



Benemérita Universidad
Autónoma de Puebla



Facultad de Ciencias de la
Computación

Operaciones con matrices

Actividad 3

Materia:

Aplicaciones web

Integrantes del equipo:

Rosa Suarez Calderon

Ana Laura Torres Marin

Alfredo de Jesus Ramos Benavides

Profesor:

Pedro Bello López

Periodo:

Primavera 2022

Fecha de entrega:

24 de enero de 2022

Desarrollo del trabajo

Antes de comenzar a desarrollar el trabajo, primero se hizo una investigación sobre cómo es que se realizan las operaciones de las matrices y además en qué condiciones se pueden realizar estas operaciones. A continuación, se muestra un pequeño resumen sobre las matrices.

¿Qué es una matriz?

Una matriz es un arreglo dimensional de números. Una matriz se representa por medio de una letra mayúscula (A, B, ...) y sus elementos con la misma letra en minúscula (a, b, ...), con un doble subíndice donde el primero indica la fila y el segundo la columna a la que pertenece [1].

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

Figura 1. Ejemplo de una matriz de n*n.

Operaciones con matrices

Suma y Resta de Matrices

La suma y/o resta de matrices es una operación donde se suman o restan los elementos de dos o más matrices que coincidan con su posición [1].

La suma/resta de matrices consiste en operar el número de una matriz con el número que tiene la misma posición en la otra matriz [1].

$$A = \begin{pmatrix} 2 & 0 & 1 \\ 3 & 0 & 0 \\ 5 & 1 & 1 \end{pmatrix} \quad B = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$
$$A + B = \begin{pmatrix} 2+1 & 0+0 & 1+1 \\ 3+1 & 0+2 & 0+1 \\ 5+1 & 1+1 & 1+0 \end{pmatrix} = \begin{pmatrix} 3 & 0 & 2 \\ 4 & 2 & 1 \\ 6 & 2 & 1 \end{pmatrix}$$
$$A - B = \begin{pmatrix} 2-1 & 0-0 & 1-1 \\ 3-1 & 0-2 & 0-1 \\ 5-1 & 1-1 & 1-0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 2 & -2 & -1 \\ 4 & 0 & 1 \end{pmatrix}$$

Figura 2. Ejemplo de la suma/resta de dos matrices.

Producto de matrices

El resultado de multiplicar dos matrices es otra matriz en la que el elemento que ocupa el lugar C_{ij} se obtiene sumando los productos parciales que se obtienen al multiplicar todos los elementos de la fila “i” de la primera matriz por los elementos de la columna “j” de la segunda matriz. Es decir, multiplicamos la primera fila por los elementos de la primera columna y el resultado será nuestro nuevo elemento. Para ello, el número de columnas de la primera matriz debe coincidir con el de filas de la segunda. Si no fuese así no podríamos realizar la operación [2] .

Ejemplo:

$$\begin{pmatrix} 1 & 3 \\ -1 & 2 \end{pmatrix} \cdot \begin{pmatrix} 1 & 3 & 2 \\ 2 & -3 & -1 \end{pmatrix} = \begin{pmatrix} 1.1 + 3.2 & 1.3 + 3.(-3) & 1.2 + 3.(-1) \\ -1.1 + 2.2 & -1.3 + 2.(-3) & -1.2 + 2.(-1) \end{pmatrix} \\ = \begin{pmatrix} 7 & -6 & -1 \\ +3 & -9 & -4 \end{pmatrix}$$

Figura 3. Producto de matrices.

Observamos como la matriz resultante tiene el número de filas de la primera y el de columnas de la segunda.

Debemos recordar, que las matrices no tienen la propiedad conmutativa. En el caso de que se pudiera operar $A.B$ y $B.A$ el resultado por lo general puede ser diferente.

Transposición de matrices

Sea A una matriz de tamaño $m \times n$ decimos que la transposición de la matriz A es una nueva matriz de tamaño $n \times m$ donde los elementos de la matriz A que están en la posición ij pasan a estar en la posición ji , a esta nueva matriz se le llama A *traspuesta* (o traspuesta)[3].

$$\begin{pmatrix} a_{11} & \cdots & a_{1n} \\ a_{21} & \cdots & a_{2n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix}^T = \begin{pmatrix} a_{11} & \cdots & a_{n1} \\ a_{12} & \cdots & a_{n2} \\ \vdots & \ddots & \vdots \\ a_{1m} & \cdots & a_{nm} \end{pmatrix}$$

Figura 4. Transpuesta de una matriz.

