



Guía Práctica de Ejercicios N° 9

En nuestra novena práctica seguiremos trabajando con la programación orientada a objetos, definiendo algunos conceptos fundamentales del paradigma POO que quedaron sin definir, como la herencia la visibilidad teniendo en cuenta la misma, las interfaces y algunas palabras claves como static y final.

Objetivo: Comprender y profundizar los conceptos más avanzados de la programación orientada a objetos, definiendo las clases y objetos con sus atributos y métodos. Definir, desarrollar y utilizar los métodos constructores, destructores, setter, getter y toString. Entender y usar parámetros por defecto en los métodos de una clase. Utilizar objetos como parámetros en los métodos de una clase. Herencia. Clases abstractas. Polimorfismo e Interfaces con PHP.

Introducción: En nuestra actualidad vemos que el desarrollo de Internet ha sido inminente y con ello las aplicaciones Web, por lo tanto, se hace indispensable el uso de un lenguaje que permita desarrollar aplicaciones Web como PHP, entre otros. Teniendo en cuenta la situación anterior es que veremos la como desarrollar aplicaciones Web que se ejecutarán del lado del servidor utilizando uno de los mejores lenguajes del ambiente del Software Libre. Teniendo en cuenta la magnitud del proyecto a desarrollar veremos la necesidad de emplear el paradigma orientado a objetos con PHP; que nos dará la posibilidad de diseñar, desarrollar y mantener el Software de manera profesional utilizando un paradigma antes mencionado.

Construcciones (palabras clave), variables globales y funciones que estudiaremos en esta unidad.

Construcciones del lenguaje con las que trabajaremos.

class nos permite crear una clase en PHP que es una colección de variables (atributos) y funciones (métodos) que trabajan con estas variables.

public nos permite definir atributos y métodos para que sean accedidos desde cualquier lugar.

private nos permite definir atributos y métodos para que sean accedidos desde la clase donde fueron definidos.

protected nos permite definir atributos y métodos para que sean accedidos desde la clase donde fueron definidos y sus descendientes.

final nos permite definir atributos y métodos para que sean accedidos desde cualquier lugar pero no pueden ser reescritos por sus descendientes.

new nos permite instanciar un objeto de acuerdo a una clase.

extends nos permite definir una clase que extiende de otra.



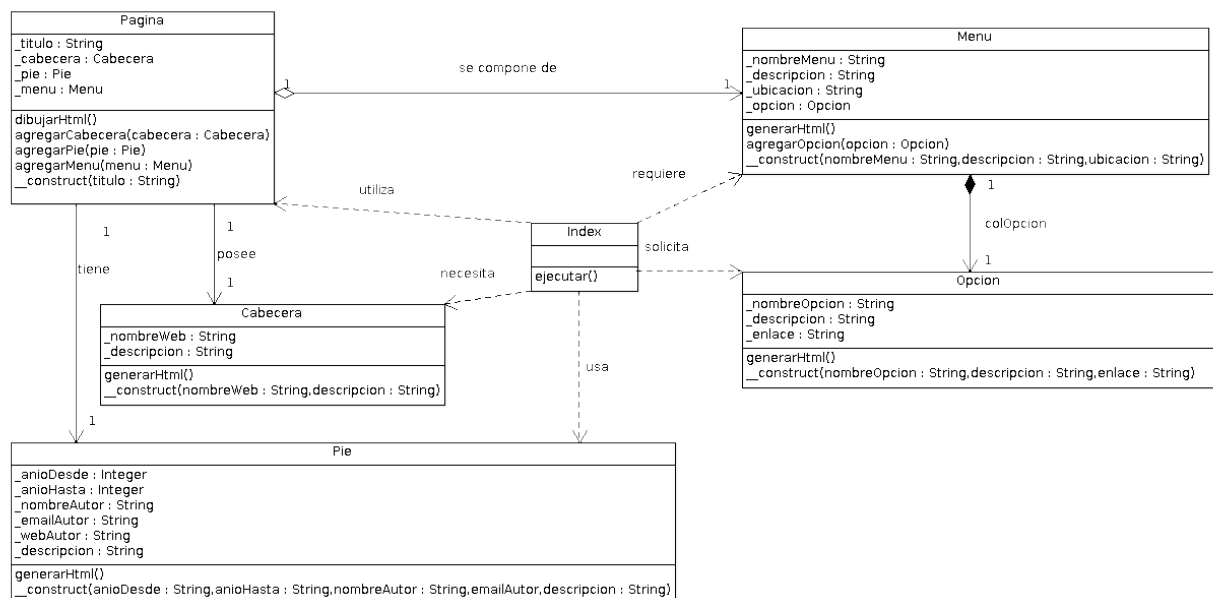
abstract nos permite definir una clase abstracta que no podrá ser instanciada pero se utilizará de molde para crear otras clases.

implements nos permite implementar el uso de interfaces.

