



Curso de Programación en PHP

Nivel I

Universidad Autónoma de Entre Ríos
Facultad de Ciencia y Tecnología - Oro Verde - v2.1



Capítulo 3: Programación Orientada a Objetos en PHP

Conceptos de programación orientada a objetos

Clases y objetos en PHP

Atributos y métodos de una clase

Métodos constructores, destructores y toString

**Parámetros por defecto y Objetos como
parámetros**



Clase 5: Conceptos de programación orientada a objetos

¿Qué es POO?

La programación orientada a objetos o POO es un paradigma de programación que utiliza objetos y las interacciones entre ellos, para diseñar tanto aplicaciones de escritorio como así también aplicaciones Web.

¿Por qué usar POO?

La programación orientada a objetos nos promete amplias mejoras en cuanto al diseño, desarrollo y el mantenimiento del Software.



Clase 5: Conceptos de programación orientada a objetos



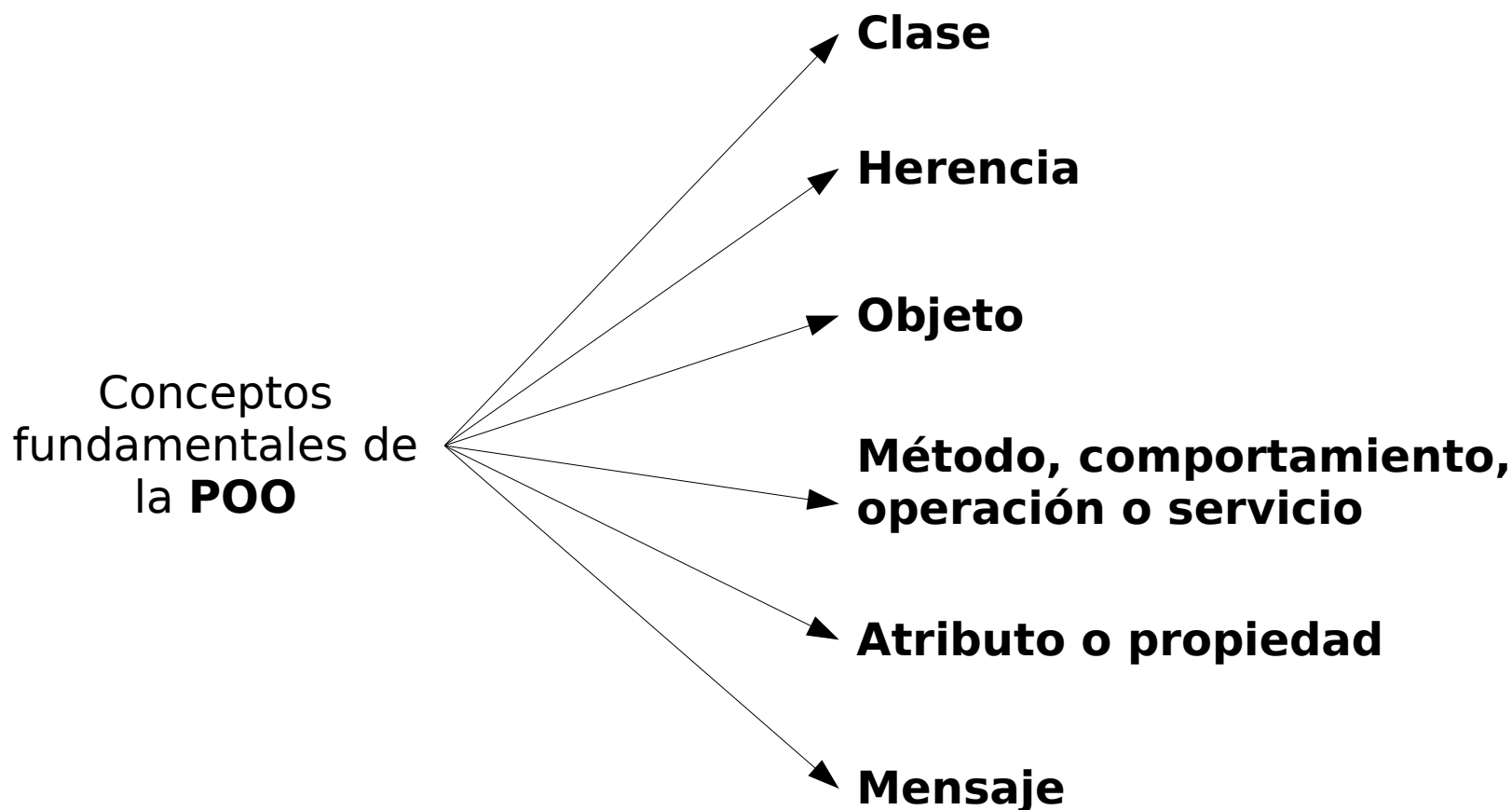
Con los objetos es realmente más fácil construir modelos (para sistemas complejos) que dedicarse a la programación secuencial.