Por lo anterior, podemos decir que la operación suma y resta deben de ser del mismo orden. A diferencia de la operación multiplicación y transpuesta, las matrices pueden o no ser del mismo tamaño, sin embargo, para la multiplicación se debe de cumplir que **“el número de columnas de la primera matriz debe coincidir con el número de filas de la segunda matriz”**.

Una vez entendida la parte teórica, procedimos a desarrollar el programa. A continuación, se explicará el código que se desarrolló, así como capturas del funcionamiento del programa.

Elaboración de form

El elemento HTML `<form>` se utiliza para crear un formulario HTML para la entrada de datos por parte del usuario y es un contenedor para diferentes tipos de elementos de entrada como: campos de texto, casillas de verificación, botones de opción, botones de envío, etc.

Para este trabajo solo necesitamos los campos de texto y el botón de envío.

Para los campos de texto se necesitó la etiqueta `<input type="text" name="fname">` y para el botón de envío se usó la etiqueta `<input type="submit">`. A continuación, se muestra el bloque de código, así como el resultado:

```
<div class="container">
  <div class="title">Operaciones de matrices</div>
  <form action="matrices.php" method="post">
    <div class="datos-user">
      <div class="input-box">
        <span class="datos"><br><h3>Datos para la matriz 1 </h3></span>
      </div>
      <div class="input-box">
        <span class="datos"></span>
      </div>
      <div class="input-box">
        <span class="datos">Numero de filas</span>
        <input type="text" name="filasm1" placeholder="Numero de filas" required>
      </div>
      <div class="input-box">
        <span class="datos">Numero de columnas</span>
        <input type="text" name="columnasm1" placeholder="Numero de columnas" required>
      </div>
      <div class="input-box">
        <span class="datos"><br><h3>Datos para la matriz 2</h3></span>
      </div>
      <div class="input-box">
        <span class="datos"></span>
      </div>
      <div class="input-box">
        <span class="datos">Numero de filas</span>
        <input type="text" name="filasm2" placeholder="Numero de filas" required>
      </div>
      <div class="input-box">
        <span class="datos">Numero de columnas</span>
        <input type="text" name="columnasm2" placeholder="Numero de columnas" required>
      </div>
    </div>
    <div class="button" align = center>
      <INPUT TYPE="submit" VALUE=" Calcular">
    </div>
  </form>
</div>
```

Operaciones de matrices

Datos para la matriz 1

Numero de filas

Numero de columnas

Datos para la matriz 2

Numero de filas

Numero de columnas

Figura 5. Form para introducir los datos.

En el bloque de código que se muestra en la figura 5, hay dos recuadros color gris, esas líneas de código nos ayudan a tener una mejor organización, este cambio solo se ve cuando le agregamos estilo a la página.

Para los campos de texto esta la palabra reservada *required*, esta palabra nos ayuda a obligar al usuario a ingresar al menos un dato en los campos de texto, de lo contrario, no se realizará la acción de “Calcular”.

El botón “Calcular” tiene la función de que, cuando se oprime, manda los datos a la página que los requiera, para que esta página pueda utilizar sus datos y realizar las operaciones que se le indiquen.

Operaciones de matrices.

Una vez que se obtiene el form que recibe los datos del usuario, podemos empezar a crear las matrices para poder realizar las operaciones. Antes de poder realizar las operaciones, necesitamos obtener los datos del form, por ello hacemos uso de la variable de php `$_REQUEST`, la cual nos ayuda a recopilar los datos.

```
$filam1 = $_REQUEST['filasm1'];  
$columnam1 = $_REQUEST['columnasm1'];  
$filam2 = $_REQUEST['filasm2'];  
$columnam2 = $_REQUEST['columnasm2'];
```

Figura 6. Uso de la variable `$_REQUEST`.

