	<b>UNIVERSIDAD DON BOSCO</b> <b>FACULTAD DE INGENIERÍA</b> <b>ESCUELA DE COMPUTACIÓN</b>
<b>CICLO 01-2024</b>	<b>GUIA DE LABORATORIO N° 4</b>
	<b>Nombre de la práctica:</b> Arreglos <b>Lugar de ejecución:</b> Laboratorio de Informática <b>Tiempo estimado:</b> 2 horas <b>Materia:</b> Lenguajes Interpretados en el Servidor

## I. COMPETENCIA

- Desarrolle la capacidad de utilizar eficazmente matrices para almacenar valores y acceder a sus valores en los scripts PHP.
- Sea capaz de utilizar estructuras de control repetitivas para trabajar con datos procedentes de una matriz escalar o asociativa.
- Adquiera el dominio de matrices unidimensionales y multidimensionales utilizando ciclos o lazos repetitivos anidados.
- **Haga uso de diversas funciones para manejo de matrices con PHP.**

## II. INTRODUCCIÓN TEORICA

### Matrices en PHP

Una matriz o arreglo es un tipo de dato compuesto que permite almacenar bajo un mismo nombre un conjunto de valores que pueden ser de tipos de datos diferentes. Cuando se define una matriz se hace de forma similar a las variables escalares con la diferencia de que la asignación de valores se debe realizar a cada una de las posiciones del arreglo. También se puede utilizar en lugar de la asignación directa estructuras de control como ciclos repetitivos o funciones para asignar varias posiciones de una vez, como veremos más adelante.



### Clasificación de las matrices

En PHP se pueden clasificar las matrices de dos formas:

- Por el tipo de índice que utilizan. Este índice se utiliza para hacer referencia a un elemento específico del arreglo— y,
- Por el número de dimensiones de la matriz. Esto es, cuántos índices son necesarios para hacer referencia a un elemento específico.

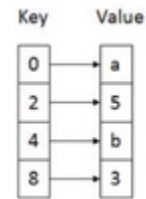
### Tipos de matrices según el tipo de índice:

- Matrices escalares

También se les conoce como matrices enumeradas o indexadas, por el hecho que utilizan números enteros como índices. En PHP el índice para el primer elemento de un arreglo empieza en cero y los siguientes elementos van aumentando en uno este valor de índice, de modo que el segundo elemento tiene índice uno, el tercero índice dos, etc.

Por ejemplo, se puede tener una matriz llamada \$numeros con cinco elementos. Para asignar un valor a cada uno de los elementos de este arreglo, habría que escribir:

```
$numeros[0] = 1;  
$numeros[1] = 2;  
$numeros[2] = 3;  
$numeros[3] = 4;  
$numeros[4] = 5;
```



Luego, si deseamos mandar a imprimir el tercer elemento de este arreglo, tendríamos que escribir:

```
echo $numero[2];
```

Ahora bien, si lo que deseamos es imprimir todos los elementos de la matriz, la mejor solución sería utilizar una sentencia repetitiva como esta:

```
for($i=0; $i<count($numeros); $i++){  
    echo $numeros[$i] . "<br />\n";}
```

#### **b) Matrices asociativas**

Estas matrices también son conocidas como tablas hash. Se caracterizan por utilizar una cadena de caracteres como índice de la matriz. Lo normal es utilizar una cadena que permita asociar fácilmente el índice con el valor que almacena el elemento.

Por ejemplo, se puede crear una matriz llamada \$monedas con cuatro elementos y asociar valores a cada uno de estos elementos de la siguiente forma:

```
$monedas["El Salvador"] = "dólar";  
$monedas["Guatemala"] = "quetzal";  
$monedas["Costa Rica"] = "colón";  
$monedas["Honduras"] = "lempira";
```

Si deseamos mandar a imprimir los valores de este arreglo, deberíamos escribir:

```
echo $monedas["El Salvador"];  
echo $monedas["Guatemala"];  
echo $monedas["Costa Rica"];  
echo $monedas["Honduras"];
```

O, haciendo uso de estructuras de control repetitivas, con la sentencia foreach:

```
foreach($monedas as $key => $value){  
    echo "Moneda de " . $key . ": " . $value . "<br />\n";  
}
```

Tipos de matrices según el número de dimensiones:

#### **a) Matrices unidimensionales**

Es una matriz que utiliza un único índice para hacer referencia a todos los elementos de la misma. Este índice puede ser escalar o asociativo.

#### **b) Matrices multidimensionales**

Es una matriz que necesita utilizar más de un índice para hacer referencia a un elemento específico. De todas las matrices multidimensionales, las más utilizados son las de dos dimensiones. Los arreglos de tres o más dimensiones no se utilizan tanto debido a que son más difíciles de comprender y manejar.

El siguiente bloque de código muestra la definición de un arreglo multidimensional (con índices asociativos) y luego manda a imprimir en la ventana del navegador uno de sus elementos:

```
<?
$pais=array
(
"espana" =>array
(
"nombre"=>"España",
"lengua"=>"Castellano",
"moneda"=>"Peseta"
),
"francia" =>array
(
"nombre"=>"Francia",
"lengua"=>"Francés",
"moneda"=>"Franco"
)
);
echo $pais["espana"]["moneda"]; //Imprime en la ventana del navegador: "Peseta"
?>
```

### **Creación de matrices en PHP**

En PHP, las matrices o arreglos se pueden crear asignando los valores a los elementos directamente, tal y como se ha hecho en los ejemplos anteriores, o se pueden utilizar funciones para ese propósito. Las funciones que permiten crear arreglos en PHP son: array() y range().

La sintaxis de cada una de ellas se muestra a continuación:

```
$arreglo = array($valor1, $valor2, ... , $valorn);
$matriz = range($valorinicial, $valorfinal, $desplazamiento);
```

La función array() se utiliza para crear una variable de tipo matriz y asignarle los valores que va a contener en una sola instrucción. En tanto que, con range() se puede crear una matriz que comienza desde el \$valorinicial y termina con el \$valorfinal.

Asignar los elementos de una matriz a variables escalares PHP proporciona una función que permite asignar en una sola instrucción los elementos de una matriz a variables escalares independientes. Esta función es list(), que posee como argumentos las variables escalares en donde se almacenarán los valores almacenados en las posiciones del arreglo de forma ordenada. La sintaxis de la función es la que se muestra a continuación:

```
list ($variable1, $variable2, ... , $variablen) = $mi_array
```

Donde,

\$mi\_array, es la matriz o arreglo del que se extraerán los elementos.

\$variable1, \$variable2, ... , \$variablen, son las variables escalares independientes en las que se asignarán los diferentes elementos de la matriz.

### Funciones importantes para trabajar con matrices

El lenguaje PHP, brinda una cantidad considerable de funciones para facilitar el trabajo con matrices. A continuación se brinda información detallada de varias de ellas:

Funciones para recorrer una matriz Cuando se trabaja con matrices, PHP ofrece una serie de funciones que permiten realizar de forma más fácil y conveniente esta tarea.

**count()**: Permite obtener el número de elementos que componen la matriz. Es muy útil cuando se pretende realizar un recorrido sobre una matriz utilizando ciclos for, while o do-while. Por ejemplo:

```
for($i=0; $i<count($matriz); $i++)
```

**sizeof()**: También se utiliza para obtener el número de elementos de una matriz. Por ejemplo:

```
for($i=0; $i<sizeof($matriz); $i++)
```

**next()**: Devuelve el valor del elemento siguiente al actual (si existe), avanzando el puntero interno una posición. En caso de que el elemento actual sea el último de la matriz, devuelve false.

**prev()**: Devuelve el valor del elemento anterior al actual (si existe), retrocediendo el puntero interno una posición. En caso de que el elemento actual sea el primero, devuelve false.

**current()**: Devuelve el valor del elemento situado en la posición actual del cursor. Si la matriz está vacía o no hay más elementos, la función devolverá false.

**key()**: Devuelve el índice del elemento situado en la posición actual del cursor. Si la matriz está vacía, o no hay más elementos, devuelve la constante NULL.

**reset()**: Sitúa el cursor interno de la matriz en el primer elemento de la misma y devuelve el valor de dicho elemento. Si la matriz está vacía, la función devuelve el valor false.

**end()**: Avanza el cursor interno de la matriz hasta el último elemento de la misma y devuelve el valor de dicho elemento. Si la matriz está vacía devuelve false.

**each()**: Se usa para recorrer matrices (especialmente matrices asociativas), devolviendo simultáneamente el índice y el valor asociado al índice actual de un elemento. Además, avanza el puntero interno de la matriz hacia el siguiente elemento. Si el puntero interno apunta a la última posición de la matriz, la función devuelve false.

**list()**: Asigna una lista de variables en una sola operación. Suele utilizarse en combinación con la función each().

OBSERVACIÓN: El argumento de todas las funciones anteriores es una matriz. A excepción de la función list().

Ejemplo:

```
$matriz = array("Primavera", "Verano", "Otoño", "Invierno");
do{
    $index = key($matriz);
    $value = current($matriz);
```

```
echo $index . ": " . $value . "<br>\n";
}while(next($matriz));
```

### III. MATERIALES Y EQUIPO

Para la realización de la guía de práctica se requerirá lo siguiente:

No.	Material	Cantidad
1	Guía de práctica #5: Arreglos	1
2	Computadora con WampServer instalado y funcionando correctamente	1
3	Editor PHP sublime Text o Eclipse PHP	1
4	Memoria USB o disco flexible	1

### IV. PROCEDIMIENTO

Indicaciones: Asegúrese de digitar el código de los siguientes ejemplos que se presentan a continuación. Tenga en cuenta que el PHP Designer no es compilador solamente un editor por lo tanto los errores de sintaxis los podrá observar únicamente hasta que se ejecute el script al cargar la página en el navegador de su preferencia.

**Ejemplo 1:** El siguiente ejemplo muestra cómo utilizar un array asociativo bidimensional para llevar las notas de los alumnos de tres materias: LIS, POO y BDI:

Archivo 1: ejemplo1.php

```
<?php
$materias = array(
    "Lenguajes Interpretados en el Servidor" => array(
        'Quique Setién' => 8.6,
        'Lionel Messi' => 4.9,
        'Nélson Semedo' => 7.9,
        'Gerard Piqué' => 6.2,
        'Clément Lenglet' => 7.5,
        'Jordi Alba' => 9.8,
        'Sergi Roberto' => 8.0,
        'Samuel Umtiti' => 7.6,
        'Júnior Firpo' => 9.5,
        'Ivan Rakitic' => 8.7,
        'Sergio Busquets' => 9.4,
        'Ousmane Dembélé' => 8.5
    ),
    "Programación Orientada a Objetos I" => array(
        'Zinedine Zidane' => 5.5,
        'Vinicius Paixao de Oliveira Junior' => 6.4,
```

```

        'Rodrygo Goes' => 8.3,
        'Daniel Carvajal Ramos' => 7.5,
        'Éder Gabriel Militão' => 9.6,
        'Sergio Ramos García' => 7.5,
        'Raphaël Varane' => 5.8,
        'José Ignacio Fernández Iglesias' => 9.2,
        'Marcelo Vieira da Silva' => 5.0,
        'Toni Kroos' => 6.0,
        'Luka Modric' => 8.2,
        'James Rodríguez' => 8.5,
        'Gareth Bale' => 6.2
    ),
    "Base de Datos I" => array(
        'Peter Parker' => 6.0,
        'Wade Wilson' => 6.8,
        'Tony Stark' => 5.3,
        'Steve Roggers' => 7.0,
        'Bruce Banner' => 4.3,
        'Thor Odinson' => 7.5,
        'Estephen Strange' => 9.6,
        'Wanda Maximoff' => 7.5,
        'Scott Lang' => 5.8,
        'natasha romanoff' => 6.8,
        'Erik Lehnsherr' => 6.0,
        'Matt Murdock' => 8.2,
        'Peter Quill' => 2.8,
        'Carol Danvers' => 6.2
    )
);

//Creando la página web con cadenas usando
//la sintaxis HERE DOC
echo "<!DOCTYPE html>";
echo "<html>";
echo "<head>";
echo "<title>Matriz</title>";
echo "<meta charset='utf-8' />";
echo '<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css">';
echo '<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>';
echo '<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>';
echo "</head>";
echo "<body>";
echo " <div class='container'>";
echo "<header class='navbar navbar-light bg-light'>";
echo "<a class='navbar-brand' class='d-inline-block align-top' href='#'>";
echo "<img src='./img/udb.jpg' style='width: 8rem; height:8rem;'>";

```


```

echo " Notas de los estudiantes";
echo"</a>";
echo "</header>";
echo "<body>";
foreach($materias as $materia => $notas){

    echo "<table class='table table-bordered'>";
    echo "<thead>";
    echo " <tr class='bg-secondary'>";
    echo "     <th scope='col' colspan='2' class='text-
center'>". $materia ."</th>";
    echo "     </tr>";
    echo " <tr class='bg-success'>";
    echo "     <th scope='col'>Nombre</th>";
    echo "     <th scope='col'>Nota</th>";
    echo "     </tr>";
    echo "</thead>";
    echo "<tbody>";
    //Variable contador para determinar
    //cuántos alumnos en total hay en cada materia
    $totalalumnos = 0;
    $sumanotas = 0;
    foreach($notas as $nombre => $nota){
echo "     <tr>";
echo "         <td>$nombre</td>";
echo "         <td> ". number_format($nota, 1, '.', ',') ."</td>";
echo "     </tr>";
        //Sumar la nota al acumulador de las notas
        $sumanotas += $nota;
        //Aumentar el contador de alumnos
        $totalalumnos++;
    }
    //Obtener la suma total
    //de las notas entre el total de notas
    //en la materia procesada actualmente
    $sumanotas /= $totalalumnos;
    echo "<tr class='bg-success' >";
    echo "     <td colspan='2'>PROMEDIO: " . number_format($sumanotas,'2'
, '.', ','). " </td>";
    echo "</tr> ";
    echo "</tbody>";
    echo "</table>";
}
echo"</div>";
echo "</body>\n";
echo "</html>\n";
?>

```

Vista en pantalla:

 Notas de los estudiantes	
Lenguajes Interpretados en el Servidor	
Nombre	Nota
Quique Setién	8.6
Lionel Messi	4.9
Nélson Semedo	7.9
Gerard Piqué	6.2
Clément Lenglet	7.5
Jordi Alba	9.8
Sergi Roberto	8.0
Samuel Umtiti	7.6

**Ejemplo 2:** El siguiente ejercicio permite crear una serie de letras o números dependiendo de la selección del usuario en un control de menú desplegable. Para crear la serie generado se utiliza la función `range()`, proporcionando todos argumentos: el valor inicial y final de la serie, así como, el desplazamiento para la secuencia de cada elemento.

Archivo: ejemplo2.php

```
<!DOCTYPE html>
<html lang="es">
<head>
  <title>Uso de la función range</title>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>
</head>
</body>
<div class="container">
<header class='navbar navbar-light bg-light'>
<a class='navbar-brand' class='d-inline-block align-top' href='#'>
  <h4>Generación de series con función de matrices</h4>
</a>
</header>
<form action="<?=$_SERVER['PHP_SELF']; ?>" method="POST" class="formoid-solid-purple">
<div class="form-group">
  <label>Tipos de serie</label>
```



```

        <select class="form-control" name="seltipo">
            <option value="AlfabetoMin" selected="selected">
                Alfabeto en minúsculas
            </option>
            <option value="AlfabetoMay">
                Alfabeto en mayúsculas
            </option>
            <option value="NumerosPares">
                Números pares
            </option>
            <option value="NumerosImpares">
                Números impares
            </option>
        </select>
    </div>
    <div class="submit">
        <input type="submit" name="enviar" value="Enviar" class="btn btn-
primary" />
    </div>
</form>
<?php
    if(isset($_POST['enviar'])){
        $tiposecuencia = isset($_POST['seltipo']) ? $_POST['seltipo'] : "";
        switch($tiposecuencia){
            case "AlfabetoMin":
                $inicio = 'a';
                $final = 'z';
                $salto = '1';
                break;
            case "AlfabetoMay":
                $inicio = 'A';
                $final = 'Z';
                $salto = '1';
                break;
            case "NumerosPares":
                $inicio = '0';
                $final = '50';
                $salto = '2';
                break;
            case "NumerosImpares":
                $inicio = '1';
                $final = '50';
                $salto = '2';
                break;
        }
        $secuencia = range($inicio,$final,$salto);
        echo "<div id=\"box\">";
        foreach($secuencia as $letra)
            echo "<span class=\"caracter\">" . $letra . "</span>&nbsp;&nbsp;&nbsp;\n";
    }
}

```

```

        echo "</div>";
    }
?>
</div>
</body>
<script src="//ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js"></script>
<script>window.jQuery || document.write('<script src="js/jquery.min.js"></script>')</script>
<script src="js/formoid-solid-purple.js"></script>
</html>

```

Vista en pantalla:

...

Generación de series con función de matrices

Tipos de serie

Alfabeto en minúsculas

Enviar

0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50

Ejemplo 3: En este ejemplo se ingresan una serie de productos a través de un formulario. Los nuevos productos se ingresan en un cuadro de texto y mediante un botón son agregados a un cuadro de lista. Mediante un botón de envío los datos ingresados y seleccionados del cuadro de lista son enviados para que sean procesados por el script PHP. La aplicación incluye un script .js que se encarga de la funcionalidad de agregar productos al cuadro de lista.

Archivo: ejemplo3.html

```

<!DOCTYPE html>
<html lang="es">
<head>
    <title>Listado de productos</title>
    <meta charset="utf-8" />
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css">'
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>'
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>'
    <script src="js/productos.js"></script>

```

```

</head>
<body>
  <div class="container">
<form name="frmproductos" action="productos.php" method="POST">

  <fieldset class="form-group">
    <legend><span>Ingreso de productos:</span></legend>

    <div class="form-group">
      <label for="producto">Nuevo producto:</label>
      <input type="text" class="form-
control" name="producto" id="producto" maxlength="60" placeholder="(Ingrese
un producto)" />
      <input type="reset" name="agregar" id="agregar" value="Agreg
ar" class="btn btn-primary" />
    </div>
    <div class="form-group">
      <label for="ingresados">Productos ingresados:</label>
      <select name="ingresados[]" class="form-
control" id="ingresados" size="6" multiple="multiple"></select>
    </div>

    <input type="submit" name="enviar" id="enviar" value="En
viar" class="btn btn-primary" />
  </fieldset>
</div>
</form>
</div>
</body>
</html>

```

Archivo 2: productos.php

```

<!DOCTYPE html>
<html lang="es">
<head>
  <title>Productos de la tienda</title>
  <meta charset="utf-8" />
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstra
p/4.4.1/css/bootstrap.min.css">'
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.m
in.js"></script>'
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstra
p.min.js"></script>'
  <script src="js/modernizr.custom.lis.js"></script>
</head>
<body>
<section>
<article>
<table class='table table-bordered'>

```

```

<thead>
  <th class="head">Productos</th>
</thead>
<tbody>
  <?php
    if(isset($_POST['enviar'])){
      if(isset($_POST['ingresados'])){
        foreach($_POST["ingresados"] as $productos){
          echo "<tr>\n<td>$productos</td>\n</tr>";
        }
      }
    }
  ?>
</tbody>
</table>
</article>
</section>
</body>
</html>

```

Vista en pantalla:

Ingreso de productos:

Nuevo producto:



Productos ingresados:

Manzana
Pera
Uva
Melocotón
Fresa

...

Productos
Pera
Uva
Fresa

Ejemplo 4: El siguiente ejemplo muestra el uso de la función `array_rand()` de PHP que permite generar un número de índices aleatorios para mostrar las imágenes de películas de la portada. Luego, se puede seleccionar un nombre de película y la cantidad de boletos de entrada que se desean adquirir. Al enviar el formulario y utilizando una matriz bidimensional asociativa se puede crear un boleto que indica el precio a pagar por el número de entradas adquirido.

Archivo 1: ejemplo4.php

```
<!DOCTYPE html>
<html lang="es">
<head>
  <title>Cartelera de cine</title>
  <meta charset="utf-8" />
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>
  <script src="js/modernizr.custom.lis.js"></script>
</head>
<body>
<div class="container">
<header class='navbar navbar-dark bg-dark'>
<a class='navbar-brand' class='d-inline-block align-top' href='#'>
Cartelera de Cine Bosco
</a>
</header>
<section>
<?php
  //Creamos una variable con el directorio o carpeta que contiene las imágenes
```

```

$dir = "img";
//Creamos una matriz con los nombres de las imágenes que se almacenan en
el directorio
$peliculas = array(
    "al-diablo-con-el-diablo.jpg", "click.jpg",
    "cruzada.jpg", "efecto-mariposa.jpg",
    "busca-de-la-felicidad.jpg", "amenaza-fantasma.jpg",
    "la-milla-verde.jpg", "la-propuesta.jpg",
    "comunidad-del-anillo.jpg", "sexto-
sentido.jpg", "Avengers.jpg", "Parasite.jpg"
);
//Obtener cuatro claves de la matriz $peliculas
$claves = array_rand($peliculas, 4);
//Creando una estructura de 3 columnas con elementos DIV
?>

<form action="pagoentrada.php" method="POST">
<fieldset class="form-group">
    <legend><span>Entrada al cine</span></legend>
    <label for="pelicula">Película</label>
    <select class="form-control" name="pelicula" id="pelicula">
        <option value="mov0001" selected="selected">
            Al diablo con el diablo
        </option>
        <option value="mov0000">
            Avengers
        </option>
        <option value="mov0002">
            Click
        </option>
        <option value="mov0003">
            Cruzada
        </option>
        <option value="mov0004">
            El efecto mariposa
        </option>
        <option value="mov0005">
            En busca de la felicidad
        </option>
        <option value="mov0006">
            La amenaza fastasma
        </option>
        <option value="mov0007">
            Milagros inesperados
        </option>
        <option value="mov0008">
            La propuesta
        </option>
        <option value="mov0009">
            La comunidad del anillo
    </select>

```

```

        </option>
        <option value="mov0010">
            El sexto sentido
        </option>
        <option value="mov0011">
            Parasite
        </option>

    </select><br />
    <label for="cantidad">Cantidad</label>
    <input class="form-
control" type="text" name="cantidad" id="cantidad" maxlength="1" placeholder
="Cantidad" pattern="\d{1}" required /><br />
    <input type="submit" name="comprar" value="Comprar" class="btn b
tn-primary" />
</fieldset>
</form>
<div id="carouselExampleSlidesOnly" class="carousel slide" data-
ride="carousel">
    <div class="carousel-inner">
        <div class="carousel-item active">
            " /><br />
        </div>
        <div class="carousel-item">
            " />
        </div>
        <div class="carousel-item">
            " />
        </div>
        <div class="carousel-item">
            " />
        </div>
    </div>
</div>
</div>
</body>
</html>

```

Archivo 2: pagoentrada.php

```

<!DOCTYPE html>
<html>
<head>

```

```
<title>Boleto de entrada</title>
<meta charset="utf-8" />
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css">'
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>'
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>'
<script src="js/modernizr.custom.lis.js"></script>
</head>
<body>
<?php
    $peliculas = array(
        "mov0000" => array(
            "titulo" => "Avengers",
            "precio" => 3.75,
            "sala" => "Sala 1",
            "hora" => "7:00 pm"
        ),
        "mov0001" => array(
            "titulo" => "Al diablo con el diablo",
            "precio" => 3.75,
            "sala" => "Sala 1",
            "hora" => "7:00 pm"
        ),
        "mov0002" => array(
            "titulo" => "Clic",
            "precio" => 3.75,
            "sala" => "Sala 3",
            "hora" => "5:00 pm"
        ),
        "mov0003" => array(
            "titulo" => "Cruzada",
            "precio" => 3.00,
            "sala" => "Sala 2",
            "hora" => "1:00 pm"
        ),
        "mov0004" => array(
            "titulo" => "El efecto mariposa",
            "precio" => 3.75,
            "sala" => "sala 3",
            "hora" => "6:00 pm"
        ),
        "mov0005" => array(
            "titulo" => "En busca de la felicidad",
            "precio" => 3.00,
            "sala" => "Sala 5",
            "hora" => "3:00 pm"
        ),
    ),
```



```

        "mov0006" => array(
            "titulo" => "La amenaza fastasma",
            "precio" => 3.75,
            "sala" => "sala 4",
            "hora" => "6:00 pm"
        ),
        "mov0007" => array(
            "titulo" => "Milagros inesperados",
            "precio" => 3.75,
            "sala" => "Sala 2",
            "hora" => "1:00 pm"
        ),
        "mov0008" => array(
            "titulo" => "La propuesta",
            "precio" => 3.75,
            "sala" => "sala 1",
            "hora" => "9:00 pm"
        ),
        "mov0009" => array(
            "titulo" => "La comunidad del anillo",
            "precio" => 3.75,
            "sala" => "Sala 5",
            "hora" => "7:00 pm"
        ),
        "mov0010" => array(
            "titulo" => "El sexto sentido",
            "precio" => 3.75,
            "sala" => "sala 3",
            "hora" => "8:45 pm"
        ),
        "mov0011" => array(
            "titulo" => "Parasite",
            "precio" => 3.75,
            "sala" => "Sala 1",
            "hora" => "7:00 pm"
        ),
    );

    //Procesando los datos enviados del formulario
    if(isset($_POST['comprar'])){
        $movie = isset($_POST['pelicula']) ? $_POST['pelicula'] : "";
        switch($movie){
            case "mov0000":
                $seleccionada = $peliculas["mov0000"];
                $cantidad = (isset($_POST['cantidad']) && is_numeric($_POST[
'cantidad']))) ? $_POST['cantidad'] : 0;
                break;
            case "mov0001":
                $seleccionada = $peliculas["mov0001"];

```

```
        $cantidad = (isset($_POST['cantidad']) && is_numeric($_POST[
'cantidad'])) ? $_POST['cantidad'] : 0;
        break;
        case "mov0002":
            $seleccionada = $peliculas["mov0002"];
            $cantidad = (isset($_POST['cantidad']) && is_numeric($_POST[
'cantidad'])) ? $_POST['cantidad'] : 0;
            break;
        case "mov0003":
            $seleccionada = $peliculas["mov0003"];
            $cantidad = (isset($_POST['cantidad']) && is_numeric($_POST[
'cantidad'])) ? $_POST['cantidad'] : 0;
            break;
        case "mov0004":
            $seleccionada = $peliculas["mov0004"];
            $cantidad = (isset($_POST['cantidad']) && is_numeric($_POST[
'cantidad'])) ? $_POST['cantidad'] : 0;
            break;
        case "mov0005":
            $seleccionada = $peliculas["mov0005"];
            $cantidad = (isset($_POST['cantidad']) && is_numeric($_POST[
'cantidad'])) ? $_POST['cantidad'] : 0;
            break;
        case "mov0006":
            $seleccionada = $peliculas["mov0006"];
            $cantidad = (isset($_POST['cantidad']) && is_numeric($_POST[
'cantidad'])) ? $_POST['cantidad'] : 0;
            break;
        case "mov0007":
            $seleccionada = $peliculas["mov0007"];
            $cantidad = (isset($_POST['cantidad']) && is_numeric($_POST[
'cantidad'])) ? $_POST['cantidad'] : 0;
            break;
        case "mov0008":
            $seleccionada = $peliculas["mov0008"];
            $cantidad = (isset($_POST['cantidad']) && is_numeric($_POST[
'cantidad'])) ? $_POST['cantidad'] : 0;
            break;
        case "mov0009":
            $seleccionada = $peliculas["mov0009"];
            $cantidad = (isset($_POST['cantidad']) && is_numeric($_POST[
'cantidad'])) ? $_POST['cantidad'] : 0;
            break;
        case "mov0010":
            $seleccionada = $peliculas["mov0010"];
            $cantidad = (isset($_POST['cantidad']) && is_numeric($_POST[
'cantidad'])) ? $_POST['cantidad'] : 0;
            break;
        case "mov0011":
```

```

        $seleccionada = $peliculas["mov0011"];
        $cantidad = (isset($_POST['cantidad']) && is_numeric($_P
OST['cantidad'])) ? $_POST['cantidad'] : 0;
        break;
    }
    //Iniciando la tabla con la información de la película seleccionada
?>
<div class="container">
<header class='navbar navbar-dark bg-dark'>
<a class='navbar-brand' class='d-inline-block align-top' href='#'>
Información del formulario
</a>
</header>

    <table id="hor-
zebra" class="table" summary="Datos recibidos del formulario">

        <thead>
            <tr class='bg-secondary'>
                <th scope="col" colspan="2">Boleto de entrada</th>
            </tr>
        </thead>
        <tbody>
<?php
    //Obteniendo los datos de la película seleccionada que está almacenada e
n $seleccionada
    $table = "";
    $i = 1;
    foreach($seleccionada as $indice => $dato){
        if($i%2 != 0){
            $table .= "\t<tr class=\"odd\">\n";
        }
        else{
            $table .= "\t<tr class=\"even\">\n";
        }
        if($indice == "precio"){
            $dato *= $cantidad;
            $dato = "\$" . number_format((double)$dato, 2, ".", ",");
        }
        $table .= "\t\t<td>$indice</td>\n";
        $table .= "\t\t<td> $dato</td>\n\n\t</tr>\n";
        $i++;
    }
    echo $table;
}
?>

        </tbody>
    </table>
    <a href="ejemplo4.php">Ingresar otra película</a>