David Taylor



Clase 5: Conceptos de programación orientada a objetos

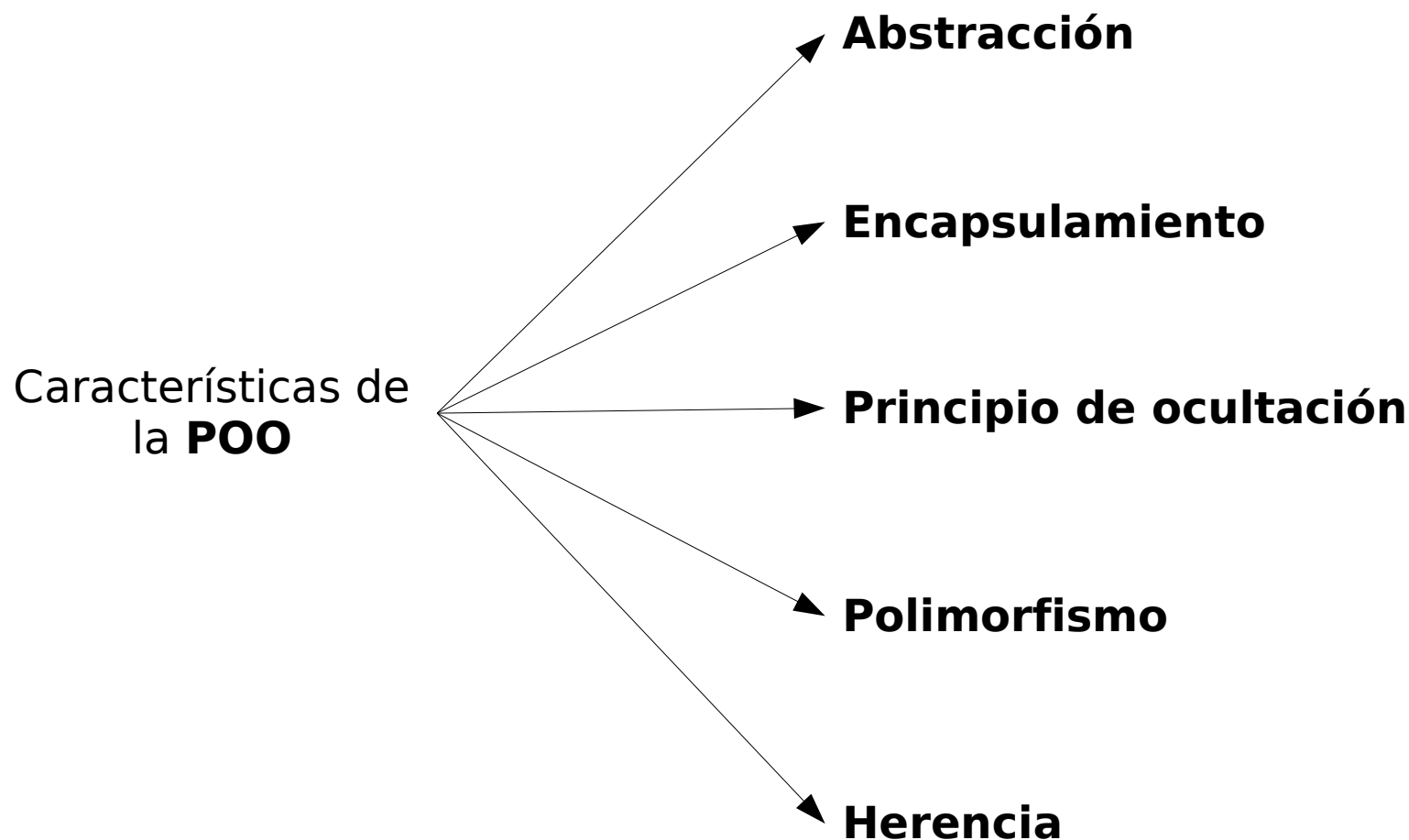
Conceptos fundamentales de la POO





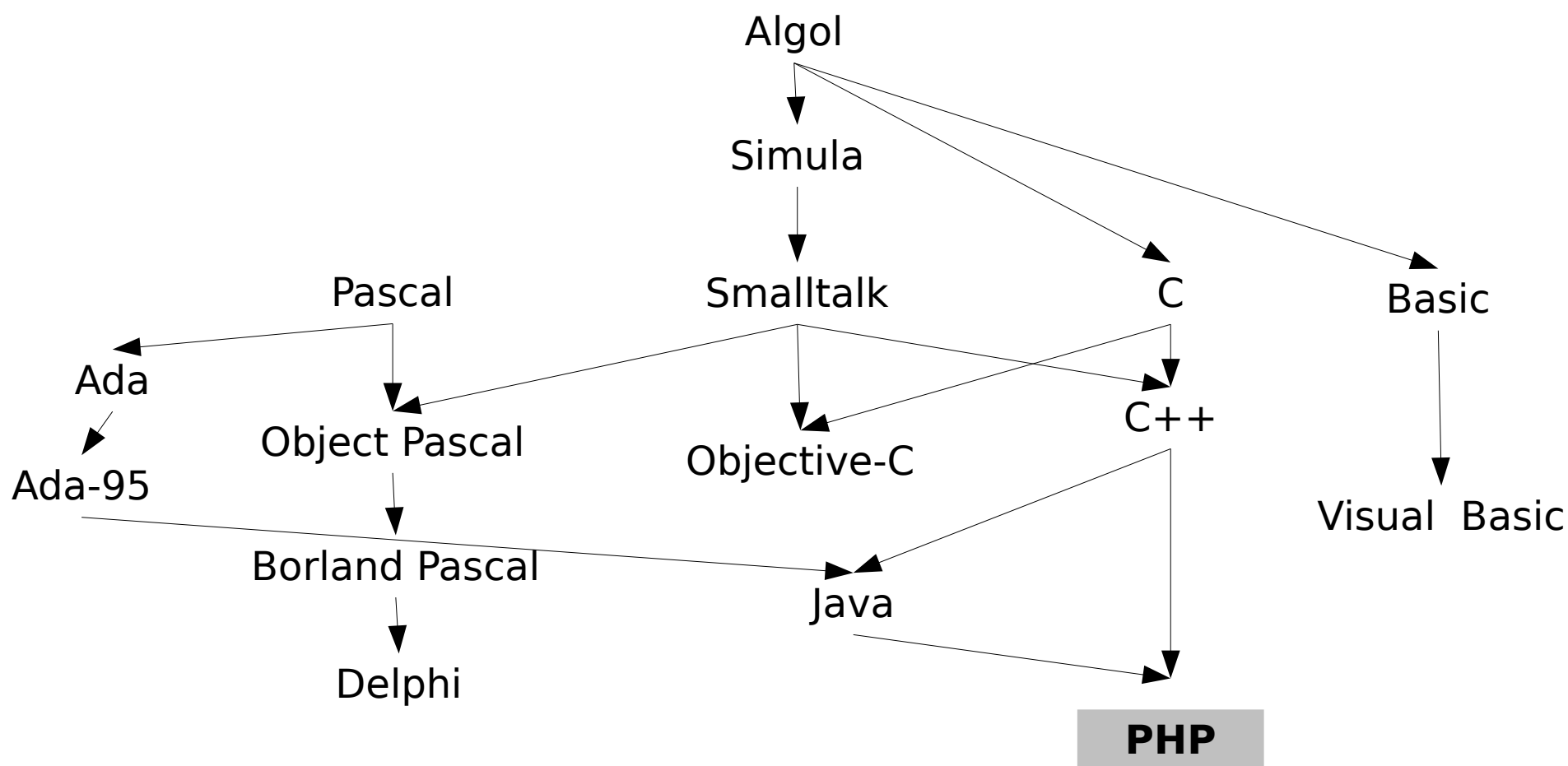
Clase 5: Conceptos de programación orientada a objetos

Características de la POO



Clase 5: Conceptos de programación orientada a objetos

Lenguajes de programación orientados a objetos





Clase 5: Conceptos de programación orientada a objetos

Conceptos fundamentales de la POO

¿Qué es una clase?

Una clase es un modelo de abstracción confeccionado a partir de objetos del mundo real.

Una clase define las propiedades y el comportamiento de un conjunto de objetos.

Una clase constituye una categoría de objetos y sirve para la creación de este tipo de objetos.