Como se observa en la figura 6, obtenemos los datos del Form, y cada variable se le asigna una nueva variable para que pueda usarse en el programa. Sino usamos la variable `$_REQUEST` no podríamos obtener los datos del Form, y por ende no se podría hacer ninguna operación.

A continuación, se muestran las funciones de las operaciones que se realizaron para mostrar las operaciones solicitadas.

Antes de realizar las operaciones de las matrices, es necesario que las matrices tengan valores, pero sería tardado si el usuario ingresa dato por dato, es por ello que se realizó una función para llenar las matrices con datos enteros aleatorios.

CÓDIGO DEL LLENADO DE UNA MATRIZ

En las siguiente Captura de pantalla les muestro como hacer un llenado de matrices con valores aleatorios en PHP.

- La [función llenarMatriz \(\)](#) genera un array multidimensional, por defecto de $n*n$, con números del 1 al 100, sin repeticiones.

```
//Funcion llenarMatriz
function llenarMatriz($f,$c){ //Se crea la funcion llenarMatriz y pasamos como paso de parametros las filas y columnas
    $numeros = array(); //Se inicializa la matriz de resultados
    for ($i=0; $i <$f; $i++) { //Se crea el ciclo for el cual hara el recorrido para las filas
        for ($j=0; $j < $c; $j++) { //Se crea el ciclo for el cual hara el recorrido para las columnas
            $numeros[$i][$j] = rand(1,100); //Se llena la matriz con los numeros aleatorios
        }
    }

    return $numeros; //Devuelve el resultado
}
```

Figura 7. Función llenarMatriz().

El código fuente ya contiene comentarios en cada línea para que se pueda entender mejor.

CODIGO QUE IMPRIME LA MATRIZ

Una vez que tenemos los datos de las matrices, necesitamos observar los datos, es por ello que se escribió una función para que realizara la impresión de las matrices, a continuación, se muestra esta función:

```
function imprimirMatriz($m,$f,$c){

    echo "<table border align = 'center'>";
    for($i=0;$i<$f;$i++){
        echo "<tr>";
        for ($j=0; $j < $c; $j++) {
            echo "<td>";
            echo $m[$i][$j];
            echo "</td>";
        }
        echo "</tr>";
    }
    echo "</table>";
}
```

Figura 8. Función que imprime la matriz.

Como se observa en la figura 8, la función imprimirMatriz esta implementada con HTML. Para organizar los valores de la matriz en una tabla usamos las etiquetas de la tabla. Con la etiqueta `<tr>` los valores se van posicionando por las filas y, con la etiqueta `<td>`, los valores se van posicionando por las columnas.

CÓDIGO DE LA SUMA DE MATRICES

En el siguiente código les muestro como hacer la suma de matrices con PHP.

- La [funcion sumaMatriz \(\)](#) la usamos para calcular la suma de los valores de los elementos de dos matrices.

En la siguiente captura de pantalla se muestra cómo hacer la suma de dos matrices mediante la [funcion sumaMatriz \(\)](#).

```
function sumaMatriz($matriz1,$matriz2){  
    $suma = array(); // Iniciamos la matriz de resultados  
  
    for($i=0; $i<count($matriz1); $i++){ // Recorremos cada fila  
        $suma[] = array(); // Creamos una entrada por cada fila  
        for($j=0; $j<count($matriz1[$i]); $j++){ // Recorremos las columnass  
            $suma[$i][$j] = $matriz1[$i][$j] + $matriz2[$i][$j]; // Realizamos la suma o resta, cambiar simbolo +/-  
        }  
    }  
    return $suma;  
}
```

Figura 9. Función sumaMatriz().

El código fuente ya contiene comentarios en cada línea para que se pueda entender mejor.

CÓDIGO DE LA RESTA DE MATRICES

En el siguiente código les muestro como hacer la resta de matrices con PHP.

- La [funcion restaMatriz \(\)](#) la usamos para calcular la resta de los valores de los elementos de dos matrices.

