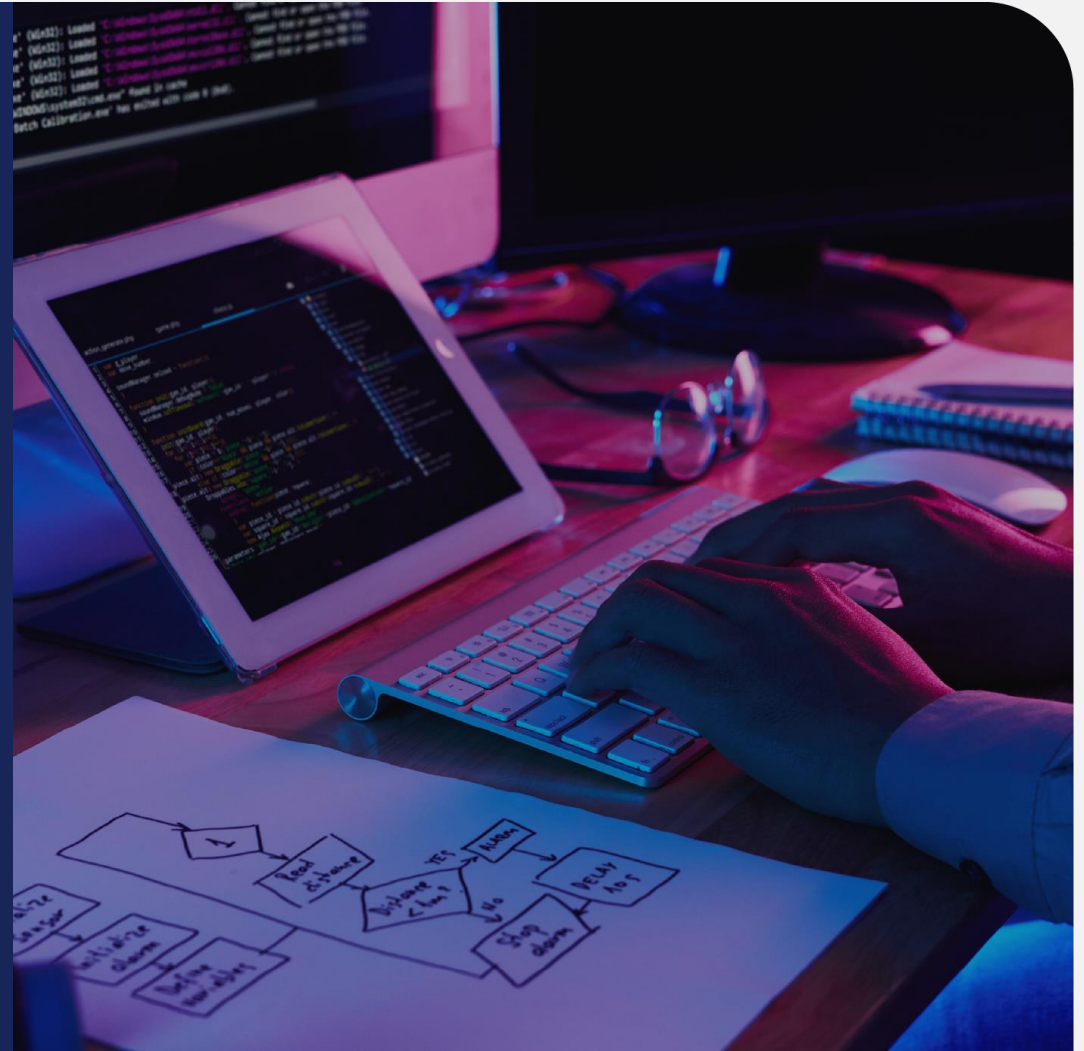


Capacitación PHP

Módulo: POO



Programación Orientada a Objetos

- Objetos que contienen data y funciones
- Fácil y rápido de ejecutar
- Estructura clara de programas
- D.R.Y
- Aplicaciones reutilizables

Pilares

- **Abstracción:** La abstracción oculta al usuario la funcionalidad interna de una aplicación.
- **Herencia:** Definir múltiples subclases a partir de una clase ya definida.
- **Polimorfismo:** Modificar ligeramente los métodos y atributos de las subclases previamente definidas en la superclase
- **Encapsulamiento:** La encapsulación es el proceso en el que protegemos la integridad interna de los datos en una clase

Clases y Objetos

- Una clase es una plantilla de un objeto
- Cuando se crea un objeto, hereda las propiedades y comportamientos de la clase.
- Cada objeto tendrá diferentes valores.

Constructor

- Permite inicializar las propiedades de un objeto al crear el objeto

`__construct()`



- Es una función, PHP la llama automáticamente cuando se crea un objeto.

Ejemplo Clase

```
• class Persona {  
    public $nombre;  
    public $edad;  
    public function __construct($nombre, $edad) {  
        $this->nombre = $nombre;  
        $this->edad = $edad;  
    }  
    public function presentacion() {  
        return "Mi nombre es: " . $this-> . " y tengo " . $this->edad . "  
        años!";  
    }  
}
```

Ejemplo Objeto

```
• $personaMilen = new Persona("Milen", "23");  
  echo $personaMilen -> presentacion();  
  echo "<br>";  
  $personaJuan = new Persona("Juan", "30");  
  echo $personaJuan -> presentacion();
```

Modificadores de Acceso

- **Public:** Se puede acceder a la propiedad o al método desde cualquier lugar. Default.
- **Protected:** Se puede acceder a la propiedad o método dentro de la clase y por clases derivadas de esa clase
- **Private:** Solo se puede acceder a la propiedad o método dentro de la clase

Herencia

- Clases derivadas de otras clases
- Clase hija hereda propiedades y métodos con acceso public y protected
- Se usa por medio de keyword ***extends***.

Clases y Métodos Abstractos

- Son clases padres que tienen métodos que están vacíos.
- Debe contener mínimo 1 método abstracto
- No pueden ser instanciadas
- Keyword: ***abstract***

```
<?php
abstract class ParentClass {
    abstract public function someMethod1();
    abstract public function someMethod2($name,
    $color);
    abstract public function someMethod3() :
    string;
}
?>
```

Reglas de Clases Abstractas

- El método de la clase hija debe definirse con el mismo nombre y re declarar el método abstracto de la clase padre
- El método de la clase hija debe definirse con el mismo modificador de acceso o uno menos restringido.
- El número de argumentos requeridos debe ser el mismo. Sin embargo, la clase secundaria puede tener argumentos opcionales además

Interfaces

- Permiten especificar qué métodos debe implementar una clase
- Las interfaces facilitan el uso de una variedad de clases diferentes de una misma manera.
- Keyword declaración: ***interface***
- Keyword para llamado: ***implements***

Interface Vs. Abstract Class

- Interfaces no tienen propiedades
- Todos los métodos de la interface son públicos
- Las clases pueden implementar una interfaz mientras heredan de otra clase al mismo tiempo