Clase 5: Conceptos de programación orientada a objetos

Conceptos fundamentales de la POO

¿Qué es un objeto?

Es cualquier entidad que posee un conjunto de propiedades o atributos (datos) y de comportamiento o funcionalidades (métodos) las cuales pueden interactuar entre si y reaccionar ante distintos eventos.

Los programas OO están compuestos por varios objetos que se comunican entre si mediante el envío de mensajes uno a otro.

Un programa OO verdadero constará de un grupo lógico de objetos los que se comunican ente sí para dar la funcionalidad total de dicho programa.



Clase 5: Conceptos de programación orientada a objetos



Mientras que un objeto es una entidad que existe en el tiempo y en el espacio, una clase representa sólo una abstracción, “la esencia” del objeto, si se puede decir así.

Grady Booch



Clase 5: Conceptos de programación orientada a objetos

Conceptos fundamentales de la POO

¿Qué es un atributo o propiedad?

Es el conjunto de valores o datos que definirán en un momento en particular un objeto de otro. Es el conjunto de características que diferencian a cada uno de los objetos de nuestra aplicación.



Clase 5: Conceptos de programación orientada a objetos

Conceptos fundamentales de la POO

¿Qué es un método o comportamiento?

Es una operación que un objeto puede realizar y mediante la cual se comunicará con otros objetos. Para cada uno de los objetos debemos preguntarnos ¿qué es lo que hace este objeto? y de esta forma podremos conocer de que manera interacciona con los demás objetos dentro del sistema.



Clase 5: Conceptos de programación orientada a objetos

Conceptos fundamentales de la POO

¿Qué es un mensaje?

Es la acción de efectuar una llamada a un método. Las acciones en POO se producen como respuesta a las peticiones de acciones, que se llaman *mensajes*.

Un objeto puede aceptar un mensaje y como respuesta realizará una acción y devolverá un valor.



Clase 5: Conceptos de programación orientada a objetos



Los mensajes y los métodos son dos caras de la misma moneda. Los métodos son los procedimientos invocados cuando un objeto recibe un mensaje.

Greg Voss



Clase 5: Conceptos de programación orientada a objetos

Conceptos fundamentales de la POO

¿Qué es herencia?

Es una relación entre clases, en la que una clase declara a otra como su principal.

Si se crea un objeto de la clase secundaria, hereda todas las propiedades de la clase principal además de las propiedades definidas en la propia clase.



Capítulo 3: Programación Orientada a Objetos en PHP

~~Conceptos de programación orientada a objetos~~

Clases y objetos en PHP

Atributos y métodos de una clase

Métodos constructores, destructores y toString

**Parámetros por defecto y Objetos como
parámetros**



Clase 5: Clases y objetos en PHP

Definición de clases en PHP

Definición 1

```
class Persona {  
    // Código de la clase  
}
```

Definición 2

```
class persona {  
    // Código de la clase  
}
```

Definición 3 (recomendada)

```
class Persona  
{  
    // Código de la clase  
}
```





Clase 5: Clases y objetos en PHP

Reglas para la definición de clases en PHP

Regla 1

Los nombres de clases son siempre *en singular*, por ejemplo: Persona, Auto, Perro, Libro, Usuario, etc.

Regla 2

La *primer letra* de cada palabra del nombre de la clase debe ir en *mayúsculas* (CamelCase).

Regla 3

Se debe generar *un script* PHP por cada clase definida.

Regla 4

El *nombre del script* PHP debería coincidir con el *nombre de la clase*.



Clase 5: Clases y objetos en PHP

Ejemplo de definición de clases en PHP

Persona.php

// Se define la clase Persona (Persona.php)

```
class Persona
{

    // Definición de atributos de la clase
    ...

    // Definición de métodos de una clase
    ...

} // Fin de la definición de la clase
```





Clase 5: Clases y objetos en PHP

Definición de un objeto en PHP

index.php

<?php

```
// Se requiere de la clase Persona (Persona.php)
```

```
require_once 'Persona.php';
```

```
// Se instancia el objeto $unaPersona de la clase Persona
```

```
$unaPersona = new Persona();
```

```
// Se instancia otro objeto $otraPersona de la clase Persona
```

```
$otraPersona = new Persona();
```

?>





Clase 5: Clases y objetos en PHP

Reglas para la definición de objetos en PHP

Regla 1

Los *nombres de objetos* son siempre *en singular*, por ejemplo: \$miPersona, \$miAuto, \$otroPerro, \$oLibro, \$usuario, etc.

