

Programación Orientada a Objetos en PHP


Materia: Tecnologías y aplicaciones web
Profesor: MSI Mario Humberto Rodríguez
Chavez
Alumno: Luis Angel Torres Grimaldo

PROGRAMACIÓN ORIENTADA A OBJETOS (POO): Se puede decir que la POO es un conjunto de técnicas que ayudan a enfocar todas las variables pertenecientes a un problema o situación y de esta manera llegar a una solución.

La aplicación de esta técnica difiere un poco en a sintaxis de los muchos lenguajes que existen en el mercado, regularmente la diferencia serían simbologías que se utilizan para delimitar, tales como:

- {}
- :
- ;

Siempre recordando que la lógica es la misma, junto con las palabras clave, elementos, componentes y reglas del paradigma.



Las 4 características que definen este paradigma son:

1. Encapsulamiento
2. Herencia
3. Polimorfismo
4. Abstracción

Sin embargo todos los conceptos que abarcan el paradigma son los siguientes:

- Encapsulamiento
- Herencia
- Abstracción
- Modularidad
- Ocultación
- Polimorfismo
- Recolección de basura

Dentro de una clase se puede generar 3 distintos tipos de elementos:

1. Variables
2. Constantes
3. Métodos

Su sintaxis en PHP es la siguiente:

```
<?php
class bloques
{
    // declaración de una propiedad
    const CONSTANTE = 'mi constante';

    // declaración de una propiedad
    public $var = 'valor por defecto';

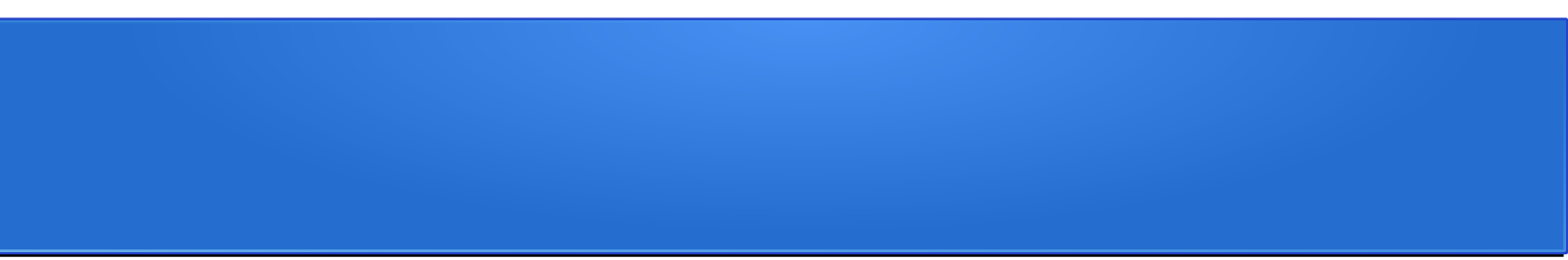
    // declaración un de un método
    public function imprimirVar() {
        echo $this->var;
    }
}

?>
```

Crea una clase PHP

Ahora se procede a mostrar la sintaxis de este paradigma en el lenguaje PHP:

```
1  <?php
2      class persona{
3          private $nombre;
4          private $edad;
5
6          public function persona($_nombre,$_edad){//Método constructor
7              $this->nombre=$_nombre;
8              $this->edad=$_edad;
9          }
10         public function mostrar_datos(){
11             echo "<br>Nombre: $this->nombre, edad: $this->edad<br><br>";
12         }
13     }
14
15     $persona_1 = new persona("Luis",22);
16     $persona_1->mostrar_datos();
17  ?>
```



Con respecto al concepto de ocultación existen 3 niveles que se aplican tanto en variables de la clase (atributos) como en las funciones de la clase (métodos). Son:

1. **Public.** A este tipo de elementos pueden acceder desde cualquier lugar, ya sea dentro de la clase o fuera por algún otro objeto de una clase distinta.
2. **Private.** Solo es visible y accesible dentro de la clase por los métodos que se implementan dentro.
3. **Protected.** Es parecido a **Public** solo que a esta accederán solo elementos que heredan la clase donde se definió el método o propiedad protegida o también se puede acceder desde la misma clase donde se declaró.

Método constructor en PHP

Otro punto importante dentro del paradigma orientado a objetos es **el método constructor**, el cual se encarga de inicializar las variables que se quieran. En el ejemplo anterior se representa la sintaxis de un constructor en PHP, como se puede observar el constructor debe ser declarado como público y debe tener el mismo nombre de la clase. Específicamente en PHP no es estrictamente necesario este método ya que de no declararse un método constructor se define un objeto de la siguiente forma:

```
1  <?php
2      class persona{
3          public $nombre;
4          public $edad;
5
6          #      public function persona($_nombre,$_edad){//Método constructor
7              #          $this->nombre=$_nombre;
8              #          $this->edad=$_edad;
9              #      }
10         public function mostrar_datos(){
11             echo "<br>Nombre: $this->nombre, edad: $this->edad<br><br>";
12         }
13     }
14
15     #      $persona_1 = new persona("Luis",22);
16     #      $persona_1->mostrar_datos();
17
18     $persona_2 = new persona();
19     $persona_2->nombre="Luis";
20     $persona_2->edad=22;
21     $persona_2->mostrar_datos();
22  ?>
```