



**UNIVERSIDAD TECNOLÓGICA DE PANAMÁ**  
**FACULTAD DE INGENIERÍA EN SISTEMAS COMPUTACIONALES**  
**DEPARTAMENTO DE ARQUITECTURA DE COMPUTADORAS Y REDES**  
**LICENCIATURA EN CIBERSEGURIDAD**

**LABORATORIO #6**

**Asignación:**

PROGRAMACIÓN WEB EN PHP ORIENTADO A OBJETOS (POO)

**Curso:**

Desarrollo Web

**Estudiantes:**

Eduardo Samaniego 8-964-2469

**Profesor:**

Ing. José Javier Chirú F

**Grupo:**

1S3122

**2023**

## ÍNDICE

INTRODUCCIÓN .....	3
LABORATORIO.....	4
<b>Problema #1.....</b>	<b>4</b>
Captura antes de ejecución .....	4
Captura después de ejecución .....	4
Codificación .....	4
<b>Problema #2.....</b>	<b>6</b>
Captura antes de ejecución .....	7
Captura después de ejecución .....	7
Codificación .....	8
<b>Problema #3.....</b>	<b>10</b>
Captura antes de ejecución .....	11
Captura después de ejecución .....	11
Codificación .....	11
<b>Problema #4.....</b>	<b>13</b>
Captura antes de ejecución .....	14
Captura después de ejecución .....	14
Codificación .....	14
<b>Problema #5.....</b>	<b>16</b>
Captura antes de ejecución .....	17
Captura después de ejecución .....	17
Codificación .....	17
CONCLUSIONES .....	21
BIBLIOGRAFÍA .....	22

### **Introducción**

Programación Orientada a Objetos (POO) en PHP se ha consolidado como una metodología esencial. PHP, un lenguaje de programación ampliamente utilizado en la creación de aplicaciones web dinámicas, encuentra en la POO un enfoque estructurado que promueve la modularidad, la reutilización del código y la gestión eficiente de recursos. En este trabajo utilizaremos la POO para realizar 5 problemas de programación en PHP para la materia de desarrollo web.

## Laboratorio

### Problema #1

Elabore un programa que permita determinar la cantidad de números múltiplos de 3 comprendidos entre un rango de valores indicado por el usuario. Para ello, el usuario debe ingresar los topes inferior y superior.

### Captura antes de ejecución



### Captura después de ejecución



### Codificación

Calcularmultiplo.php

```
<?php
require_once('multiplos.php');
session_start();
if($_SERVER["REQUEST_METHOD"]=="POST"){
    $menor=$_POST['menor'];
    $mayor=$_POST['mayor'];
    $objMultiplos = new Multiplo($menor,$mayor);
    $cantidadMultiplos =$objMultiplos->calcularMultiplos();
    $_SESSION["cantidadMultiplos"]= $cantidadMultiplos;
    $_SESSION["menor"]=$menor;
    $_SESSION["mayor"]=$mayor;
}
```

```
header('Location:ejemplo1taller.php');

?>
```

Múltiplos.php

```
<?php
class Multiplo {
    private $menor;
    private $mayor;

    public function __construct($rmen, $rmay) {
        $this->menor = $rmen;
        $this->mayor = $rmay;
    }

    public function calcularMultiplos() {
        $rmen = $this->menor;
        $rmay = $this->mayor;

        if ($rmen <= $rmay) {
            $respuesta = 0;
            for ($i = $rmen; $i <= $rmay; $i++) {
                if (intval($i) % 3 == 0) {
                    $respuesta++;
                }
            }
            return $respuesta;
        } else {
            return "El rango mayor es más pequeño que el menor";
        }
    }
}

?>
```

Ejemplo1taller.php

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="css/bootstrap.min.css">
    <title>Taller Ejemplo 1</title>
```

```

</head>
<body>
    <?php session_start();?>
    <div class="container" align="center">
        <h1>Calculadora de Multiplos de 3</h1>
        <form action="calcularMultiplos.php" method="post">
            <div class="form-group col-md-5">
                <label>Ingrese el rango inferior </label>
                <input type="text" name="menor" class="form-control"
placeholder="Ingrese un número" value="<?php isset($_SESSION['menor'])?
print($_SESSION['menor']):"" ?>" >
            </div>
            <div class="form-group col-md-5">
                <label>Ingrese el rango superior </label>
                <input type="text" name="mayor" class="form-control"
placeholder="Ingrese un número" value="<?php isset($_SESSION['mayor'])?
print($_SESSION['mayor']):"" ?>" >
            </div>