Regla 2

La *primer letra* de cada palabra del nombre de un objeto debe ir en *mayúsculas* (CamelCase), menos la primer letra del nombre.

Regla 3

Se instancian dentro del mismo script PHP donde se definió clase, o bien, desde otro script PHP.



Clase 5: Clases y objetos en PHP

Ejemplo de definición de clases y objetos en PHP

Persona.php

<?php

// Se define la clase Persona (Persona.php)

class Persona

{

 // Definición de atributos y métodos de la clase

 ...

} // Fin de la definición de la clase

?>

index.php

<?php

// Se requiere de la clase Persona (Persona.php)

require_once 'Persona.php';

// Se instancia el objeto \$unaPersona de la clase Persona

\$unaPersona = **new** Persona();

?>





Clase 5: Clases y objetos en PHP

Funciones para el manejo de objetos en PHP

Ver el contenido y la estructura de un objeto

<?php

```
// Se requiere de la clase Persona (Persona.php)
require_once 'Persona.php';
// Se instancia el objeto $unaPersona de la clase Persona
$unaPersona = new Persona();
// Se muestra el contenido y la estructura de un objeto
var_dump($unaPersona);
```

Destruir un objeto en memoria

<?php

```
// Se requiere de la clase Persona (Persona.php)
require_once 'Persona.php';
// Se instancia el objeto $unaPersona de la clase Persona
$unaPersona = new Persona();
// Se destruye el objeto $unaPersona
unset($unaPersona);
```

?>





Clase 5: Clases y objetos en PHP



Programar sin una arquitectura o diseño en mente es como explorar una gruta sólo con una linterna: no sabes dónde estás, dónde has estado ni hacia dónde vas.

Danny Thorpe



Capítulo 3: Programación Orientada a Objetos en PHP

~~Conceptos de programación orientada a objetos~~

~~Clases y objetos en PHP~~

Atributos y métodos de una clase

Métodos constructores, destructores y toString

**Parámetros por defecto y Objetos como
parámetros**



Clase 5: Atributos y métodos de una clase

Definición de atributos en una clase

Definición

```
class Persona
{
    // Definición de atributos de la clase

    // Visibilidad pública
    public $nombreAtributoPublico;

    // Visibilidad privada
    private $_nombreAtributoPrivado;

    // Visibilidad protegida
    protected $nombreAtributoProtegido;
}
```





Clase 5: Atributos y métodos de una clase

Reglas para la definición de atributos en una clase

Regla 1

Los **nombres** de atributos que son de visibilidad **privada** deben definirse con un **guión bajo** (_) como primer carácter, por ejemplo: \$_apellidos, \$_nombres, \$_domicilio, \$_nombreUsuario, \$_edad, etc.

Regla 2

La **primer letra** de cada palabra del nombre de un atributo debe ir en **mayúsculas** (CamelCase), menos la primer letra del nombre.

Regla 3

Los atributos de una clase son por **defecto públicos**.



Clase 5: Atributos y métodos de una clase

Definición de métodos en una clase

Definición

```
class Persona
{
    // Definición de atributos de la clase

    // Visibilidad pública
    public $nombreAtributo;

    // Definición de métodos de la clase

    // Método público de la clase Persona
    public function nombreMetodo()
    {
        // Código del método
    }
}
```





Clase 5: Atributos y métodos de una clase

Reglas para la definición de métodos en una clase

Regla 1

Los **nombres** de métodos que son de visibilidad privada deben definirse con un **guión bajo** (_) como primer carácter, por ejemplo: `_mostrarApellidos`, `_obtenerDomicilio`, `_setearNombreUsuario`, `_calcularEdad`, etc.

Regla 2

La **primer letra** de cada palabra del nombre de un método debe ir en **mayúsculas** (CamelCase), menos la primer letra del nombre.



Clase 5: Atributos y métodos de una clase

Ejemplo de definición de atributos y métodos en una clase PHP

Persona.php

```
<?php
class Persona
{
    private $_apellidos;
    private $_nombres;
    private $_fechaNacimiento;

    public function decirEdad()
    {
        return $this->_calcularEdad();
    }

    private function _calcularEdad()
    {
        // Código del método que calcula la edad de la Persona
    }
}
```