Ejercicios

Utilizar los conceptos de programación Web y de base de datos

1) – Escriba los scripts PHP necesarios para completar el ejemplo de la clase teórica. A continuación el diagrama de clases:



index.php

```
<?php
```

```
require_once 'Pagina.php';
```

```
require_once 'Cabecera.php';
```

```
require_once 'Pie.php';
```

```
require_once 'Menu.php';
```

```
require_once 'Opcion.php';
```

```
class Index
```



```
{  
    public function ejecutar()  
    {  
        // Se crea el objeto de clase Pagina con un título  
        $oPagina = new Pagina('Prueba');  
        // Se crea la cabecera de la Página con nombre y descripción  
        $oCabecera = new Cabecera('Página Prueba', 'descripción de  
página de prueba');  
        // Se crea el menú de la página  
        $oMenu = new Menu('Principal', 'Menú principal', 'horizontal');  
        // Se crean las opciones de menú para la página  
        $oOpcionInicio = new Opcion('Inicio', 'Página de Inicio', 'index.php');  
        $oMenu->agregarOpcion($oOpcionInicio);  
        $oOpcionProductos = new Opcion('Productos', 'Página de Productos',  
'productos.php');  
        $oMenu->agregarOpcion($oOpcionProductos);  
        $oOpcionServicios = new Opcion('Servicios', 'Página de Servicios',  
'servicios.php');  
        $oMenu->agregarOpcion($oOpcionServicios);  
        $oOpcionEnlaces = new Opcion('Enlaces', 'Página de Enlaces',  
'enlaces.php');  
        $oMenu->agregarOpcion($oOpcionEnlaces);  
        $oOpcionContacto = new Opcion('Contacto', 'Página de Contacto',  
'contacto.php');  
        $oMenu->agregarOpcion($oOpcionContacto);  
        // Se crea el pie de página  
        $oPie = new Pie('2010', '2010', 'GUGLER', 'webmaster@gugler.com.ar',  
'http://www.gugler.com.ar', 'Todos los derechos reservados.');
```

```
        // Se agregan los objetos a la página
```



```
        $oPagina->agregarCabecera($oCabecera);

        $oPagina->agregarMenu($oMenu);

        $oPagina->agregarPie($oPie);

        // Se muestra el código HTML

        echo $oPagina->dibujarHtml();

    }

}

$oIndex = new Index();
$oIndex->ejecutar();

Pagina.php
<?php
require_once 'Cabecera.php';
require_once 'Pie.php';
require_once 'Menu.php';
class Pagina
{
    private $_titulo;
    private $_cabecera;
    private $_pie;
    private $_menu;
    public function __construct($titulo)
    {
        $this->_titulo = $titulo;
    }
    public function agregarCabecera(Cabecera $cabecera)
    {
        $this->_cabecera = $cabecera;
    }
    public function agregarPie(Pie $pie)
```



```
{
    $this->_pie = $pie;
}
public function agregarMenu(Menu $menu)
{
    $this->_menu = $menu;
}
public function dibujarHtml()
{
    $cadena = "";
    $cadena .= '<html><head><title>'.$this->_titulo.'</title></head>';
    $cadena .= '<body>';
    $cadena .= $this->_cabecera->generarHtml();
    $cadena .= $this->_menu->generarHtml();
    $cadena .= $this->_pie->generarHtml();
    $cadena .= '</body>';
    $cadena .= '</html>';
    return $cadena;
}
}
```

