



|   |   |                          |   |
|---|---|--------------------------|---|
|  | <b>GUÍA DE TRABAJO N° 8 – GRADO 11</b>                          |                          |  |
|   | Articulación SENA   | Programación de Software |   |
|   | <b>Ing. Néstor Raúl Suarez Perpiñan</b><br><b>Página 1 de 6</b> |                          |   |

## GUIA 8 – PROGRAMACION ORIENTADA A OBJETOS EN PHP

### Objetivo:

- ✓ Conocer y analizar los principales conceptos de la programación Orientada a Objetos, aplicándolos en el lenguaje de Programación PHP

### I. DEFINICIONES BASICAS

**CLASE:** Una clase es el esquema o definición a partir de la cual se obtienen objetos. Una clase es una representación abstracta de un elemento o entidad real que contiene la definición de las características y comportamientos esenciales del elemento que representa. Una clase consta de tres elementos básicos:

**Un Nombre:** Debe estar asociado al elemento que representa. Por lo general el nombre de una clase lleva la primera letra en mayúscula.

**Atributos:** Son las características relacionadas con el objeto representado. Los atributos deben capturar la esencia de dicho objeto..

**Métodos:** Son funciones que modelan el comportamiento de un objeto. Pueden modelar cambios en los atributos de la clase o modelar relaciones entre clases a través del intercambio de mensajes. Los métodos pueden presentarse de cuatro tipos: Con o sin parámetros de entrada, de retorno o de no retorno.

Tanto métodos como atributos tienen una visibilidad asociada, la cual se presenta de tres formas.

**Publica (+):** representada por un signo más; los métodos y atributos pueden ser vistos por todos los objetos del sistema.

**Privada (-):** representada por el signo menos; los métodos y atributos solo pueden ser vistos por el objeto al cual pertenecen.



**Protegida (#):** representada por el signo número; los métodos y atributos pueden ser vistos por el objeto que los contiene y sus hijos (Lease Herencia).

En resumen una clase no es más que un conjunto de atributos (variables) y métodos (funciones) agrupadas en 1 entidad instanciable. Es la definición o declaración general de los atributos y métodos que describen un tipo de objeto.

**OBJETO:** Es la Instancia de una clase en tiempo de ejecución. En la mayoría de los casos Solo cuando un objeto es instanciado se puede hacer uso de los métodos públicos definidos en su clase.

**ENCAPSULAMIENTO:** Principio que consiste en ocultar al exterior de los objetos los atributos que utiliza para realizar sus tareas o funcionalidades. Este concepto se aplica para utilizar los objetos como si fueran cajas negras donde solo se conocen sus funcionalidades y no los atributos que utiliza.

**HERENCIA:** Cualidad de una clase mediante la cual hereda (puede usar) atributos y métodos de una clase padre que le transfiere todas sus definiciones.

|   |   |                          |   |
|---|---|--------------------------|---|
|  | <b>GUÍA DE TRABAJO N° 8 – GRADO 11</b>            |                          |  |
|   | Articulación SENA                                 | Programación de Software |   |
|   | Ing. Néstor Raúl Suarez Perpiñan<br>Página 2 de 6 |                          |   |

**CONSTRUCTOR:** Método que se ejecuta al instanciar un objeto, este método permite crear nuevos objetos a partir de una clase. El nombre de un constructor siempre coincide con el nombre de la clase.

## II. IMPLEMENTACIÓN DE LA POO EN PHP

### 1. Definición de una clase

```
<?php
class NombreClase
{
    private $atributo1;
    private $atributo2;

    public function Metodo1 ()
    {
        . . . ;
        . . . ;
    }

    public function metodo2 ($parametro1, $parametro2)
    {
        . . . ;
        . . . ;
    }
}
?>
```

### 2. Definición de una Herencia

```
<?php
include ("ClasePadre.php");
class ClaseHija extends ClasePadre
{
    . . .
    . . .
    . . .
}
?>
```

### 3. Establecer valor a un atributo dentro de una clase

```
$this->atributo = $_variable;
```



### 4. Instanciar una clase (Crear un nuevo objeto)

```
include ("NombreClase.php");
$NombreObjeto = new NombreClase();
```

### 5. Llamar métodos de un objeto

```
$NombreObjeto ->Metodo1 ();
```

```
$NombreObjeto ->Metodo2 ($parametro1, $parametro2
```

|   |   |                          |   |
|---|---|--------------------------|---|
|  | <b>GUÍA DE TRABAJO N° 8 – GRADO 11</b>            |                          |  |
|   | Articulación SENA                                 | Programación de Software |   |
|   | Ing. Néstor Raúl Suarez Perpiñan<br>Página 3 de 6 |                          |   |

## EJERCICIO

Tomando como base el ejercicio de validaciones con javascript de la guía 5 (funciones y validaciones) y las validaciones HTML5 de la guía 6 implemente el siguiente ejercicio.