En la siguiente captura de pantalla se muestra cómo hacer la resta de dos matrices mediante la [funcion restaMatriz \(\)](#).

```
function restaMatriz($matriz1,$matriz2){  
    $resta = array(); // Iniciamos la matriz de resultados  
  
    for($i=0; $i<count($matriz1); $i++){ // Recorremos cada fila  
        $resta[] = array(); // Creamos una entrada por cada fila  
        for($j=0; $j<count($matriz1[$i]); $j++){ // Recorremos las columnass  
            $resta[$i][$j] = $matriz1[$i][$j] - $matriz2[$i][$j]; // Realizamos la suma o resta, cambiar simbolo +/-  
        }  
    }  
    return $resta;  
}
```

Figura 10. Funcion restaMatriz().

El código fuente ya contiene comentarios en cada línea para que se pueda entender mejor.

CÓDIGO DE LA MULTIPLICACION DE MATRICES

Función para la operación de Multiplicación de dos matrices, como principio tenemos como parámetros a la matriz 1 y 2. Así mismo muestra una condición dicha condición verifica si las matrices (1,2) son diferentes, en caso de que sean distintas muere el proceso, caso contrario realiza las iteraciones y retorna la nueva matriz (3).

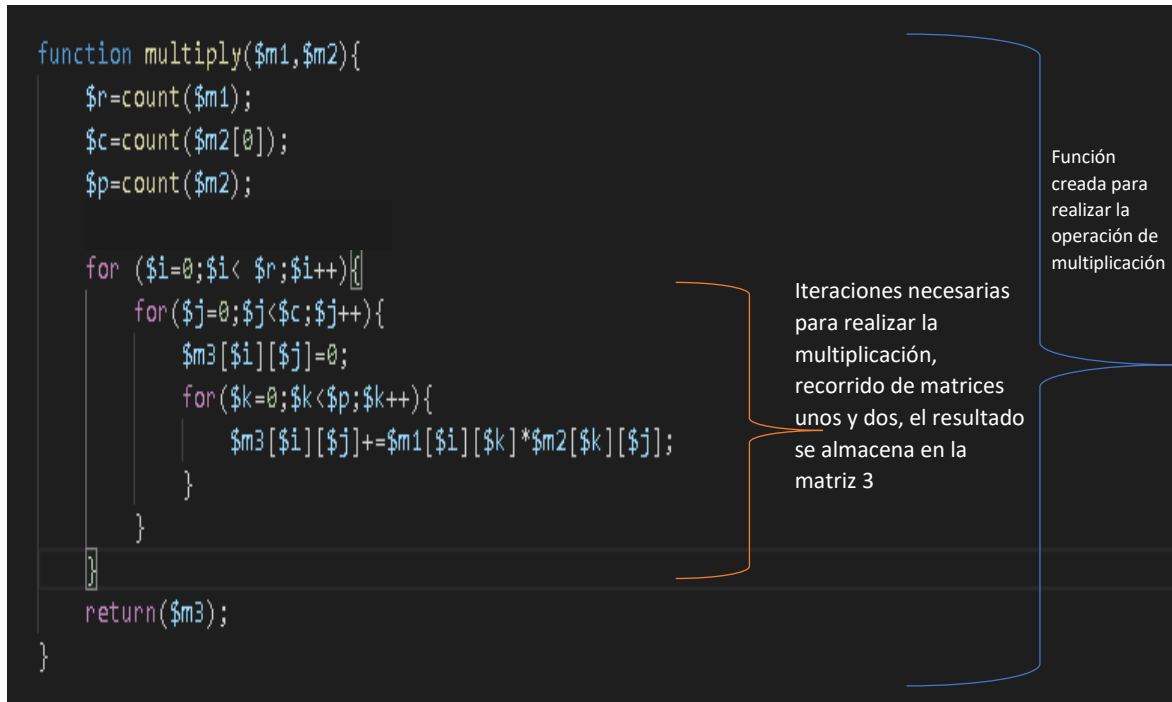


Figura 11. Función que multiplica dos matrices.

CÓDIGO DE LA TRANSPUESTA DE MATRIZ.

