

BENEMERITA UNIVERSIDAD AUTONOMA DE PUEBLA

Facultad: Facultad de ciencias de la computación

Materia: Desarrollo de Aplicaciones Web

Alumno: Alvarez Galicia Julio Cesar

Matricula 201911145

Profesor: Pedro Bello López

Tarea: Actividad 5



Facultad de Ciencias
de la Computación



Índice

BENEMERITA UNIVERSIDAD AUTONOMA DE PUEBLA.....	1
Introducción.....	3
Resumen	4
Capturas de pantalla de la ejecucion	5
Código HTML y PHP (p4.php)	7
Código PHP (p4a.php)	9
Conclusión.....	11
Referencias	12

Introducción

En el presente documento se muestra el código en HTML y PHP solicitando por el docente, el cual debe es:

Elaborar un programa en PHP con Programación Orientada a Objetos que realice las operaciones de Unión, Intersección y Diferencia entre Conjuntos.

- Deberá tener un programa con formulario para leer el número de datos del conjunto 1 y el número de datos del conjunto 2.
- Generar datos aleatorios en el rango 1 a 20 para almacenarse en dichos conjuntos.
- Mostrar las operaciones entre conjuntos descritas.

Ejemplo:

- Conjunto 1 de tamaño = 5, los datos aleatorios son: {5, 8 12, 8,7}
- Conjunto 2 de tamaño = 8, los datos aleatorios son: {2,8,7,16,19,19,1,17}
- Conjunto 3 (unión) = {5,8,12,7,2,16,19,1,17}
- Conjunto 4 (intercesión) = {8,7}
- Conjunto 5 (diferencia C1-C2) = {5,12}
- Conjunto 6 (diferencia C2-C1) = {2,16,19,1,17}

Resumen

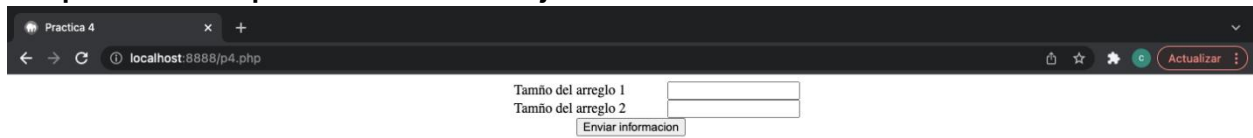
En el programa presentado lo primero fue investigar respecto a la creación de los arreglos, generar números aleatorios y las operaciones que se pueden realizar con ellos, al ser un arreglo se puede emplear teoría de conjuntos con las funciones próximamente mencionadas las cuales permiten realizar la operación Unión, Intersección y diferencia de conjuntos, Las funciones que se emplean son:

- `rand(rangoInferior, rangoSuperior)`: Operación empleada para generar números aleatorios entre un rango determinado el cual se define como argumento de la función.
- `array_intersect(arreglo, arreglo)`: Función para calcular la intersección entre dos arreglos.
- `array_merge(arreglo, arreglo, arreglo)`: Función que combina los elementos de uno o más arreglos juntándolos de modo que los valores de uno se anexan al final del anterior y retorna un arreglo resultante.
- `array_diff(arreglo, arreglo)`:función que calcula la diferencia entre dos arreglos.

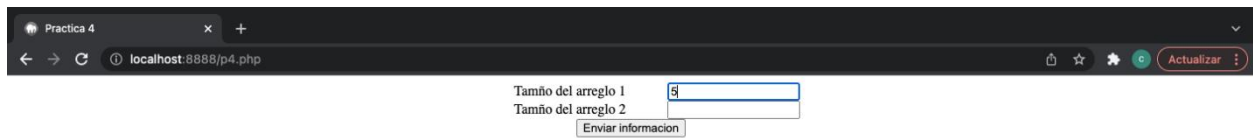
Posterior a la investigación de las funciones de conjuntos en PHP se analizo el programa proporcionado por el docente que esta realizado con funciones para poder implementar lo aprendido en el programa solicitado, dando como resultado los dos arreglos o conjuntos que se generan de tamaño n y m donde n y m son los tamaños proporcionados por el usuario, el conjunto intersección, unión y las diferencias de los conjuntos generados.