Ubique el directorio de Trabajo del servidor apache (www) , Cree una carpeta y colóquela el nombre "SitioPOOPHP", dentro de esta cree dos carpetas y llámelas *Php\_Code* y *JavaScript* . Desarrolle dicho Sitio Web implementando los códigos que se muestran a continuación, pruebe el ejercicio variando las entradas y observe los resultados obtenidos

1. **Archivo JavaScript:** guárdelo dentro de la carpeta "JavaScript" con el nombre "funciones.js"

```

1. function ValidarText(elEvento, permitidos) {
2.     // Variables que definen los caracteres permitidos
3.     var numeros = "0123456789";
4.     var caracteres = " abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNÑOPQRSTUVWXYZ";
5.     var numeros_caracteres = numeros + caracteres;
6.     var teclas_especiales = [8, 37, 39, 46, 13];
7.     // 8 = BackSpace, 46 = Supr, 37 = flecha izquierda,
8.     // 13=Enter , 39 = flecha derecha
9.     // Seleccionar los caracteres a partir del parámetro de la función
10.    switch(permitidos)
11.    {
12.        case 'SoloNumeros':
13.            permitidos = numeros;
14.            break;
15.        case 'SoloLetras':
16.            permitidos = caracteres;
17.            break;
18.        case 'NumerosyLetras':
19.            permitidos = numeros_caracteres;
20.            break;
21.    }
22.    // Obtener la tecla pulsada
23.    var evento = elEvento || window.event;
24.    var codigoCaracter = evento.charCode || evento.keyCode;
25.    var caracter = String.fromCharCode(codigoCaracter);
26.    // Comprobar si la tecla pulsada es alguna de las teclas especiales
27.    // (teclas de borrado y flechas horizontales)
28.    var tecla_especial = false;
29.    for(var i in teclas_especiales) {
30.        if(codigoCaracter == teclas_especiales[i]) {
31.            tecla_especial = true;
32.            break;
33.        }
34.    }
35.    if (permitidos.indexOf(caracter) == -1) {
36.        if (tecla_especial==false) {
37.            alert('Caracter No valido para este campo, Por favor rectifique e
38.            intente nuevamente');
39.        }
40.    }
41.    // Comprobar si la tecla pulsada se encuentra en los caracteres permitidos
42.    // o si es una tecla especial
43.    return permitidos.indexOf(caracter) != -1 || tecla_especial;}

```

|   |   |                          |   |
|---|---|--------------------------|---|
|  | <b>GUÍA DE TRABAJO N° 8 – GRADO 11</b>                          |                          |  |
|   | Articulación SENA   | Programación de Software |   |
|   | <b>Ing. Néstor Raúl Suarez Perpiñan</b><br><b>Página 4 de 6</b> |                          |   |

2. **Clase Persona:** guárdela dentro de la carpeta “PHP\_CODE” con el nombre “Persona.php”

```
<?php
class Persona {

    private $cedula;
    private $nombre;

    public function SetCedula($_cedula) {
        $this->cedula=$_cedula; }

    public function getCedula(){
        return $this->cedula;
    }

    public function SetNombre($_nombre) {
        $this->nombre=$_nombre; }

    public function getNombre(){
        return $this->nombre;
    }

    public function CalcularEdad($ano_actual,$ano_naci) {
        $edad = $ano_actual - $ano_naci;
        return $edad;}

}
?>
```

3. **Clase Trabajador:** guárdela dentro de la carpeta “PHP\_CODE” con el nombre “Trabajador.php”



```
<?php
include ("Persona.php");

class Trabajador extends Persona
{
    private $cargo;

    public function setCargo($_cargo) {
        $this->cargo=$_cargo; }

    public function getCargo(){
        return $this->cargo;}

}
?>
```

|   |   |                                 |   |
|---|---|---------------------------------|---|
|  | <b>GUÍA DE TRABAJO N° 8 – GRADO 11</b>                          |                                 |  |
|   | <b>Articulación SENA</b>  | <b>Programación de Software</b> |   |
|   | <b>Ing. Néstor Raúl Suarez Perpiñan</b><br><b>Página 5 de 6</b> |                                 |   |

