

# P PROGRAMACION ORIENTADA A OBJETOS EN PHP



CRISTIAN ALEXANDER ZUÑIGA  
PALOMO

UNIVERSIDAD POLITECNICA  
DE VICTORIA

5/6/2019

## **PROGRAMACION ORIENTADA A OBJETOS EN PHP**

La POO es un paradigma de programación (o técnica de programación) que utiliza objetos e interacciones en el diseño de un sistema.

### **Características conceptuales de la POO**

La POO debe guardar ciertas características que la identifican y diferencian de otros paradigmas de programación. Dichas características se describen a continuación:

- **Encapsulamiento**  
Reúne al mismo nivel de abstracción, a todos los elementos que puedan considerarse pertenecientes a una misma entidad.
- **Abstracción**  
Aislación de un elemento de su contexto. Define las características esenciales de un objeto.
- **Polimorfismo**  
Es la capacidad que da a diferentes objetos, la posibilidad de contar con métodos, propiedades y atributos de igual nombre, sin que los de un objeto interfieran con el de otro
- **Modularidad**  
Característica que permite dividir una aplicación en varias partes más pequeñas (denominadas módulos), independientes unas de otras.
- **Ocultación**  
Los objetos están aislados del exterior, protegiendo a sus propiedades para no ser modificadas por aquellos que no tengan derecho a acceder a las mismas.
- **Polimorfismo**  
Es la capacidad que da a diferentes objetos, la posibilidad de contar con métodos, propiedades y atributos de igual nombre, sin que los de un objeto interfieran con el de otro.
- **Herencia**  
Es la relación existente entre dos o más clases, donde una es la principal (padre) y otras son secundarias y dependen (heredan) de ellas (clases “hijas”), donde a la vez, los objetos heredan las características de los objetos de los cuales heredan.

- **Recolección de basura**  
Es la técnica que consiste en destruir aquellos objetos cuando ya no son necesarios, liberándolos de la memoria.

## **CLASES O CLASES CONCRETAS**

Una clase es un modelo que se utiliza para crear objetos que comparten un mismo comportamiento, estado e identidad.

### **Objeto**

Es una entidad provista de métodos o mensajes a los cuales responde (comportamiento); atributos con valores concretos (estado); y propiedades (identidad).

### **Método**

Es el algoritmo asociado a un objeto que indica la capacidad de lo que éste puede hacer.

### **Propiedades y atributos**

Las propiedades y atributos, son variables que contienen datos asociados a un objeto

## **¿Qué es un objeto?**

- Un objeto permite generar orden al encapsular las variables.
- Las funciones no están agrupadas, no se sabe cuáles trabajan con que variables.
- Dos clases distintas pueden tener métodos con los mismos nombres (las funciones globales son únicas).
- Los objetos tienen una estructura definida mientras que los arreglos son más volátiles.

## **Palabras clave**

Abstracción: nuevos tipos de datos (el que tú quieras tú lo creas).

Encapsulación: organizar el código en grupos lógicos.

Ocultamiento: ocultar detalles de implementación y exponer solo los detalles que sean necesarios para el resto del sistema.

## **POO básica**

Una clase es una agrupación de variables, funciones y constantes.

Una variable en una clase se llama una propiedad.

Una función en una clase se llama un método.

Las clases son globales son accesibles desde cualquier lugar sin usar global.

Un objeto es una variable más (como los arreglos o enteros), solo que actúa como una copia de la clase.

Podemos tener todas las copias(objetos) que se desean de la misma clase.

Un objeto suele denominarse como una instancia de una clase.

## **Como de define una clase**

La definición básica de clases comienza con la palabra clave class, seguido por un nombre de clase, continuado por un par de llaves que encierran las definiciones de las propiedades y métodos pertenecientes a la clase. El nombre de clase puede ser cualquier etiqueta válida que no sea una palabra reservada de PHP. Un nombre válido de clase comienza con una letra o un guion bajo, seguido de la cantidad de letras, números o guiones bajos que sea.

## **Herencia de clases**

Los objetos pueden heredar propiedades y métodos de otros objetos. Para ello, PHP permite la “extensión” (herencia) de clases, cuya característica representa la relación existente entre diferentes objetos. Para definir una clase como extensión de una clase “padre” se utiliza la palabra clave extends.