Para la creación y empleo de las funciones se trabajo con la clase `set` la cual fue diseñada con la finalidad de poder cumplir con la programación orientada a objetos permitiendo realizar la actividad solicitada

Capturas de pantalla de la ejecucion



A screenshot of a web browser window. The address bar shows 'localhost:8888/p4.php'. The page content includes two text input fields labeled 'Tamño del arreglo 1' and 'Tamño del arreglo 2'. Below these fields is a button labeled 'Enviar informacion'. The browser's developer tools are open, showing the 'Actualizar' button in the top right corner.



A second screenshot of the same web browser window. The address bar shows 'localhost:8888/p4.php'. The page content includes two text input fields labeled 'Tamño del arreglo 1' and 'Tamño del arreglo 2'. Below these fields is a button labeled 'Enviar informacion'. The browser's developer tools are open, showing the 'Actualizar' button in the top right corner.

Practica 4

localhost:8888/p4.php

Actualizar

Tamaño del arreglo 1

5

Tamaño del arreglo 2

7

Enviar informacion

Practica 4

localhost:8888/p4.php

Actualizar

Tamaño del arreglo 1

Tamaño del arreglo 2

Enviar informacion

El primer arreglo es:
Array ([0] => 12 [1] => 3 [2] => 5 [3] => 9 [4] => 10)

El segundo arreglo es:
Array ([0] => 14 [1] => 8 [2] => 13 [3] => 17 [4] => 18 [5] => 16 [6] => 16)

La interseccion de los arreglos es:
Array ()

La union de los arreglos es:
Array ([0] => 12 [1] => 3 [2] => 5 [3] => 9 [4] => 10 [5] => 14 [6] => 8 [7] => 13 [8] => 17 [9] => 18 [10] => 16)

La diferencia del primer arreglo con el segundo es:
Array ([0] => 12 [1] => 3 [2] => 5 [3] => 9 [4] => 10)

La diferencia del segundo arreglo con el primero es:
Array ([0] => 14 [1] => 8 [2] => 13 [3] => 17 [4] => 18 [5] => 16)

Código HTML y PHP (p4.php)

[illegible]

```
<?PHP
```

```
    if($_POST){  
        $size1=$_POST['n1'];  
        $size2=$_POST['n2'];  
        require_once "p4a.php";  
        echo"El primer arreglo es:<br>";  
        $set1=new set($size1);  
        $set1->show();  
        echo"<br><br>El segundo arreglo es:<br>";  
        $set2=new set($size2);  
        $set2->show();  
        echo"<br><br>La interseccion de los arreglos es:<br>";  
        $result=new set(1);  
        $result->interseccion($set1,$set2);  
        $result->show();  
        echo"<br><br>La union de los arreglos es:<br>";  
        $result->union($set1,$set2);  
        $result->show();  
        echo"<br><br>La diferencia del primer arreglo con el segundo es:<br>";  
        $result->diference($set1,$set2);  
        $result->show();  
        echo"<br><br>La diferencia del segundo arreglo con el primero es:<br>";  
        $result->diference($set2,$set1);  
        $result->show();  
    }  
}
```

```
?>
```


Código PHP (p4a.php)

<?PHP

```
class set{

    private $set=array();


    public function __construct($size){

        for($i=0;$i<$size;$i++){

            $this->set[]=rand(1,20);

        }

    }


    public function getData(){

        return $this->set;

    }


    public function show(){

        print_r ($this->set);

    }


    public function interseccion($set1,$set2){

        $this->set=array();

        $set3=$set1->getData();

        $set4=$set2->getData();

        $this->set=array_unique(array_intersect($set3,$set4));

    }


    public function union($set1,$set2){

        $this->set=array();
```

```

        $set3=$set1->getData();

        $set4=$set2->getData();

        $this->
>set=array_unique(array_merge(array_intersect($set3,$set4),array_diff($set3,$set4),array_diff($set4,$s
et3)));
    }

    public function difference($set1,$set2){
        $this->set=array();

        $set3=$set1->getData();

        $set4=$set2->getData();

        $this->set=array_unique(array_diff($set3,$set4));
    }
}

?>

```

Conclusión

Como conclusión respecto a la actividad solicitada se adquirió conocimiento respecto al manejo de arreglos y funciones de conjuntos en php además del manejo de funciones o módulos creados por el alumno, y como es la construcción de un programa en este lenguaje con el paradigma orientado a objetos.

Referencias

<https://www.php.net/>

<https://www.youtube.com/watch?v=e2bD6mhCVc4>

<https://www.youtube.com/watch?v=ZwVn9Yb1U-U>