2.php" selected. The top bar includes the "File", "Edit", "Selection", "View", "Go", "Run", "Terminal", and "Help" menus. The bottom status bar shows "Ln 15, Col 28", "Spaces: 4", "UTF-8", "LF", "PHP", and a search icon.

Vista previa del resultado en el navegador:



Arreglo tridimensional (multidimensional)

Al igual que el anterior es un array de arrays, en este caso es de 3 dimensiones. Aquí el ejemplo:

```
$datos = array(  
    array(array(0, 0, 0),  
        array(0, 0, 1),  
        array(0, 0, 2)  
    ),  
    array(array(0, 1, 0),  
        array(0, 1, 1),  
        array(0, 1, 2)  
    ),  
    array(array(0, 2, 0),  
        array(0, 2, 1),  
        array(0, 2, 2)  
    )  
);
```

Como se observa, en la matriz del ejemplo hay una profundidad de 3 arrays, es decir, el primero contiene un segundo y el segundo de 3 terceros.

el código necesario en este ejemplo para recorrer el array tridimensional:

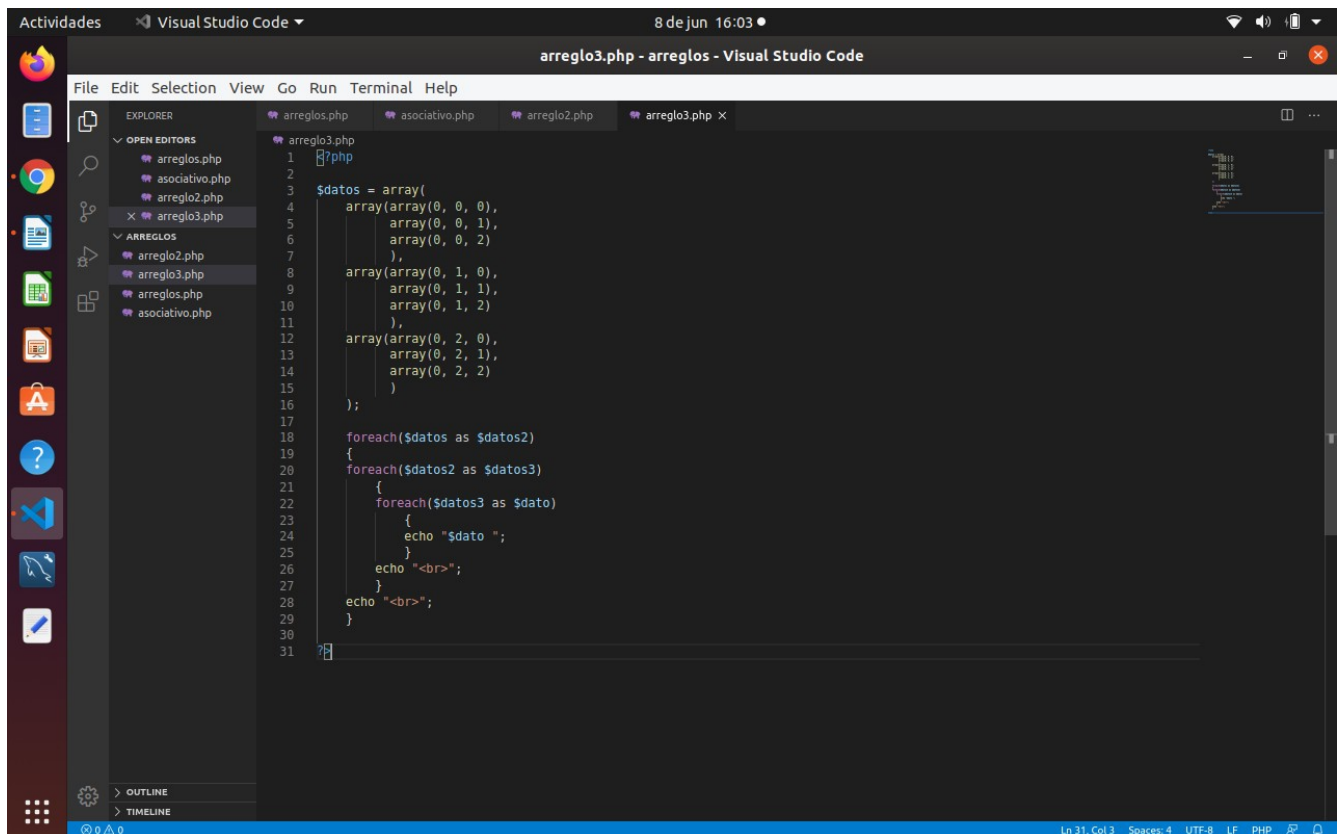
```

foreach($datos as $datos2)
{
    foreach($datos2 as $datos3)
    {
        foreach($datos3 as $dato)
        {
            echo "$dato ";
        }
        echo "<br>";
    }
    echo "<br>";
}

```

En caso de necesitar acceder a un elemento concreto:
`$datos[0][2][1];`

Vista del código en el visual studio code



Vista previa del resultado en el navegador:

