



Benemérita Universidad Autónoma
de Puebla

Práctica #3. Manejo de formularios

Aplicaciones web

Facultad de Ciencias de la Computación

Equipo:

Vargas Arenas Pedro - 201734553



Descripción

En este documento se describe el desarrollo de la tercera práctica de la materia que consiste en elaborar un programa que realice las siguientes operaciones con matrices: Suma, resta, multiplicación y transpuesta. Además, se creará un formulario para recibir la cantidad de filas y columnas de cada matriz para verificar cuáles operaciones se pueden ejecutar. Las matrices se llenarán con números aleatorios dentro del programa.

Desarrollo

Formulario

El formulario se desarrolló en HTML, éste cuenta con cuatro entradas donde se ingresa el número de filas y columnas de las dos matrices. Todas las entradas son de tipo texto. Además, se tiene un botón para enviar los datos que se hayan ingresado. Una vez que se presiona el botón, se llama al archivo *OperacionMatriz.php* para realizar las validaciones correspondientes y así ejecutar las operaciones de matrices.

```
<form action="OperacionMatriz.php" method="POST">
  <label>Número de filas de la primera matriz </label>
  <input id="filas" type="text" name="fila1"/>
  <br>
  <label>Número de columnas de la primera matriz</label>
  <input id="columnas" type="text" name="columna1"/>
  <br>
  <label>Número de filas de la segunda matriz </label>
  <input id="filas" type="text" name="fila2"/>
  <br>
  <label>Número de columnas de la segunda matriz</label>
  <input id="columnas" type="text" name="columna2"/>
  <br>
  <input type="submit" value=" Enviar">
```

Operaciones de matrices

El desarrollo de las operaciones solicitadas con matrices se realizó con el lenguaje de programación PHP. La cantidad de filas y columnas de cada matriz se adquieren del formulario mediante la propiedad `$_REQUEST['nombre_input']`, estas se almacenan en las variables `$f1`, `$c1`, `$f2` y `$c2`.

```
$f1=$_REQUEST['fila1'];
$c1=$_REQUEST['columna1'];
$f2=$_REQUEST['fila2'];
$c2=$_REQUEST['columna2'];
```

Para llenar las matrices con datos aleatorios se creó la función *llenarMatriz(\$fila, \$columna)* que recibe la cantidad de filas y columnas, se tienen dos ciclos for que van de cero hasta el número de filas y columnas respectivamente, dentro de ellos se recorre la matriz y en cada posición se asigna un número aleatorio con ayuda de la función *rand(0, 30)*.

```
function llenarMatriz($fila, $columna){  
    for($i = 0; $i < $fila; $i++)  
        for($j = 0; $j < $columna; $j++)  
            $res[$i][$j] = rand(0, 30);  
    return $res;  
}
```

La suma se realiza con la función *suma(\$matriz1, \$matriz2)* que recibe las dos matrices a operar. La función consiste en dos ciclos for anidados que recorren ambas matrices para obtener cada elemento y sumarlos, el resultado se almacena en la posición correspondiente dentro de una nueva matriz.

```
function suma($matriz1, $matriz2){  
    $fila = count($matriz1);  
    $columna = count($matriz1[0]);  
    echo "<h4>Suma</h4>";  
    for($i = 0; $i < $fila; $i++)  
        for($j = 0; $j < $columna; $j++)  
            $res[$i][$j] = $matriz1[$i][$j] + $matriz2[$i][$j];  
    mostrar($res);  
}
```

La resta tiene un procedimiento similar que en la suma, se realiza con la función *resta(\$matriz1, \$matriz2)* que recibe las dos matrices a operar. La función consiste en dos ciclos for anidados que recorren ambas matrices para obtener cada elemento y restarlos, el resultado se almacena en la posición correspondiente dentro de una nueva matriz.

```
function resta($matriz1, $matriz2){  
    $fila = count($matriz1);  
    $columna = count($matriz1[0]);  
    echo "<h4>Resta</h4>";  
    for($i = 0; $i < $fila; $i++)  
        for($j = 0; $j < $columna; $j++)  
            $res[$i][$j] = $matriz1[$i][$j] - $matriz2[$i][$j];  
    mostrar($res);  
}
```

La multiplicación se ejecuta mediante la función *multiplicacion(\$matriz1, \$matriz2)* que recibe dos matrices. El proceso se realiza con ayuda de tres for, para delimitarlos se utiliza el número de filas de la primera matriz, el número de columnas de la segunda y primera matriz respectivamente. En cada iteración del segundo ciclo, se asigna un cero a la respectiva posición de la nueva matriz, en el último ciclo se multiplican los elementos de cada fila de la primera matriz por los elementos de cada columna de la segunda matriz, el resultado de cada multiplicación se suma con el acumulado que exista en la posición de la matriz nueva. Esto se repite por la cantidad de columnas de la primera matriz para pasar a la siguiente posición de la nueva matriz.

```
function multiplicacion($matriz1, $matriz2){

    $fila1 = count($matriz1);
    $columna2 = count($matriz2[0]);
    $columna1 = count($matriz1[0]);
    echo "<h4>Multiplicación</h4>";
    for($i = 0; $i < $fila1; $i++){
        for($j = 0; $j < $columna2; $j++){
            $res[$i][$j] = 0;
            for($k = 0; $k < $columna1; $k++){
                $res[$i][$j] += $matriz1[$i][$k] * $matriz2[$k][$j];
            }
        }
    }
    mostrar($res);
}
```

