



Actividad 5.

Integrantes:

- Escalante Hernández Alejandro.
- Moro Ortiz Uriel.
- Silva Ministro Eva.
- Zaca Tlacuilo José Manuel.

Materia: Aplicaciones web.

Periodo escolar: Primavera 2022.

Actividad 5.

Elaborar un programa en PHP con formularios y archivos de texto que permitan manipular las operaciones de unión, intersección y diferencia entre conjuntos, para lo cual es necesario.

- Crear la clase conjunto.
- Crear un formulario para pedir al usuario (cliente) el número de elementos del conjunto 1 y el número de elementos del conjunto 2.
- Almacenar datos aleatorios en el rango 1...20 para cada uno de los conjuntos.
- Escribir las operaciones indicadas entre conjuntos generando nuevos conjuntos.

A continuación se presenta el código correspondiente a la práctica:

Código en PHP

Conjunto

En este fragmento de código se muestra como la clase conjunto realiza las funciones: agregar, llenar, unión, intersección, diferencia y toString, esta última función nos servirá para almacenar los datos del arreglo.

```
1  <?php
2
3  class Conjunto{
4      private $elementos;
5      private $n;
6
7      public function __construct($n) {
8          $this->n=$n;
9          $this->elementos=array();
10         $this->llenar();
11     }
12
13     public function agregar($elemento){
14         if(!in_array($elemento, $this->elementos)){
15             $this->elementos[]=$elemento;
16             return true;
17         }
18         return false;
19     }
20
21     public function llenar(){
22         for($i=0;$i<$this->n;$i++){
23             $agregado=false;
24             while(!$agregado){
25                 $elemento=rand(1,20);
26                 $agregado=$this->agregar($elemento);
27             }
28         }
```

```

29     }
30
31     public function union($conjunto1, $conjunto2){
32         for($i=0;$i<$conjunto1->n;$i++){
33             $this->agregar($conjunto1->elementos[$i]);
34         }
35         for($i=0;$i<$conjunto2->n;$i++){
36             $this->agregar($conjunto2->elementos[$i]);
37         }
38
39     }
40
41     public function interseccion($conjunto1, $conjunto2){
42         for($i=0;$i<$conjunto1->n;$i++){
43             if(in_array($conjunto1->elementos[$i], $conjunto2->elementos)){
44                 $this->agregar($conjunto1->elementos[$i]);
45             }
46         }
47     }
48
49     public function diferencia($conjunto1, $conjunto2){
50         for($i=0;$i<$conjunto1->n;$i++){
51             if(!in_array($conjunto1->elementos[$i], $conjunto2->elementos)){
52                 $this->agregar($conjunto1->elementos[$i]);
53             }
54         }
55     }
56
57     public function toString(){
58         $string="";
59         for($i=0;$i<count($this->elementos);$i++){
60             $string=$string.$this->elementos[$i]." ";
61         }
62         return $string;
63     }
64 }
65
66 ?>

```

Principal

En este programa se leen los datos recibidos por el usuario y manda a llamar a la clase “Conjunto” para poder realizar las operaciones requeridas, posteriormente muestra los resultados en pantalla.

```

1  <?php
2
3      require_once "Conjunto.php";
4
5      $a = $_REQUEST['NumeroDeElementos_C1'];

```

```

6     $b = $_REQUEST['NumeroDeElementos_C2'];
7
8
9     $C1=new Conjunto($a);
10    echo "Conjunto 1 : ";
11    echo $C1->toString();
12    echo "<br>";
13
14    $C2=new Conjunto($b);
15    echo "Conjunto 2 : ";
16    echo $C2->toString();
17    echo "<br>";
18    echo "<br>";
19
20    $C3=new Conjunto(0);
21    $C3->union($C1,$C2);
22    echo "Union de C1 y C2: ";
23    echo $C3->toString();
24    echo "<br>";
25
26    $C4=new Conjunto(0);
27    $C4->interseccion($C1,$C2);
28    echo "Interseccion de C1 y C2: ";
29    echo $C4->toString();
30    echo "<br>";
31
32    $C5=new Conjunto(0);
33    $C5->diferencia($C1,$C2);
34    echo "Diferencia de C1 menos C2: ";
35    echo $C5->toString();
36    echo "<br>";
37
38    $C6=new Conjunto(0);
39    $C6->diferencia($C2,$C1);
40    echo "Diferencia de C2 menos C1: ";
41    echo $C6->toString();
42    echo "<br>";
43
44
45 ?>

```

Formulario

Aquí muestra la interfaz gráfica y pide al usuario que ingrese los datos para realizar las operaciones.

```
1 <!DOCTYPE html>
```

```

2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Formulario Operaciones de conjuntos</title>
8 </head>
9 <body>
10  <h3> Operaciones con conjuntos </h3>
11  <hr>
12  <FORM ACTION="Principal.php" METHOD="POST">
13    Introduce el numero de elementos del conjunto 1:
14    <INPUT TYPE="text" NAME="NumeroDeElementos_C1">
15    <br><br>
16    Introduce el numero de elementos del conjunto 2:
17    <INPUT TYPE="text" NAME="NumeroDeElementos_C2">
18    <br><br>
19    <INPUT TYPE="submit" VALUE=" Enviar">
20  </FORM>
21 </body>
22 </html>

```

Ejecución de programa en PHP.

En esta imagen se muestra la interfaz del formulario, donde pide al usuario que ingrese el número de elementos de dos conjuntos.

Operaciones con conjuntos

Introduce el numero de elementos del conjunto 1:

Introduce el numero de elementos del conjunto 2:

Después de dar “Enviar”, nos mostrara el número de elementos aleatorios de cada conjunto con sus respectivas operaciones.

Conjunto 1 : 2 14 5 7 16 8
Conjunto 2 : 12 6 17 2 10 1 13 19

Union de C1 y C2: 2 14 5 7 16 8 12 6 17 10 1 13 19
Interseccion de C1 y C2: 2
Diferencia de C1 menos C2: 14 5 7 16 8
Diferencia de C2 menos C1: 12 6 17 10 1 13 19

Bibliografía

https://www.php.net/manual/es/language.oop5.php?fbclid=IwAR0DACxb0qbszgEhzbpuHxnsaH9NFfxgfwXR6xS4bh_UzlgCmbHBUjaeQsA