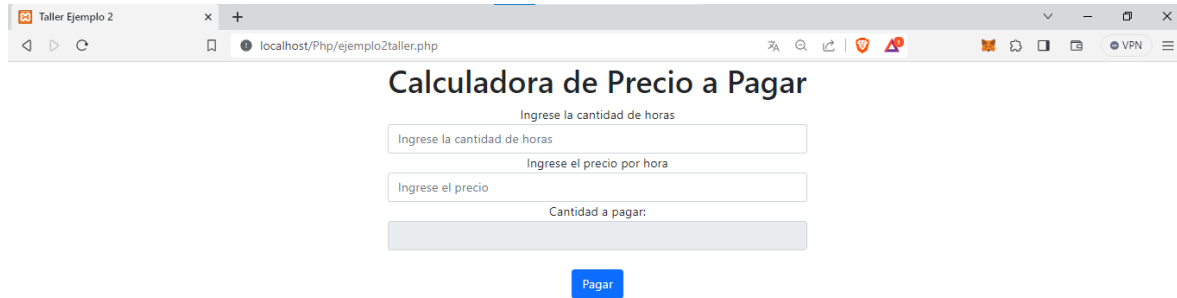
            <div class="form-group col-md-5">
                <label>Cantidad de Multiplos:</label>
                <input type="text" name="respuesta" class="form-control" disabled
value="<?php isset($_SESSION['cantidadMultiplos'])?
print($_SESSION['cantidadMultiplos']):"" ?>" >
                <br>
            </div>
            <button type="submit" class="btn btn-primary">Calcular
Cantidad</button>
        </form>
        <?php session_destroy() ?>
    </div>
</body>
</html>

```

## Problema #2

Calcule el salario semanal de un trabajador, a partir de las horas trabajadas y el precio cobrado por hora. Ambos valores deben ser solicitados al usuario. La jornada normal es de 40 horas semanales. Las horas extra se pagan un cincuenta por ciento más caras que las normales, y pasan a pagarse al doble que las normales a partir de las 50 horas trabajadas.

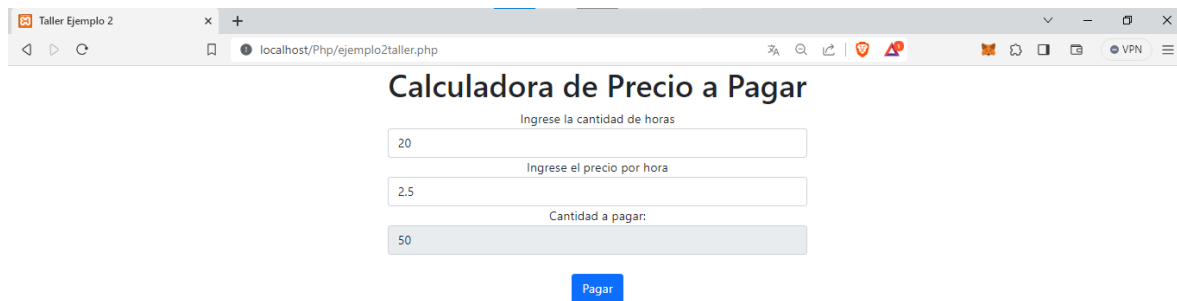
## Captura antes de ejecución



The screenshot shows a web browser window with the title 'Taller Ejemplo 2' and the URL 'localhost/Php/ejemplo2taller.php'. The page displays a form titled 'Calculadora de Precio a Pagar'. The form has three input fields: 'Ingrese la cantidad de horas' (empty), 'Ingrese el precio por hora' (empty), and 'Ingrese el precio' (empty). Below these is a label 'Cantidad a pagar:' followed by a greyed-out output field. A blue 'Pagar' button is at the bottom.

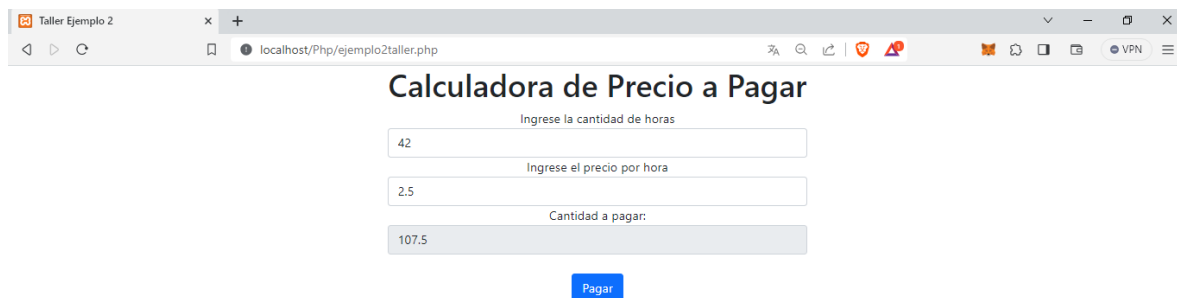
## Captura después de ejecución

### Condición 1



The screenshot shows the same form as before, but with values entered: '20' in the first field, '2.5' in the second field, and '50' in the third field. The 'Cantidad a pagar:' label is now above the output field, which now displays '50'. The 'Pagar' button remains at the bottom.

### Condición 2



The screenshot shows the same form with values entered: '42' in the first field, '2.5' in the second field, and '107.5' in the third field. The 'Cantidad a pagar:' label is now above the output field, which now displays '107.5'. The 'Pagar' button remains at the bottom.

### Condición 3

The screenshot shows a web browser window with the title 'Taller Ejemplo 2'. The address bar shows 'localhost/Php/ejemplo2taller.php'. The page content is a form titled 'Calculadora de Precio a Pagar'. It contains three input fields: 'Ingrese la cantidad de horas' with the value '55', 'Ingrese el precio por hora' with the value '2.5', and 'Cantidad a pagar:' with the value '181.25'. Below the fields is a blue button labeled 'Pagar'.

## Codificación

Ejemplo2taller.php

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="css/bootstrap.min.css">
  <title>Taller Ejemplo 2</title>
</head>
<body>
  <?php session_start();?>
  <div class="container" align="center">
    <h1>Calculadora de Precio a Pagar</h1>
    <form action="calcularPago.php" method="post">
      <div class="form-group col-md-5">
        <label>Ingrese la cantidad de horas </label>
        <input type="text" name="horas" class="form-control"
placeholder="Ingrese la cantidad de horas" value="<?php
isset($_SESSION['horas'])? print($_SESSION['horas']):""    ?>" >
      </div>
      <div class="form-group col-md-5">
        <label>Ingrese el precio por hora </label>
        <input type="text" name="precio" class="form-control"
placeholder="Ingrese el precio" value="<?php isset($_SESSION['precio'])?
print($_SESSION['precio']):""    ?>" >
      </div>

      <div class="form-group col-md-5">
        <label>Cantidad a pagar:</label>
        <input type="text" name="CantidadPago" class="form-control"
disabled value="<?php isset($_SESSION['cantidadPago'])?
print($_SESSION['cantidadPago']):""    ?>" >
        <br>
```



```

        </div>
        <button type="submit" class="btn btn-primary">Pagar</button>
    </form>
    <?php session_destroy() ?>
</div>
</body>
</html>

```

### Pago.php

```

<?php
class Pago {
    private $horas;
    private $precio;

    public function __construct($choras, $cprecio) {
        $this->horas = $choras;
        $this->precio = $cprecio;
    }

    public function calcularPago() {
        $choras = $this->horas;
        $cprecio = $this->precio;

        if (intval($choras) >= 0) {
            $cantidadPago = 0.0;
            if (intval($choras) <= 40){
                $cantidadPago = $choras * $cprecio;
            } elseif (intval($choras) > 40 && intval($choras) < 50){
                $restoHoras = $choras - 40;
                $nuevprecio = (($cprecio * 0.5) + $cprecio);
                $cantidadPago = (($cprecio * 40) + ($restoHoras *
($nuevprecio)));
            } elseif (intval($choras) >= 50){
                $extraNormales = $choras - 40;
                $extraMayores = $extraNormales - 10;
                $nuevPrecioNormales = (($cprecio * 0.5) + $cprecio);
                $nuevoPrecioMayores = ($cprecio * 2);
                $cantidadPago = (($cprecio * 40) + ($extraNormales *
($nuevPrecioNormales)) + ($extraMayores * ($nuevoPrecioMayores)));
            }

            return $cantidadPago;
        } else {
            return "No puedes tener horas negativas";
        }
    }
}

```

```
}  
}  
?>
```

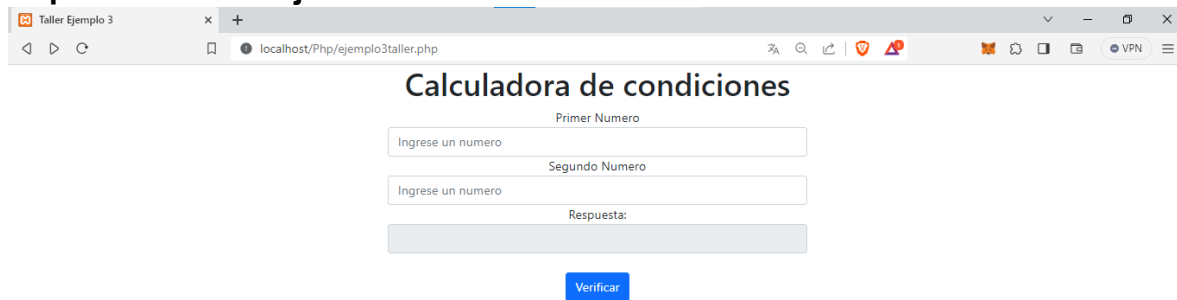
Calcularpago.php

```
<?php  
    require_once('pago.php');  
    session_start();  
    if($_SERVER["REQUEST_METHOD"]=="POST"){  
        $horas=$_POST['horas'];  
        $precio=$_POST['precio'];  
        $objPago = new Pago($horas,$precio);  
        $cantidadPago = $objPago->calcularPago();  
        $_SESSION["cantidadPago"] = $cantidadPago;  
        $_SESSION["horas"] = $horas;  
        $_SESSION["precio"] = $precio;  
    }  
    header('Location:ejemplo2taller.php');  
?>
```

**Problema #3**

Diseñe un programa que permita ingresar dos números enteros y determinar si el primer número está formado únicamente por cifras pares y si el segundo número está compuesto exclusivamente por cifras impares. Mostrar mensaje, indicando si cumplen o no con la condición

## Captura antes de ejecución



Taller Ejemplo 3

localhost/Php/ejemplo3taller.php

### Calculadora de condiciones

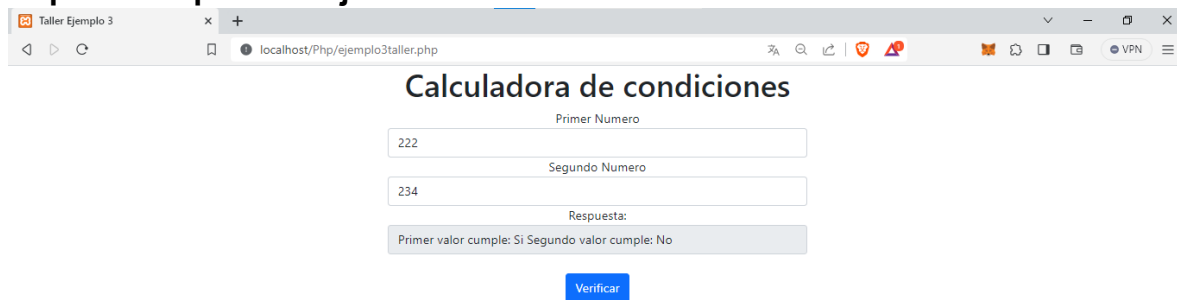
Primer Numero

Segundo Numero

Respuesta:

Verificar

## Captura después de ejecución



Taller Ejemplo 3

localhost/Php/ejemplo3taller.php

### Calculadora de condiciones

Primer Numero

Segundo Numero

Respuesta:

Verificar

## Codificación

### Ejemplo3taller.php

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="css/bootstrap.min.css">
  <title>Taller Ejemplo 3</title>
</head>
<body>
  <?php session_start();?>
```

```

<div class="container" align="center">
    <h1>Calculadora de condiciones</h1>
    <form action="calcularVer.php" method="post">
        <div class="form-group col-md-5">
            <label>Primer Numero</label>
            <input type="text" name="primerNumero" class="form-control"
placeholder="Ingrese un numero" value="<?php isset($_SESSION['primerNumero'])?
print($_SESSION['primerNumero']):"" ?>" >
        </div>
        <div class="form-group col-md-5">
            <label>Segundo Numero</label>
            <input type="text" name="segundoNumero" class="form-control"
placeholder="Ingrese un numero" value="<?php isset($_SESSION['segundoNumero'])?
print($_SESSION['segundoNumero']):"" ?>" >
        </div>

        <div class="form-group col-md-5">
            <label>Respuesta:</label>
            <input type="text" name="respuesta" class="form-control" disabled
value="<?php isset($_SESSION['condiciones'])?
print($_SESSION['condiciones']):"" ?>" >
            <br>
        </div>
        <button type="submit" class="btn btn-primary">Verificar</button>
    </form>
    <?php session_destroy() ?>
</div>
</body>
</html>

```

### Calcularver.php

```

<?php
    require_once('verificar.php');
    session_start();
    if($_SERVER["REQUEST_METHOD"]=="POST"){
        $primerNumero=$_POST['primerNumero'];
        $segundoNumero=$_POST['segundoNumero'];
        $objVer = new Verificar($primerNumero,$segundoNumero);
        $condiciones = $objVer->calcularVerificado();
        $_SESSION["condiciones"] = $condiciones;
        $_SESSION["primerNumero"]=$primerNumero;
        $_SESSION["segundoNumero"]=$segundoNumero;
    }
    header('Location:ejemplo3taller.php');

```

```
?>
```

Verificar.php

```
<?php
class Verificar {
    private $primerNumero;
    private $segundoNumero;

    public function __construct($pNumero, $sNumero) {
        $this->primerNumero = $pNumero;
        $this->segundoNumero = $sNumero;
    }

    public function calcularVerificado() {
        $respuestaPrimerNumero = "Si";
        $respuestaSegundoNumero = "Si";
        $pNumero = $this->primerNumero;
        $sNumero = $this->segundoNumero;
        $primero = str_split($pNumero);
        $segundo = str_split($sNumero);
        foreach ($primero as $posicion) {
            if ($posicion % 2 !== 0) {
                $respuestaPrimerNumero = "No";
            }
        }
        foreach ($segundo as $posicion2) {
            if ($posicion2 % 2 == 0) {
                $respuestaSegundoNumero = "No";
            }
        }
        $respuesta = "Primer valor cumple: " . $respuestaPrimerNumero . " Segundo
valor cumple: " . $respuestaSegundoNumero;
        return $respuesta;
    }
}
```

**Problema #4**

El domingo de Pascua es el primer domingo después de la primera luna llena posterior al equinoccio de primavera y se determina mediante el siguiente cálculo:

A= año resto 19

B = año resto 4

C = año resto 7

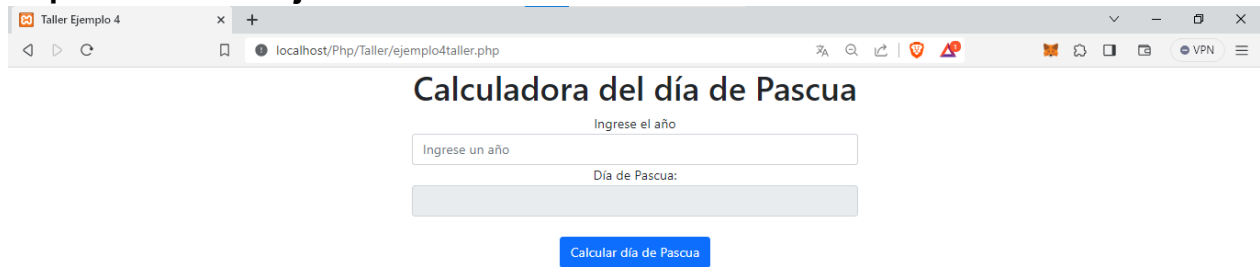
D= (19 \* A +24) resto 30

$$E=(2*B+4*C+6*D+5) \text{ resto } 7$$

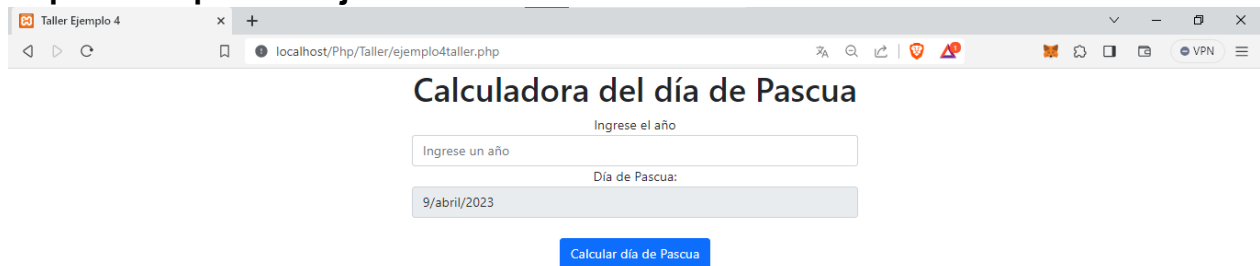
$$N=(22+D+E)$$

Donde N indica el número de día del mes de marzo, si es igual a o menor que 31; o abril, si es mayor que 31. Construir un programa orientado a objetos (POO) que determine la fecha del domingo de Pascua.

### Captura antes de ejecución



### Captura después de ejecución



### Codificación

Ejemplo4taller.php

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="css/bootstrap.min.css">
  <title>Taller Ejemplo 4</title>
</head>
<body>
  <?php session_start();?>
  <div class="container" align="center">
```

```

        <h1>Calculadora del día de Pascua</h1>
        <form action="calcularPascua.php" method="post">
            <div class="form-group col-md-5">
                <label>Ingrese el año </label>
                <input type="text" name="year" class="form-control"
placeholder="Ingrese un año" value="<?php echo isset($_SESSION['year']) ?
$_SESSION['year'] : "" ?>" >
            </div>

            <div class="form-group col-md-5">
                <label>Día de Pascua:</label>
                <input type="text" name="diaPascua" class="form-control" disabled
value="<?php echo isset($_SESSION['diaPascua']) ? $_SESSION['diaPascua'] : "" ?>"
>
                <br>
            </div>
            <button type="submit" class="btn btn-primary">Calcular día de
Pascua</button>
        </form>
        <?php session_destroy() ?>
    </div>
</body>
</html>

```

Pascua.php

```

<?php

class Pascua {
    private $year;

    public function __construct($year) {
        $this->year = $year;
    }

    public function calcularFechaPascua() {
        $A = $this->year % 19;
        $B = $this->year % 4;
        $C = $this->year % 7;
        $D = (19 * $A + 24) % 30;
        $E = (2 * $B + 4 * $C + 6 * $D + 5) % 7;
        $N = 22 + $D + $E;

        if($N > 31){
            $dia= $N-31;

```

```
        $diaPascua="$dia/abril/$this->year";
    }else{
        $diaPascua="$N/marzo/$this->year";
    }

    return $diaPascua;
}
?>
```

Calcularpascua.php

```
<?php
    require_once('pascua.php');
    session_start();
    if($_SERVER["REQUEST_METHOD"]=="POST"){
        $year=$_POST['year'];
        $objPascua = new Pascua($year);
        $diaPascua = $objPascua->calcularFechaPascua();
        $_SESSION["diaPascua"]= $diaPascua;
    }
    header('Location:ejemplo4taller.php');
?>
```

### Problema #5

Genere una matriz cuadrada ( $N \times N$ ) en la que se genere un valor aleatorio entre [1,100] en la esquina, mientras que los demás valores sean cero. El valor de N se lee a través de una entrada de datos (input text). Luego, sume los valores de las esquinas.