Cabecera.php

```
<?php
```

```
class Cabecera
```

```
{
    private $_nombreWeb;
    private $_descripcion;
    public function __construct($nombre, $descripcion)
    {
        $this->_nombreWeb = $nombre;
```



```
        $this->_descripcion = $descripcion;
    }
    public function generarHtml()
    {
        $cadena = "";
        $cadena .= '<h1>'.$this->_nombreWeb.'</h1>';
        $cadena .= '<br/>';
        $cadena .= '<h3>'.$this->_descripcion.'</h3>';
        $cadena .= '<br/>';
        $cadena .= '<hr/>';
        return $cadena;
    }
}

Menu.php
<?php
require_once 'Opcion.php';
class Menu
{
    private $_nombreMenu;
    private $_ubicacion;
    private $_descripcion;
    private $_opcion = array();
    public function __construct($nombreMenu, $ubicacion='horizontal', $descripcion)
    {
        $this->_nombreMenu = $nombreMenu;
        $this->_ubicacion = $ubicacion;
        $this->_descripcion = $descripcion;
    }
    public function agregarOpcion(Opcion $opcion)
```



```
{
    $this->_opcion[] = $opcion;
}

public function generarHtml()
{
    $cadena = "";
    foreach ($this->_opcion as $opcion) {
        $cadena .= $opcion->generarHtml().' ';
    }
    $cadena .= '<br/><hr/>';
    return $cadena;
}
}

Opcion.php
<?php
class Opcion
{
    private $_nombreOpcion;
    private $_descripcion;
    private $_enlace;
    public function __construct($nombreOpcion, $descripcion, $enlace)
    {
        $this->_nombreOpcion = $nombreOpcion;
        $this->_descripcion = $descripcion;
        $this->_enlace = $enlace;
    }
    public function generarHtml()
    {
        $cadena = "";
```



```
        $cadena .= "<a href='".$this->_enlace.'" title='".$this->_descripcion.'">".$this->_nombreOpcion."</a>";  
        return $cadena;  
    }  
}
```

```
}
```

Pie.php

```
<?php
```

```
class Pie
```

```
{
```

```
    private $_anioDesde;
```

```
    private $_anioHasta;
```

```
    private $_nombreAutor;
```

```
    private $_emailAutor;
```

```
    private $_webAutor;
```

```
    private $_descripcion;
```

```
    public function __construct($anioDesde, $anioHasta, $nombreAutor, $emailAutor,  
$webAutor, $descripcion)
```

```
    {
```

```
        $this->_anioDesde = $anioDesde;
```

```
        $this->_anioHasta = $anioHasta;
```

```
        $this->_nombreAutor = $nombreAutor;
```

```
        $this->_emailAutor = $emailAutor;
```

```
        $this->_webAutor = $webAutor;
```

```
        $this->_descripcion = $descripcion;
```

```
    }
```

```
    public function generarHtml()
```

```
    {
```

```
        $cadena = "";
```

```
        $cadena .= '<hr/>';
```




```
$cadena .= '<p>&copy;'.$this->_anioDesde.'-'.$this->_anioHasta;
$cadena .= ' '.$this->_nombreAutor.' - '.$this->_descripcion.'<br/>';
$cadena .= $this->_emailAutor.' - '.$this->_webAutor.'<br/>';
return $cadena;
}
}
```

2) – Teniendo en cuenta el ejemplo de la clase y el ejercicio anterior se le solicitará que agregue una clase Cuerpo (Cuerpo.php) y que ella se permita agregar párrafos y formularios. Genere las clases que se requieran para los párrafos y formularios.