Para realizar esta función utilizamos Foreach la cual proporciona una manera fácil de iterar sobre matrices. Foreach funciona solo en matrices y objetos, y generará un error cuando intente usarlo en una variable con un tipo de datos diferente o una variable no inicializada. Hay dos sintaxis:

foreach (expresión_iterable como \$valor)

declaración

foreach (expresión_iterable como \$clave => \$valor)

declaración

```
function transpose($array) {
    $transposed_array = array();
    foreach ($array as $row_key => $row) {
        foreach ($row as $column_key => $element) {
            $transposed_array[$column_key][$row_key] = $element;
        }
    }
    return $transposed_array;
}
```

Figura 12. Función transpuesta de matrices.

Una vez que tenemos las funciones solo nos resta hacer los llamados de las funciones para que se puedan hacer las operaciones correspondientes. A continuación, se muestra el llamado de las funciones:

```
if($tam1 == $tam2){
    if($columnam1 == $columnam2){
        echo "<h2>Suma de matrices</h2>";
        $matsuma = sumaMatriz($mat1,$mat2);
        imprimirMatriz($matsuma,$filam1,$columnam1);

        echo "<h2>Resta de matrices</h2>";
        $matresta = restaMatriz($mat1,$mat2);
        imprimirMatriz($matresta,$filam1,$columnam1);
    }else{
        echo'<h3> Las matrices no tienen el mismo número de filas y columnas, por lo tanto, no se pueden realizar las operaciones Suma y Resta</h3>';
    }
}else{
    echo'<h3> Las matrices no son del mismo tamaño, por lo tanto, no se puede realizar las operaciones Suma y Resta</h3>';
}
}
```

Figura 13. Llama de funciones llamarMatriz, imprimirMatriz, sumaMatriz y restaMatriz.

```
if($columnam1 == $filam2){
    echo "<h2>Multiplicación de matrices </h2>";
    $matmul = multiply($mat1,$mat2);
    imprimirMatriz($matmul,$filam1,$columnam2);
}else{
    echo "<h2>Multiplicación de matrices </h2>";
    echo'<h3> <h3>El número de columnas de la primera matriz debe coincidir con el número de filas de la segunda matriz</h3>';
}
}
```

Figura 14. Llamado a la función multiply y imprimirMatriz.

```
echo "<h2>Transpuesta de la matriz 1 <h2>";
$matrans = transpose($mat1);
imprimirMatriz($matrans,$columnam1,$filam1);

echo "<h2>Transpuesta de la matriz 2 <h2>";
$matrans = transpose($mat2);
imprimirMatriz($matrans,$columnam2,$filam2);
```

Figura 15. Llamado a la función transpose y imprimirMatriz.

En la figura 13, observamos que hay dos condiciones que explicaremos. La primera condición nos da como condición de que ambas matrices deben de ser del mismo tamaño para que se puedan efectuar las operaciones suma y resta. La segunda condición nos da como condición que al menos las columnas deben de tener el mismo valor, de lo contrario tampoco se podrán mostrar las operaciones suma y resta. Por ejemplo: Si tenemos la matriz 1 de 3x6 y la matriz 2 de 6x3, cumplen que sean del mismo tamaño, pero no cumplen la condición de que tengan el mismo numero de columnas, se puede poner también las filas, pero con el echo de que no tengan el mismo numero de columnas, hace que no se puedan efectuar las operaciones.

La figura 14 muestra las condiciones que debe de cumplirse para que pueda efectuarse la operación de multiplicación de matrices, estos ya se han descrito al inicio del documento.

La figura 15 solo muestra el llamado de las funciones para mostrar la transpuesta de la matriz.

Ahora que ya tenemos los llamados, podemos poner un ejemplo de la salida del programa, a continuación, se muestran las capturas de pantalla:

Operaciones de matrices

Datos para la matriz 1

Numero de filas
Numero de columnas

Datos para la matriz 2

Numero de filas
Numero de columnas

Calcular

Resultado de las operaciones con Matrices

Tamaño de la matriz 1 es de [5][5] Tamaño de la matriz 2 es de [5][5]

Matriz 1

65	21	52	69	89
50	58	58	47	93
4	62	10	23	79
97	66	79	29	2
68	80	42	15	6

Matriz 2

2	65	62	63	91
55	78	39	6	62
7	27	61	69	98
41	67	65	100	45
78	25	47	54	95

Suma de matrices

67	86	114	132	180
105	136	97	53	155
11	89	71	92	177
138	133	144	129	47
146	105	89	69	101

Figura 16. Ejecución del programa.