```

```
</div>
</body>
</html>
```

Vista en pantalla:

Cartelera de Cine Bosco

Entrada al cine


Película

Avengers

Cantidad

4

Comprar



Información del formulario

Boleto de entrada

título	Avengers
precio	\$15.00
sala	Sala 1
hora	7:00 pm

[Ingresar otra película](#)

## V. DISCUSIÓN DE RESULTADOS

1. Cree un script que le permita ingresar un número del 1 al 30 a través de un formulario, que solamente deberá contener un campo de texto, su etiqueta y un botón de envío. El script PHP que realizará deberá mostrar la tabla de multiplicar de ese número de forma ordenada y utilizando hojas de estilo para una buena apariencia visual.
2. Realice un script PHP que dadas tres notas de un listado de estudiantes organizadas en una matriz asociativa, le permita obtener el promedio de ellas sabiendo que las notas son de una tarea, una investigación y el examen parcial con la ponderación de 50%, 30% y 20% respectivamente. Por último, muestre las notas promediadas de cada alumno en una tabla

bien presentable, mostrando el nombre del alumno, la nota que tiene en cada actividad y en una celda combinada su promedio.

#### **VI. INVESTIGACIÓN COMPLEMENTARIA**

1. Investigue la utilización de las funciones `include_once()` y `require_once()` e indique la diferencia o diferencias que tienen con respecto a sus homólogas `include()` y `require()`.
2. Investigue sobre el uso de funciones anónimas en PHP, indicando los motivos por los cuáles pueden ser utilizadas e implemente un ejemplo simple de cada caso.

#### **VII. BIBLIOGRAFIA**

- Gutiérrez, Abraham / Bravo, Ginés. PHP 5 a través de ejemplos. Editorial Alfaomega RAMA. 1ra edición. México. Junio 2005.
- Gil Rubio, Francisco Javier/Villaverde, Santiago Alonso/Tejedor Cerbel, Jorge A. Creación de sitios web con PHP 5. Editorial McGraw-Hill. 1ra edición. Madrid, España, 2006.
- John Coggeshall. La Biblia de PHP 5. 1ra Edición. Editorial Anaya Multimedia. Madrid España.
- <http://www.php.net/manual/en>