index.php

```
<?php
require_once 'Pagina.php';
require_once 'Cabecera.php';
require_once 'Pie.php';
require_once 'Menu.php';
require_once 'Opcion.php';
require_once 'Cuerpo.php';
require_once 'Parrafo.php';
require_once 'Formulario.php';
class Index
{
    public function ejecutar()
    {
        // Se crea el objeto de clase Pagina con un título
        $oPagina = new Pagina('Prueba');
        // Se crea la cabecera de la Página con nombre y descripción
        $oCabecera = new Cabecera('Página de Prueba GUGLER',
'descripción de página de prueba de GUGLER');
        // Se crea el menú de la página
        $oMenu = new Menu('Principal', 'Menú principal', 'horizontal');
```



```
// Se crean las opciones de menú para la página
$opcionInicio = new Opcion('Inicio', 'Página de Inicio', 'index.php');
$menu->agregarOpcion($opcionInicio);
$opcionProductos = new Opcion('Productos', 'Página de Productos',
'productos.php');
$menu->agregarOpcion($opcionProductos);
$opcionServicios = new Opcion('Servicios', 'Página de Servicios',
'servicios.php');
$menu->agregarOpcion($opcionServicios);
$opcionEnlaces = new Opcion('Enlaces', 'Página de Enlaces',
'enlaces.php');
$menu->agregarOpcion($opcionEnlaces);
$opcionContacto = new Opcion('Contacto', 'Página de Contacto',
'contacto.php');
$menu->agregarOpcion($opcionContacto);
// Se crea el cuerpo de la página
$cuerpo = new Cuerpo();
$parrafo = new Parrafo('Texto de ejemplo dentro del párrafo. ');
$formulario = new Formulario();
$cuerpo->agregarParrafo($parrafo);
$cuerpo->agregarFormulario($formulario);
// Se crea el pie de página
$pie = new Pie('2010', '2010', 'GUGLER', 'webmaster@gugler.com.ar',
'http://www.gugler.com.ar', 'Todos los derechos reservados. ');
// Se agregan los objetos a la página
$page->agregarCabecera($cabecera);
$page->agregarMenu($menu);
$page->agregarCuerpo($cuerpo);
$page->agregarPie($pie);
```



```
// Se muestra el código HTML
echo $oPagina->dibujarHtml();
}
}

$oIndex = new Index();
$oIndex->ejecutar();
Pagina.php
<?php
require_once 'Cabecera.php';
require_once 'Pie.php';
require_once 'Menu.php';
require_once 'Cuerpo.php';
class Pagina
{
    private $_titulo;
    private $_cabecera;
    private $_pie;
    private $_menu;
    private $_cuerpo;
    public function __construct($titulo)
    {
        $this->_titulo = $titulo;
    }
    public function agregarCabecera(Cabecera $cabecera)
    {
        $this->_cabecera = $cabecera;
    }
    public function agregarPie(Pie $pie)
    {
```



```
        $this->_pie = $pie;
    }
    public function agregarMenu(Menu $menu)
    {
        $this->_menu = $menu;
    }
    public function agregarCuerpo(Cuerpo $cuerpo)
    {
        $this->_cuerpo = $cuerpo;
    }
    public function dibujarHtml()
    {
        $cadena = "";
        $cadena .= '<html><head><title>'.$this->_titulo.'</title></head>';
        $cadena .= '<body>';
        $cadena .= $this->_cabecera->generarHtml();
        $cadena .= $this->_menu->generarHtml();
        $cadena .= $this->_cuerpo->generarHtml();
        $cadena .= $this->_pie->generarHtml();
        $cadena .= '</body>';
        $cadena .= '</html>';
        return $cadena;
    }
}

Cabecera.php
<?php
class Cabecera
{
    private $_nombreWeb;
```



```
private $_descripcion;

public function __construct($nombre, $descripcion)
{
    $this->_nombreWeb = $nombre;
    $this->_descripcion = $descripcion;
}

public function generarHtml()
{
    $cadena = "";
    $cadena .= '<h1>'.$this->_nombreWeb.'</h1>';
    $cadena .= '<br/>';
    $cadena .= '<h3>'.$this->_descripcion.'</h3>';
    $cadena .= '<br/>';
    $cadena .= '<hr/>';
    return $cadena;
}
}
```

Menu.php

```
<?php
```

```
require_once 'Opcion.php';
```

```
class Menu
```

```
{
    private $_nombreMenu;
    private $_ubicacion;
    private $_descripcion;
    private $_opcion = array();
    public function __construct($nombreMenu, $ubicacion='horizontal', $descripcion)
    {
        $this->_nombreMenu = $nombreMenu;
```



```
        $this->_ubicacion = $ubicacion;
        $this->_descripcion = $descripcion;
    }
    public function agregarOpcion(Opcion $opcion)
    {
        $this->_opcion[] = $opcion;
    }
    public function generarHtml()
    {
        $cadena = "";
        foreach ($this->_opcion as $opcion) {
            $cadena .= $opcion->generarHtml().' ';
        }
        $cadena .= '<br/><hr/>';
        return $cadena;
    }
}

Opcion.php
<?php
class Opcion
{
    private $_nombreOpcion;
    private $_descripcion;
    private $_enlace;
    public function __construct($nombreOpcion, $descripcion, $enlace)
    {
        $this->_nombreOpcion = $nombreOpcion;
        $this->_descripcion = $descripcion;
        $this->_enlace = $enlace;
    }
}
```



```
}  
  
public function generarHtml()  
{  
    $cadena = "";  
    $cadena .= "<a href='".$this->_enlace.'" title='".$this->_descripcion.'">".$this->_nombreOpcion."</a>";  
    return $cadena;  
}  
}  
  
Cuerpo.php  
<?php  
require_once 'Parrafo.php';  
require_once 'Formulario.php';  
class Cuerpo  
{  
    private $_parrafo;  
    private $_formulario;  
    public function agregarParrafo(Parrafo $parrafo)  
    {  
        $this->_parrafo = $parrafo;  
    }  
    public function agregarFormulario(Formulario $formulario)  
    {  
        $this->_formulario = $formulario;  
    }  
    public function generarHtml()  
    {  
        $cadena = "";  
        $cadena .= $this->_parrafo->generarHtml();  
    }  
}
```



```
$cadena .= $this->_formulario->generarHtml();
```

```
return $cadena;
```

```
}
```

```
}
```