Como observamos en la figura 16, vemos que se generan las dos matrices y muestra la operación suma de matrices, no se ponen capturas de pantalla de lo demás para no saturar el documento de muchas imágenes, pero al ver que las matrices son del mismo orden se puede deducir que las demás operaciones se han realizado.

Estilo de las paginas

Para que nuestras paginas no se vieran tan simples usamos la hoja de estilo CSS, esto nos permitió implementar diferentes cosas para obtener una mejor visualización de las páginas, no se explicaran a detalle cada variable, solo mostraremos un antes y un después de que se le aplicara estilo. A continuación, se muestran estos cambios con su respectivo bloque de código:

Sin estilo

Operaciones de matrices

Datos para la matriz 1

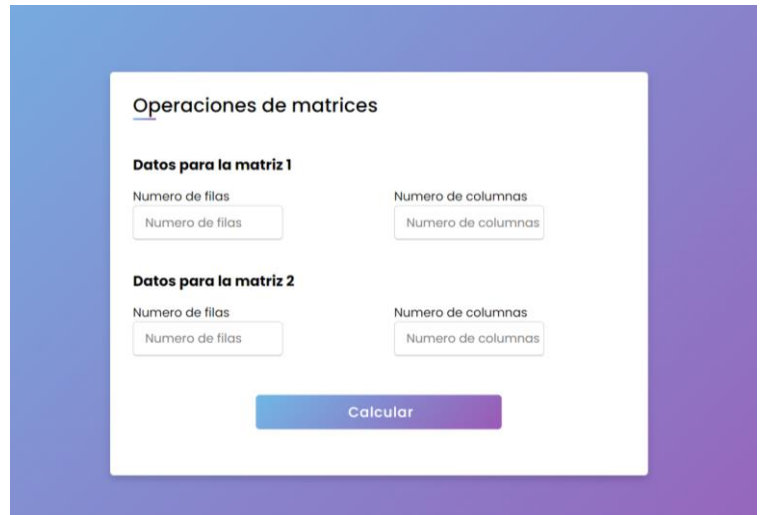
Numero de filas
Numero de columnas

Datos para la matriz 2

Numero de filas
Numero de columnas

Calcular

Con estilo



Operaciones de matrices

Datos para la matriz 1

Numero de filas Numero de columnas

Datos para la matriz 2

Numero de filas Numero de columnas

Calcular

Figura 17. Aplicación de estilo al Form.

```
@import url('https://fonts.googleapis.com/css2?family=Poppins:wght@200;300;400;500;600;700&display=swap');
*{
  margin: 0;
  padding: 0;
  box-sizing: border-box;
  font-family: 'Poppins',sans-serif;
}
body{
  height: 100vh;
  display: flex;
  justify-content: center;
  align-items: center;
  padding: 10px;
  background: linear-gradient(135deg, #71b7e6, #9b59b6);
}
.container{
  max-width: 700px;
  width: 100%;
  background-color: #fff;
  padding: 25px 30px;
  border-radius: 5px;
  box-shadow: 0 5px 10px rgba(0,0,0,0.15);
}
.container .title{
  font-size: 25px;
  font-weight: 500;
  position: relative;
}
.container .title:before{
  content: "";
  position: absolute;
  left: 0;
  bottom: 0;
  height: 3px;
  width: 30px;
  border-radius: 5px;
  background: linear-gradient(135deg, #71b7e6, #9b59b6);
}
```

```

.container form .datos-user{
  display: flex;
  flex-wrap: wrap;
  justify-content: space-between;
  margin: 20px 0 12px 0;
}
form .datos-user .input-box{
  margin-bottom: 15px;
  width: calc(100% / 2 - 20px);
}
form .input-box span.details{
  display: block;
  font-weight: 500;
  margin-bottom: 5px;
}
.datos-user .input-box input{
  height: 45px;
  width: 65%;
  outline: none;
  font-size: 16px;
  border-radius: 5px;
  padding-left: 15px;
  border: 1px solid #ccc;
  border-bottom-width: 2px;
  transition: all 0.3s ease;
}
.datos-user .input-box input:focus,
.datos-user .input-box input:valid{
  border-color: #9b59b6;
}

form .button{
  height: 45px;
  margin: 35px 0
}
form .button input{
  height: 100%;
  width: 50%;
  border-radius: 5px;
  border: none;
  color: #fff;
  font-size: 18px;
  font-weight: 500;
  letter-spacing: 1px;
  cursor: pointer;
  transition: all 0.3s ease;
  background: linear-gradient(135deg, #71b7e6, #9b59b6);
}

