



# PROGRAMACIÓN ORIENTADA A OBJETOS EN PHP

Leonardo Daniel Alonso Cepeda

# ¿Qué es?

- La programación Orientada a objetos (POO) es una forma especial de programar, más cercana a como expresaríamos las cosas en la vida real que otros tipos de programación.
- Con la POO tenemos que aprender a pensar las cosas de una manera distinta, para escribir nuestros programas en términos de objetos, propiedades, métodos y otras cosas que veremos rápidamente para aclarar conceptos y dar una pequeña base que permita soltarnos un poco con este tipo de programación.



¿Cómo se  
compone?

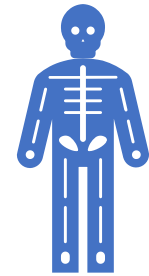
Este paradigma se  
compone de 4  
pilares:

- Encapsulamiento.
- Herencia.
- Polimorfismo.
- Abstracción.

# 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:

# CARACTERÍSTICAS CONCEPTUALES DE LA POO

**Abstracción:** Es la aislación de un elemento de su contexto.

Define las características esenciales de un objeto, sin que este muestre la forma de como lo hace, esto se hace con el uso de la palabra **abstract**.

**Encapsulamiento:** Es el ocultamiento del estado, es decir, de los datos miembro de un objeto de manera que solo se pueda cambiar mediante las operaciones definidas para ese objeto.

**Modularidad:** Es la característica que permite dividir una aplicación en varias partes más pequeñas denominadas módulos, independientes unas de otras.

# CARACTERÍSTICAS CONCEPTUALES DE LA POO



**Ocultación (aislamiento):** Es cuando 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 queda 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 cuando una clase nueva se crea a partir de una clase existente. La **herencia** (a la que habitualmente se denomina subclase) proviene del hecho de que la subclase (la nueva clase creada) contiene las atributos y métodos de la clase primaria.



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

# CARACTERÍSTICAS CONCEPTUALES DE LA POO

01

**Objeto:** Es una entidad creada a partir de la instancia de un objeto con el uso principal de un constructor de la clase a crear el objeto.

02

**Método:** Es el algoritmo asociado a un objeto que indica la capacidad de lo que éste puede hacer en la programación estructurada a esto se le llama función.

03

**Propiedades y atributos:** Las propiedades y atributos, son variables que contienen datos asociados a un objeto.

# CARACTERÍSTICAS CONCEPTUALES DE LA POO



**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.



**Declaración de Clases finales En PHP**



Estas se definen anteponiendo la palabra “final” y significa que no pueden ser heredadas por otras.



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



**Propiedades privadas:** 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.



**Propiedades protegidas:** 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.




# NIVELES DE ACCESO

- **Propiedades estáticas:** 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.

# ACCEDIENDO A LAS PROPIEDADES DE UN OBJETO

Se accede a una propiedad no estática dentro de la clase, utilizando la pseudo-variable **\$this** siendo esta pseudo-variable una referencia al objeto mismo, se debe tener en cuenta que la variable que se llamara no llevara adelante el \$:



Para tener acceso a una variable estática se debe de emplear el punto de :: anteponiendo la palabra self o parent dependiendo si se trata de una variable de la misma clase o de otra de la cual se ha heredado.

# ACCESO A VARIABLES DESDE EL EXTERIOR DE LA CLASE

Para acceder a una variable desde fuera de la clase se usa la siguiente sintaxis.

**\$objeto->variable**

Es importante mencionar que para acceder de esta forma la variable debe de ser publica.

Para acceder a una variable publica y estática es necesaria la siguiente sintaxis:

**Clase::\$variable\_estática**

# CONSTANTES

- Las constantes son aquellas que mantienen su valor inicial a lo largo de todo el programa y estas no pueden ser declaradas dentro de clases.

# METODOS

LOS MÉTODOS SON PRÁCTICAMENTE LO MISMO QUE UNA FUNCIÓN CON REFERENCIA A LA PROGRAMACIÓN ESTRUCTURADA, SOLO QUE ESTOS SON CREADOS DENTRO DE LAS CLASES PARA CUMPLIR CON LAS TAREAS HACIENDO USO O NO DE LAS PROPIEDADES O ATRIBUTOS DE LA CLASE.

EXISTEN TRES FORMAS DE DECLARARLOS:  
**STATIC, PROTECTED Y PRIVATE.**



# METODOS ABSTRACTOS

- Estos deben de ser definidos después de heredar de la clase padre y solo de esta forma se pueden usar.

# Métodos mágicos en PHP

- Son una variedad de métodos que incluye php y nos ayudan a realizar variedad de tareas de forma rápida y este vuelve al lenguaje más robusto.
- **El Método Mágico `__construct()`**
- El método `__construct()` será invocado de manera automático a la hora de crear una instancia de un objeto.
- **El Método magico `__destruct()`;**
- Es el encargado de liberar memoria en el caso de que ya no sea necesario utilizarla. Por ejemplo la destrucción de objetos.
- 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

- Funcionan como moldes o plantillas y al tener métodos abstractos estos no tienen funcionalidad, su tarea es la de declarar la forma la cual tendrá la clase y después de esto está será declarada para añadir funcionalidad.