Capítulo 3: Programación Orientada a Objetos en PHP

~~Conceptos de programación orientada a objetos~~

~~Clases y objetos en PHP~~

~~Atributos y métodos de una clase~~

Métodos constructores, destructores y toString

**Parámetros por defecto y Objetos como
parámetros**



Clase 5: Métodos constructores, destructores y toString

Método `__construct()` en una clase PHP

Persona.php

```
<?php
```

```
class Persona  
{
```

```
    private $_fechaNacimiento;
```

```
    // Se declara el constructor de la clase Persona
```

```
    public function __construct($fechaNacimiento)
```

```
    {
```

```
        $this->_fechaNacimiento = $fechaNacimiento;
```

```
    }
```

```
}
```

```
?>
```





Clase 5: Métodos constructores, destructores y toString

Método `__destruct()` en una clase PHP

Persona.php

```
<?php
```

```
class Persona
{
    private $_fechaNacimiento;

    // Se declara el destructor de la clase Persona
    public function __destruct()
    {
        $this->_fechaNacimiento = "";
    }
}
```

```
?>
```





Clase 5: Métodos constructores, destructores y toString

Método `__toString()` en una clase PHP

Persona.php

```
<?php
```

```
class Persona  
{
```

```
    private $_apellidos;  
    private $_nombres;
```

```
// Se declara el método toString de la clase Persona
```

```
public function __toString()
```

```
{  
    return $this->_apellidos.', '.$this->_nombres;  
}
```

```
}
```

```
?>
```





Clase 5: Métodos constructores, destructores y toString

Ejemplo de uso de un constructor, destructor y toString() en PHP

<?php

```
// Se requiere de la clase Persona (Persona.php)  
require_once 'Persona.php';
```

```
// Se instancia el objeto $oPersona de la clase Persona  
$oPersona = new Persona('23-04-1987');
```

```
// Se llama el método toString del objeto  
echo $oPersona->__toString();
```

```
// Se llama el método toString del objeto  
echo $oPersona;
```

```
// Se destruye el objeto $oPersona  
unset($oPersona);
```

?>





Capítulo 3: Programación Orientada a Objetos en PHP

~~Conceptos de programación orientada a objetos~~

~~Clases y objetos en PHP~~

~~Atributos y métodos de una clase~~

~~Métodos constructores, destructores y toString~~

**Parámetros por defecto y Objetos como
parámetros**



Clase 5: Parámetros por defecto y objetos como parámetros

Ejemplo de uso de un constructor con parámetros por defecto

Persona.php

<?php

```
class Persona
{

    private $_fechaNacimiento;

    // Se declara el constructor de la clase Persona
    public function __construct($fechaNacimiento='00-00-0000')
    {
        $this->_fechaNacimiento = $fechaNacimiento;
    }

}
```

?>





Clase 5: Parámetros por defecto y objetos como parámetros

Objetos como parámetros en los métodos

Persona.php

<?php

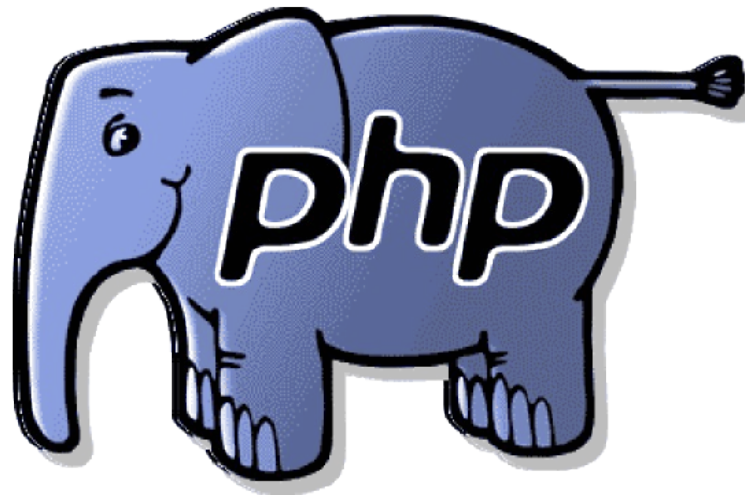
```
class Persona
{
    private $_nombres;
    private $_hermanos = array();

    // Se declara el agregar hermano de la clase Persona
    public function agregarHermano(Persona $oPersona)
    {
        $this->_hermanos[] = $oPersona;
    }
}
```

?>



¿Dudas?



¿Consultas?



Información de contacto

Web:

<http://www.gugler.com.ar>

<http://campusvirtual.gugler.com.ar>

<http://www.facebook.com/gugler.com.ar>

<http://www.twitter.com/cgugler>

Mail:

contacto@gugler.com.ar

academica@gugler.com.ar

administracion@gugler.com.ar