

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

### **CLASES**

Son una plantilla que definen características y funciones.

### **CLASES ADSTRACTAS**

Las clases abstractas son clases que no pueden ser instanciadas directamente.

### **CONSTRUCTORES**

Los métodos constructores sirven para las clases. Aquellas que tengan un método constructor lo invocarán en cada nuevo objeto creado

### **VISIBILIDAD (modificadores de acceso)**

La visibilidad de una propiedad, un método o (a partir de PHP 7.1.0) una constante se puede definir

anteponiendo a su declaración una de las palabras reservadas public, protected y private.

- A los miembros de clase declarados como 'public' se puede acceder desde cualquier parte del código.
- A los miembros declarados como 'protected', solo desde la misma clase, mediante clases heredadas o desde la clase padre.
- A los miembros declarados como 'private' únicamente se puede acceder desde la clase que los definió.

### **ATRIBUTOS (datos)**

Características individuales que diferencian un objeto de otro y determinan su apariencia, estado u otras cualidades.

### **METODOS (funcionalidad)**

Se definen como métodos, a las funcionalidades y tareas que ejecutan los objetos

## ENCAPSULAMIENTO Y RELACIONES ENTRE CLASES EN PHP

En PHP, el encapsulamiento se refiere a ocultar los detalles de implementación de una clase y solo exponer una interfaz pública para interactuar con ella, utilizando modificadores de acceso como **public**, **private** y **protected**.

Las relaciones entre clases en la programación orientada a objetos se pueden clasificar en tres tipos:

- **Dependencia:** Una clase utiliza los servicios de otra sin una relación estructural permanente. Por ejemplo, una clase **CarritoCompras** puede depender de una clase **Producto**.
- **Agregación:** Una clase (agregador) contiene a otra clase (agregado) como parte de su estructura, pero la clase agregada puede existir de manera independiente. Por ejemplo, una clase **Equipo** puede tener varios objetos de la clase **Jugador**.
- **Composición:** Una clase (compuesta) contiene a otra clase (componente) como parte de su estructura, y la clase componente no puede existir independientemente de la clase compuesta. Por ejemplo, una clase **Casa** puede estar compuesta por objetos de la clase **Habitación**.

Estas relaciones entre clases tienen diferentes implicaciones y niveles de dependencia en el diseño y la implementación del código en PHP.

## **Objetos en PHP**

Los objetos en PHP son instancias de clases que contienen datos y comportamientos definidos en esa clase.

Se pueden acceder a las propiedades y métodos de una clase a través de los objetos.

Las clases son plantillas que definen la estructura y el comportamiento de un tipo de objeto.

### **Herencia en PHP:**

La herencia permite que una clase (subclase) herede propiedades y métodos de otra clase (superclase).

Se utiliza la palabra clave `extends` para establecer la herencia en PHP.

PHP permite la herencia única pero ofrece la implementación de múltiples interfaces.

### **Ejecución de Método:**

Llamar a un método en PHP implica crear un objeto de esa clase y luego llamar al método utilizando ese objeto.

Los métodos son funciones definidas dentro de una clase que pueden realizar ciertas tareas.

### **Operador de Resolución de Ámbito (::):**

Se utiliza en el contexto de POO para acceder a constantes, métodos estáticos y propiedades estáticas dentro de una clase sin necesidad de crear una instancia de la clase.

Permite acceder a funcionalidades compartidas que no dependen del estado de un objeto específico.

### **Palabra Reservada `static`:**

Se utiliza para declarar propiedades y métodos como estáticos dentro de una clase.

Las propiedades estáticas son compartidas entre todas las instancias de la clase y se accede a ellas directamente utilizando el operador de resolución de ámbito `::`.

Los métodos estáticos se pueden llamar directamente desde la clase y son útiles cuando la funcionalidad que proporcionan no depende del estado de un objeto específico.

En resumen, este documento proporciona una visión general de cómo trabajar con objetos en PHP, incluida la herencia, la ejecución de métodos, el operador de resolución de ámbito y el uso de la palabra reservada `static` para crear propiedades y métodos estáticos.

## **SELF, PARENT, POLIFORMISMO Y CONEXIÓN DE BASE DE DATOS EN PHP**

### **SELF**

La palabra clave self se utiliza dentro de la definición de una clase para acceder a propiedades y métodos estáticos desde el interior de la misma clase. Con self, puedes acceder a miembros estáticos (como propiedades, métodos y constantes) sin necesidad de crear una instancia de la clase.

### **PARENT**

La palabra clave parent se utiliza para acceder a propiedades y métodos de la clase padre (superclase) desde una clase hija (subclase). Permite extender o sobrescribir funcionalidad heredada.

### **POLIFORMISMO**

El polimorfismo es un concepto que permite que objetos de diferentes clases se comporten de manera similar a través de una interfaz común. En PHP, puedes lograr el polimorfismo mediante interfaces o clases abstractas.

### **CONEXIÓN DE BASE DE DATOS**

A diferencia de MySQLi, PDO está exclusivamente orientado a objetos y admite varios tipos de bases de datos diferentes (como MySQL, MSSQL, Informix y PostgreSQL) que utilizan PHP. Es una opción más flexible si necesitas trabajar con diferentes bases de datos en tus proyectos.