**4. Archivo PHP Principal:** guárdelo dentro de la raíz del sitio con el nombre “index.php”

```

<HTML>
<HEAD>
<TITLE>POO CON PHP</TITLE>
<script type="text/javascript" src="javascript/funciones.js"></script>
</HEAD>
<BODY>
<H1>Ejemplo POO CON PHP</H1>
<FORM ACTION="index.php METHOD="POST" onsubmit="return confirm('Desea continuar,
presione ACEPTAR, si no presione CANCELAR')">

<table width="40%" border="1" cellspacing="1" cellpadding="0">
  <tr>
    <td width="35%">1.Cedula: </td>
    <td width="65%"><INPUT TYPE="text" NAME="cedula" required="required"
onkeypress="return ValidarText(event, 'SoloNumeros')"></td>
  </tr>



  <tr>
    <td>2.Nombres: </td>
    <td><INPUT TYPE="text" NAME="nombre" required="required" onkeypress="return
ValidarText(event, 'SoloLetras')"></td>
  </tr>

  <tr>
    <td>3. Año de nacimiento: </td>
    <td><INPUT TYPE="text" NAME="ano_naci" required="required" onkeypress="return
ValidarText(event, 'NumerosyLetras')"></td>
  </tr>

  <tr>
    <td>4. Seleccione un cargo: </td>
    <td>
      <SELECT NAME="cargo">
        <OPTION VALUE="SEL" SELECTED>SELECCIONE
        <OPTION VALUE="Gerente">Gerente
        <OPTION VALUE="Administrativo">Administrativo
        <OPTION VALUE="Empleado ">Empleado
      </SELECT>
    </td>
  </tr>
</table>
<INPUT TYPE='SUBMIT' NAME='btn_enviar' VALUE='Enviar'>
</body>
</html>

<?php
if (isset($_POST['btn_enviar']))
{
  $nombre = $_POST['nombre'];
  $cedula = $_POST['cedula'];
  $cargo = $_POST['cargo'];
  $ano_naci= $_POST['ano_naci'];
}

```

|   |   |                                 |   |
|---|---|---------------------------------|---|
|  | <b>GUÍA DE TRABAJO N° 8 – GRADO 11</b>                          |                                 |  |
|   | <b>Articulación SENA</b>  | <b>Programación de Software</b> |   |
|   | <b>Ing. Néstor Raúl Suarez Perpiñan</b><br><b>Página 6 de 6</b> |                                 |   |

```

include ("Php_Code/Trabajador.php");
$objTrabajador=new Trabajador();

$objTrabajador->setNombre($nombre);
$objTrabajador->setCedula($cedula);
$objTrabajador->setCargo($cargo);

echo ("<Br><Br>");
echo ("DATOS INGRESADOS: <Br>");
echo ("<Br>");
echo ("Cedula: ");
echo $objTrabajador->Getcedula();
echo ("<Br>");
echo ("Nombres: ");
echo $objTrabajador->GetNombre();
echo ("<Br>");
echo ("Cargo: ");
echo $objTrabajador->GetCargo();
echo ("<Br>");
$ano_actual=2012;
$edad = $objTrabajador->CalcularEdad($ano_actual,$ano_naci);
echo ("<Br> Su EdAd es $edad");
}
?>

```

### TALLER

Realice un sitio PHP que satisfaga los siguientes requerimientos usando programación orientada a objetos es decir, por medio de clases y objetos

1. Permita al usuario seleccionar entre las siguientes operaciones matemáticas:
  - ✓ Sumar dos números
  - ✓ Restar dos números
  - ✓ Multiplicar dos números
  - ✓ Dividir dos números
  - ✓ Potenciación entre dos números
  - ✓ Seno de un numero
  - ✓ Coseno de numero
  - ✓ Factorial de un número
  - ✓ Serie de Fibonacci hasta un numero
2. Las operaciones básicas (Suma, resta, multiplicación y división) deben definirse en una clase padre y heredarse a una clase hija donde estarán definidas el resto de operaciones. La instancia en el formulario principal debe hacerse desde la clase hija.
3. Valide cada uno de los campos dependiendo del tipo de dato que corresponda
4. En todos los problemas Valide que ningún campo este vacío antes de realizar las operaciones.

**Nota:** Puede encontrar información sobre operaciones matemáticas en PHP a través del siguiente link. <http://php.net/manual/es/book.math.php>