```

Figura 18. Código para el estilo del Form.

De manera general, el código nos permitió tener mejor organizado las cajas de texto, nos permitió encerrarlo en un recuadro y centrarlo en medio de la página. También nos permitió agregarle un estilo diferente de letra, entre otras cosas.

Sin estilo

Resultado de las operaciones con Matrices

Tamaño de la matriz 1 es de [5][5] Tamaño de la matriz 2 es de [5][5]

Matriz 1

65	21	52	69	89
50	58	58	47	93
4	62	10	23	79
97	66	79	29	2
68	80	42	15	6

Matriz 2

2	65	62	63	91
55	78	39	6	62
7	27	61	69	98
41	67	65	100	45
78	25	47	54	95

Suma de matrices

67	86	114	132	180
105	136	97	53	155
11	89	71	92	177
138	133	144	129	47
146	105	89	69	101

Con estilo

Resultado de las operaciones con Matrices

Tamaño de la matriz 1 es de [5][5] Tamaño de la matriz 2 es de [5][5]

Matriz 1

45	63	69	59	22
25	17	42	77	48
55	56	31	51	58
34	12	52	36	2
13	38	32	34	34

Matriz 2

49	30	98	48	56
31	93	46	11	72
59	33	41	71	20
48	19	93	97	96
5	88	30	18	30

Figura 19. Aplicación de estilo a la página de las operaciones con matrices

```
@import url('https://fonts.googleapis.com/css2?family=Poppins:wght@200;300;400;500;600;700&display=swap');
*{
  margin: 20px;
  padding: 5px;
  box-sizing: border-box;
  font-family: 'Poppins', sans-serif;
}
body{
  display: table-row-group;
  justify-content: left;
  align-items: center;
  padding: 10px;
  background-image: linear-gradient(to bottom, #50d1d4, #59d4d7, #61d7da, #69dadd, #70dde0);
}
table{
  border: 1px solid black;
  color: black;
}
```

Figura 20. Código para el estilo de la página que muestra las operaciones.

De manera general, la figura 18 nos muestra que, con la aplicación de estilo, se pudo obtener otro estilo de letra, la matriz se muestra un poco más grande y mejor distribuido. Además de que tiene un color y también se ve que hay más espacio.

Referencias

- [1] A. Ramos, «EVILNAPSIS,» 18 Noviembre 2019. [En línea]. Available:
<https://evilnapsis.com/2019/11/18/suma-y-resta-de-matrices-con-php/>. [Último acceso: 19 Enero 2022].
- [2] yosoytuprofe, «YO SOY TU PROFE 20 MINUTOS,» 4 Junio 2017. [En línea]. Available:
<https://yosoytuprofe.20minutos.es/2017/06/04/operaciones-con-matrices/>. [Último acceso: 18 Enero 2022].
- [3] A. ARIAS, «Totumat,» 14 Junio 2020. [En línea]. Available:
<https://totumat.com/2020/06/14/operaciones-entre-matrices/>. [Último acceso: 19 Enero 2022].
- [4] D. J, «Stackoverflow,» 19 Marzo 2021. [En línea]. Available:
<https://es.stackoverflow.com/posts/244459/revisions>. [Último acceso: 19 Enero 2022].

WebSites

https://www.w3schools.com/html/html_forms.asp

<https://mosaic.uoc.edu/ac/le/es/m6/ud7/index.html>

<https://mycolor.space/gradient?ori=to+bottom&hex=%2350D1D4&hex2=%2370DDE0&sub=1>