**Captura antes de ejecución**

Generado de Matriz

Ingrese el valor de N:

Ingrese N

Generar Matriz

**Captura después de ejecución**

Generado de Matriz

Ingrese el valor de N:

4

Generar Matriz

Matriz Generada:

83	0	0	27
0	0	0	0
0	0	0	0
6	0	0	90

Suma de las Esquinas:

206

**Codificación**

Ejemplo5taller.php

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="css/bootstrap.min.css">
    <title>Generar Matriz</title>
</head>
<body>
    <?php session_start(); ?>
    <div class="container" align="center">
        <h1>Generado de Matriz</h1>
        <form action="generarMatriz.php" method="post">
            <div class="form-group col-md-5">
                <label>Ingrese el valor de N:</label>

```

```

        <input type="text" name="n" class="form-control"
placeholder="Ingrese N" value="<?php isset($_SESSION['n'])?
print($_SESSION['n']):"" ?>" >
    </div>
    <br>

    <button type="submit" class="btn btn-primary">Generar Matriz</button>
</form>
</div>
<div>
    <?php if (isset($_SESSION['matriz'])): ?>
        <div class="container" align="center">
            <h2>Matriz Generada:</h2>
            <table class='table table-striped'>
                <?php for ($i = 0; $i < $_SESSION['n']; $i++): ?>
                    <tr>
                        <?php for ($j = 0; $j < $_SESSION['n']; $j++): ?>
                            <td><?= $_SESSION['matriz'][$i][$j] ?></td>
                            <?php endfor; ?>
                        </tr>
                    <?php endfor; ?>
                </table>

                <h2>Suma de las Esquinas:</h2>
                <p><?= $_SESSION['sumaEsquinas'] ?></p>
            </div>
        <?php endif; ?>
    </div>
    <?php session_destroy(); ?>
</body>
</html>

```

Matriz.php

```

<?php
class Matriz {
    private $n;
    private $matriz;

    public function __construct($n) {
        $this->n = $n;
        $this->generarMatriz();
    }
}

```

```

private function generarMatriz() {
    $this->matriz = array();

    for ($i = 0; $i < $this->n; $i++) {
        $fila = array();
        for ($j = 0; $j < $this->n; $j++) {
            if (($i == 0 && $j == 0) || ($i == 0 && $j == $this->n - 1) ||
($i == $this->n - 1 && $j == 0) || ($i == $this->n - 1 && $j == $this->n - 1)) {
                $fila[] = rand(1, 100);
            } else {
                $fila[] = 0;
            }
        }
        $this->matriz[] = $fila;
    }
}

public function sumarEsquinas() {
    $suma = 0;
    $suma += $this->matriz[0][0];
    $suma += $this->matriz[0][$this->n - 1];
    $suma += $this->matriz[$this->n - 1][0];
    $suma += $this->matriz[$this->n - 1][$this->n - 1];

    return $suma;
}

public function obtenerMatriz() {
    return $this->matriz;
}
}
?>

```

Generarmatriz.php

```

<?php
require_once('matriz.php');

session_start();

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $n = $_POST['n'];
    $objMatriz = new Matriz($n);
    $matriz = $objMatriz->obtenerMatriz();
}

```

```
$sumaEsquinas = $objMatriz->sumarEsquinas();

$_SESSION["matriz"] = $matriz;
$_SESSION["sumaEsquinas"] = $sumaEsquinas;
$_SESSION["n"] = $n;
}

header('Location: ejemplo5taller.php');
?>
```

### **Conclusiones**

A lo largo del desarrollo de este laboratorio pudimos aprender como utilizar de manera básica y efectiva el PHP para la realización de problemas con la programación orientada a objetos. Me sorprendió el hecho de que fuera tan parecido a Java, pude desarrollar un poco mas mi conocimiento en el área.

### **Bibliografía**

*PHP generar matriz de 2x2.* (s. f.). Stack Overflow en español.

<https://es.stackoverflow.com/questions/540242/php-generar-matriz-de-2x2>

por: Mehdi Achour. (s. f.). *PHP: manual de PHP - manual.*

<https://www.php.net/manual/es/index.php>

Soni, S. (2020, 26 octubre). *Estructuras y bucles de control de PHP: if, else, for, foreach, while,*

*y más.* Code Envato Tuts+. <https://code.tutsplus.com/es/php-control-structures-and-loops-cms-31999t>

*XAMPP installers and downloads for Apache Friends.* (s. f.).

<https://www.apachefriends.org/es/index.html>