La transpuesta de la matriz se logra con ayuda de la función *transpuesta(\$matriz)*, consiste en dos ciclos for anidados que van de cero hasta la cantidad de columnas y filas de la matriz, dentro de los ciclos se obtienen los elementos de cada fila de la matriz y se guardan en las columnas de una nueva matriz.

```
function transpuesta($matriz){

    $fila = count($matriz);
    $columna = count($matriz[0]);
    echo "<h4>Transpuesta</h4>";
    for($i = 0; $i < $columna; $i++){
        for($j = 0; $j < $fila; $j++){
            $res[$i][$j] = $matriz[$j][$i];
        }
    }
    mostrar($res);
}
```

Al final de cada operación se llama a la función *mostrar(\$matriz)* que recibe la matriz resultante de la operación.

```
function mostrar($matriz){

    $fila = count($matriz);
    $columna = count($matriz[0]);

    for($i = 0; $i < $fila; $i++){
        for($j = 0; $j < $columna; $j++){
            echo " ".$matriz[$i][$j]. " ";
        }
        echo "<br>";
    }
}
```

Para validar las operaciones que se pueden realizar con las columnas y filas obtenidas, se utiliza la función `verificarMatriz($filaM1, $columnaM1, $filaM2, $columnaM2, $m1, $m2)`. Para validar la suma y resta, la cantidad de las columnas y de las filas de ambas matrices deben ser iguales, es decir, deben tener la misma dimensión. La validación de la multiplicación solo verifica que el número de columnas de la primera matriz y de filas de la segunda matriz sean iguales. Finalmente, la validación para la operación de transposición se basa en que la matriz tenga la misma cantidad de filas y columnas, es decir, que sea cuadrada.

```
function verificarMatriz($filaM1, $columnaM1, $filaM2, $columnaM2, $m1, $m2){
    if($filaM1 == $filaM2 && $columnaM1 == $columnaM2){
        suma($m1, $m2);
        resta($m1, $m2);
    } if($columnaM1 == $filaM2)
        multiplicacion($m1, $m2);
    if($filaM1 == $columnaM1)
        transpuesta($m1);
    if($filaM2 == $columnaM2)
        transpuesta($m2);
}
```

Ejecución

Primeramente, se debe activar WampServer para activar un servidor y así lograr ejecutar PHP en el navegador. Posteriormente, se ejecuta el archivo `index.html`, ya que aquí se ingresarán los datos correspondientes a las filas y columnas de las dos matrices.

Número de filas de la primera matriz

Número de columnas de la primera matriz

Número de filas de la segunda matriz

Número de columnas de la segunda matriz

Primero se ingresará dos filas y dos columnas para ambas matrices, es decir, serán cuadradas. Con esto, se realizarán las cuatro operaciones.

Matriz 1	Suma	Transpuesta	
11 22 16 1	20 52 35 24	11 16 22 1	
Matriz 2	Resta	Transpuesta	Multiplicación
9 30 19 23	2 -8 -3 -22	9 19 30 23	517 836 163 503

Matrices generadas

Suma y resta de matrices

Transpuesta de ambas matrices

Multiplicación de matrices

Ahora se ingresa la primera matriz con dimensión 2x3 y la segunda matriz es de dimensión 3x2. Con esto solo se realizará la multiplicación.

Matriz 1

13 26 24
4 24 6

Transpuesta

13 4
26 24
24 6

Matriz 2

17 10
5 13
8 26

Multiplicación

543 1092
236 508

Transpuesta

17 5 8
10 13 26

Matrices generadas

Multiplicación de
matrices

Transpuesta de
ambas matrices

Por último, se ingresa la primera matriz con dimensión 2x2 y la segunda matriz tiene dimensión 3x3. Así, solo se genera la transpuesta de ambas matrices.

Matriz 1

27 4
10 15

Transpuesta

27 10
4 15

Matriz 2

20 5 0
13 26 1
29 9 29

Transpuesta

20 13 29
5 26 9
0 1 29

Matrices generadas

Transpuesta de
ambas matrices

Referencias

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

https://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=569:matrices-array-arreglo-multidimensional-php-arrays-anidados-concepto-ejemplos-y-ejercicios-cu00824b&catid=70&Itemid=193

<https://www.mclibre.org/consultar/php/lecciones/php-matrices.html#:~:text=Qu%C3%A9%20es%20una%20matriz,-Una%20matriz%20es&text=En%20PHP%2C%20una%20matriz%20es,de%20tipos%20de%20datos%20diferentes.>

https://developer.mozilla.org/es/docs/Learn/Forms/Your_first_form