Formulario.php

```
<?php
```

```
class Formulario
```

```
{
```

```
    public function generarHtml()
```

```
    {
```

```
        $cadena = "";
```

```
        $cadena .= "<form action='procesoForm.php' method='POST'>";
```

```
        $cadena .= "<input type='reset' value='Limpiar'></input>";
```

```
        $cadena .= "<input type='submit' value='Enviar'></input>";
```

```
        $cadena .= "</form>";
```

```
        return $cadena;
```

```
    }
```

```
}
```

Parrafo.php

```
<?php
```

```
class Parrafo
```

```
{
```

```
    private $_textoParrafo;
```

```
    public function __construct($textoParrafo)
```

```
    {
```

```
        $this->_textoParrafo = $textoParrafo;
```

```
    }
```

```
    public function generarHtml()
```

```
    {
```




```
$cadena = "";  
$cadena .= '<p>'.$this->_textoParrafo.'</p>';  
return $cadena;  
}
```

```
}
```

Pie.php

```
<?php
```

```
class Pie
```

```
{
```

```
    private $_anioDesde;
```

```
    private $_anioHasta;
```

```
    private $_nombreAutor;
```

```
    private $_emailAutor;
```

```
    private $_webAutor;
```

```
    private $_descripcion;
```

```
    public function __construct($anioDesde, $anioHasta, $nombreAutor, $emailAutor,  
$webAutor, $descripcion)
```

```
    {
```

```
        $this->_anioDesde = $anioDesde;
```

```
        $this->_anioHasta = $anioHasta;
```

```
        $this->_nombreAutor = $nombreAutor;
```

```
        $this->_emailAutor = $emailAutor;
```

```
        $this->_webAutor = $webAutor;
```

```
        $this->_descripcion = $descripcion;
```

```
    }
```

```
    public function generarHtml()
```

```
    {
```

```
        $cadena = "";
```

```
        $cadena .= '<hr/>';
```



```
$cadena .= '<p>&copy;'.$this->_anioDesde.'-'.$this->_anioHasta;  
$cadena .= ' '.$this->_nombreAutor.' - '.$this->_descripcion.'<br/>';  
$cadena .= $this->_emailAutor.' - '.$this->_webAutor.'<br/>';  
return $cadena;  
}  
}
```

3) – Para los puntos 1 y 2 agregar el método `__toString()` mostrando información que identifique el objeto.

Pagina.php

```
public function __toString()  
{  
    return $this->_titulo;  
}
```

Cabecera.php

```
public function __toString()  
{  
    return $this->_nombreWeb.' - '.$this->_descripcion;  
}
```

Pie.php

```
public function __toString()  
{  
    return $this->_nombreAutor.' - '.$this->_email.' - '.$this->_webAutor;  
}
```

4) – Utilizando en ejemplo de clase y una vez que se trabajo con los puntos anteriores se la pedirá que incorpore una interfaz en el modelo que posee el método `generarHtml()`.

GeneraHtmlInterface.php

```
interface GeneraHtml  
{  
    public function generarHtml();  
}
```



Ejemplo de implementación de la interfaz en la clase Pie

Pie.php

```
<?php
require_once 'GeneraHtmlInterface.php';
class Pie implements GeneraHtml
{
    // Código de la clase
    public function generarHtml()
    {
        // Código del método
    }
}
```

