

Programación Orientada a Objetos

Especialidad PHP



Felipe Izquierdo Romero

Índice

<u>1</u>	<u>Descripción del problema</u>
<u>2</u>	<u>Desarrollo</u>

1. Descripción del problema

Desarrollar un programa para una biblioteca donde un usuario ingrese unos datos de un libro o una revista y posteriormente se muestren por pantalla. Se deberá hacer uso de lo aprendido en el Lab para la realización.

Ambos ítems tendrán en común los datos:

- Título
- Autor
- Editor
- Año de publicación

Además, los libros tendrán el dato de número de paginas mientras las revistas tendrán como dato número de edición.

2. Desarrollo

Para comenzar y siguiendo la guía del problema realicé el formulario den HTML. Los campos en común con el atributo “required” para obligar al usuario a introducir un valor.

```
9      <h1>Formulario de Entrada de Datos</h1>
10     <form action="/index.php" method="post">
11
12         <label for="titulo">Título:</label>
13         <input type="text" id="titulo" name="titulo" required><br><br>
14
15         <label for="autor">Autor:</label>
16         <input type="text" id="autor" name="autor" required><br><br>
17
18         <label for="editor">Editor:</label>
19         <input type="text" id="editor" name="editor" required><br><br>
20
21         <label for="anio">Año de publicación:</label>
22         <input type="number" id="anio" name="anio" min="1700" max="2100" required><br><br>
23
24         <label for="numero_pg">Número de páginas:</label>
25         <input type="number" id="numero_pg" name="numero_pg"><br><br>
26
27         <label for="numero_ed">Número de edición:</label>
28         <input type="number" id="numero_ed" name="numero_ed"><br><br>
29
30         <input type="submit" value="Enviar">
31     </form>
```

Ilustración 1- Formulario

Dentro de este documento se encuentra el script que procesa los datos, donde condiono con bloques “if” y “else if” para validar los datos introducidos y determinar si es un libro, una revista o los campos número de edición o número de páginas están vacíos.

```

34 <?php
35     require_once('./clases/libro.php');
36     require_once('./clases/revista.php');
37
38     if ($_SERVER['REQUEST_METHOD'] == 'POST'){
39         $titulo = isset($_POST['titulo']) ? $_POST['titulo'] : '';
40         $autor = isset($_POST['autor']) ? $_POST['autor'] : '';
41         $editor = isset($_POST['editor']) ? $_POST['editor'] : '';
42         $anio = isset($_POST['anio']) ? $_POST['anio'] : '';
43         $numero_pg = isset($_POST['numero_pg']) ? $_POST['numero_pg'] : '';
44         $numero_ed = isset($_POST['numero_ed']) ? $_POST['numero_ed'] : '';
45     }
46
47     if ($numero_pg != '' && $numero_ed != '') {
48         echo 'Error <br> Un libro tiene numero de páginas <br> '.
49             'Una revista tiene número de edición <br>';
50     }
51     else if($numero_pg != ''){
52         $libro = new Libro ($titulo, $autor, $editor, $anio, $numero_pg);
53         $libro->getDatos();
54     }
55     else if($numero_ed != ''){
56         $revista = new Revista ($titulo, $autor, $editor, $anio, $numero_ed);
57         $revista->getDatos();
58     }
59     else{
60         echo'Rellene número de páginas o numero de edición';
61     }
62
63
64 ?>

```

Ilustración 2- Procesamiento de datos del formulario

Según el tipo de ítem que sea revista o libro, se crea un objeto con los datos y se llama a la función getDatos().

La clase Revista y Libro tienen la siguiente estructura.

```

1 <?php
2 require_once('item.php');
3 class Libro extends Item {
4
5     private $numero_pg;
6
7     public function __construct($titulo,$autor,$editor,$anio,$numero_pg) {
8         parent::__construct($titulo,$autor,$editor,$anio,$numero_pg);
9         $this->numero_pg = $numero_pg;
10    }
11
12    public function getDatos(){
13        parent::getDatos();
14        echo "Numero de paginas:" . $this->numero_pg . "<br>";
15    }
16 }
17 ?>

```

Ilustración 3- Clase Libro

```

1  <?php
2  require_once('item.php');
3  class Revista extends Item {
4
5      private $numero_ed;
6
7      public function __construct($titulo,$autor,$editor,$anio,$numero_ed) {
8          parent::__construct($titulo,$autor,$editor,$anio,$numero_ed);
9          $this->numero_ed = $numero_ed;
10     }
11
12     public function getDatos(){
13         parent::getDatos();
14         echo "Numero de edicion:" . $this->numero_ed . "<br>";
15     }
16 }
17 ?>

```

Ilustración 4- Clase Revista

Ambas clases extienden de la clase abstracta ítem, puesto que tienen muchos atributos en común y métodos en común también y solo una propiedad que las diferencia.

La clase ítem tiene la siguiente estructura.

```

1  <?php
2  abstract class Item{
3
4      protected $titulo;
5      protected $autor;
6      protected $editor;
7      protected $anio;
8
9      public function __construct($titulo,$autor,$editor,$anio) {
10         $this->titulo = $titulo;
11         $this->autor = $autor;
12         $this->editor = $editor;
13         $this->anio = $anio;
14     }
15
16     public function getDatos(){
17         echo "Titulo:" . $this->titulo . "<br>";
18         echo "Autor:" . $this->autor . "<br>";
19         echo "Editor:" . $this->editor . "<br>";
20         echo "Anio:" . $this->anio . "<br>";
21     }
22 }
23 ?>

```

Ilustración 5- Clase Ítem