## **Declaración de clases abstractas**

Las clases abstractas son aquellas que no necesitan ser instanciadas pero sin embargo, serán heredadas en algún momento. Se definen anteponiendo la palabra clave abstract.

Este tipo de clases, será la que contenga métodos abstractos (que veremos más adelante) y generalmente, su finalidad, es la de declarar clases “genéricas” que necesitan ser declaradas pero a las cuales, no se puede otorgar una definición precisa (No se pueden instanciar), de eso, se encargarán las clases que la hereden).

## ¿Cómo saber qué tipo de clase declarar?

Necesito...	Clases instantables	Abstracta	Heredada
Crear una clase que pueda ser instanciada y/o heredada	X		
Crear una clase cuyo objeto guarda relación con los métodos y propiedades de otra clase			X
Crear una clase que solo sirva de modelo para otra clase (clase padre), sin que pueda ser instanciada		X	
Crear una clase que pueda instanciarse pero que no pueda ser heredada por ninguna otra clase			

## OBJETOS E INSTANCIAS

Una vez que las clases han sido declaradas, será necesario crear los objetos y utilizarlos, aunque hemos visto que algunas clases, como las clases abstractas son solo modelos para otras, y por lo tanto no necesitan instanciar al objeto.

### Instanciar una clase

Para instanciar una clase, solo es necesario utilizar la palabra clave `new`. El objeto será creado, asignando esta instancia a una variable (la cual, adoptará la forma de objeto). Lógicamente, la clase debe haber sido declarada antes de ser instanciada, como se muestra a continuación:

### Tipos de propiedades

Las propiedades públicas se definen anteponiendo la palabra clave `public` al nombre de la variable. Éstas, pueden ser accedidas desde cualquier parte de la aplicación, sin restricción.

Las propiedades privadas se definen anteponiendo la palabra clave `private` al nombre de la variable. Éstas solo pueden ser accedidas por la clase que las definió.

Las propiedades protegidas pueden ser accedidas por la propia clase que la definió, así como por las clases que la heredan, pero no, desde otras partes de la aplicación. Éstas, se definen anteponiendo la palabra clave `protected` al nombre de la variable:

Las propiedades estáticas representan una característica de “variabilidad” de sus datos, de gran importancia en PHP. Una propiedad declarada como estática, puede ser accedida sin necesidad de instanciar un objeto y su valor es estático (es decir, no puede ser modificada para cada objeto, es como una variable global para

todas las instancias que se crean de ese objeto). Ésta, se define anteponiendo la palabra clave static al nombre de la variable:

## **CONSTANTES**

Otro tipo de “propiedad” de una clase, son las constantes, aquellas que mantienen su valor de forma permanente y sin cambios. A diferencia de las propiedades estáticas que pueden ser declaradas dentro de una clase, las constantes solo pueden tener una visibilidad pública y no deben ser creadas dentro de las clases. El acceso a constantes es exactamente igual que al de otras propiedades.

## **MÉTODOS PHP**

La forma de declarar un método es anteponiendo la palabra clave function al nombre del método, seguido por un par paréntesis de apertura y cierre y llaves que encierren el algoritmo.

Al igual que cualquier otra función en PHP, los métodos recibirán los parámetros necesarios indicando aquellos requeridos, dentro de los paréntesis.

Métodos públicos, privados, protegidos y estáticos

Los métodos, al igual que las propiedades, pueden ser públicos, privados, protegidos o estáticos. La forma de declarar su visibilidad tanto como las características de ésta, es exactamente la misma que para las propiedades.

## **Otros métodos mágicos**

PHP nos ofrece otros métodos mágicos tales como \_\_call, \_\_callStatic, \_\_get, \_\_set, \_\_isset, \_\_unset, \_\_sleep, \_\_wakeup, \_\_toString, \_\_invoke, \_\_set\_state y \_\_clone.

## **INTERFACES**

Una interfaz es un conjunto de métodos abstractos y de constantes cuya funcionalidad es la de determinar el funcionamiento de una clase, es decir, funciona como un molde o como una plantilla. Al ser sus métodos abstractos estos no tiene funcionalidad alguna, sólo se definen su tipo, argumento y tipo de retorno.

Para implementar una interface es necesario que la clase que quiera hacer uso de sus métodos utilice la palabra reservada implements. La clase que la implemente, de igual modo debe sobrescribir los métodos y añadir su funcionalidad.