

**Universidad Nacional de La Matanza**

Departamento de Ingeniería e Investigaciones Tecnológicas

**TRABAJO PRÁCTICO N°1**

**GRUPO 6**

Materia: Paradigmas de Programación (3646)

Comisión: 02 - 2900

Grupo:

* Gomez Aylen Paula
* Redes Julian
* Rodriguez Christian Martín
* Pizzo Federico Daniel

Lenguaje asignado: PHP

Fecha de entrega: 30/05/2024

Índice

[Introducción 2](#_Toc167975971)

[Desarrollo 2](#_Toc167975972)

[Plataforma de uso 2](#_Toc167975973)

[Tipado 2](#_Toc167975974)

[Clases 3](#_Toc167975975)

[Visibilidad 3](#_Toc167975976)

[Atributos 3](#_Toc167975977)

[Métodos 4](#_Toc167975978)

[Creación y uso de objetos 4](#_Toc167975979)

[Herencia 5](#_Toc167975980)

[Traits 5](#_Toc167975981)

[Resolución de conflictos 6](#_Toc167975982)

[Herencia abstracta 7](#_Toc167975983)

[Interfaces 7](#_Toc167975984)

[Polimorfismo 9](#_Toc167975985)

[Polimorfismo mediante Interfaces 9](#_Toc167975986)

[Polimorfismo mediante Clases Abstractas 9](#_Toc167975987)

[Conclusión 10](#_Toc167975988)

[Referencias 10](#_Toc167975989)

# Introducción

El objetivo de esta investigación es explorar y comparar convenciones de nomenclatura, definiciones y cómo se modela la herencia en dos lenguajes de programación orientados a objetos, Java y PHP. Se resaltarán sus similitudes y diferencias con el fin de comprender mejor cada lenguaje y cómo estas diferencias pueden influir en su diseño e implementación.

# Desarrollo

## Plataforma de uso

En el caso de PHP es esencialmente un lenguaje de secuencias de comandos diseñado principalmente para el desarrollo web que se ejecuta desde el lado del servidor. Por otro lado, Java es más versátil y se utiliza en una variedad de aplicaciones, desde desarrollo de escritorio hasta desarrollo web. Es conocido por su capacidad de portabilidad debido a la JVM (Java Virtual Machine).

## Tipado

PHP es un lenguaje **débilmente tipado**, lo que significa que las variables no tienen un tipo definido y pueden cambiar de tipo durante la ejecución, en cambio Java es todo lo contrario, es un lenguaje **fuertemente tipado**, donde las variables deben ser declaradas con un tipo específico y no pueden cambiar de tipo.

* **PHP**:

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

* **Java:**

Texto

Descripción generada automáticamente

## Clases

**PHP**:

En PHP, al igual que en Java, las clases se declaran con la palabra clave **class**. Por ejemplo:

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

## Visibilidad

En PHP, se puede especificar la visibilidad de propiedades y métodos utilizando las palabras clave public, protected o private.

* **Public**: Indica que el elemento es accesible desde cualquier parte del código.
* **Protected**: significa que solo es accesible dentro de la clase y sus subclases. A diferencia de java, que utiliza protected para que se acceda en todo un paquete.
* **Private**: limita el acceso solo a la clase que lo define.

En el caso de las clases PHP solo admite clases tipo public, a diferencia de java, que pueden ser public y default (package).

## Atributos

En PHP, los atributos de una clase pueden definirse usando el signo de dólar **$**.

No es estrictamente necesario declarar el tipo de dato de un atributo ni su visibilidad al crear una clase. PHP asignará automáticamente una visibilidad por defecto (**public**) si no se especifica. Sin embargo, declarar explícitamente la visibilidad es una práctica recomendada para mejorar la claridad, seguridad y control de acceso a los atributos.

Por otro lado, en Java, los atributos se declaran con un **tipo** y **visibilidad** específico. Java no permite la existencia de variables sin un tipo explícito. Además, siempre se debe especificar la visibilidad de los atributos. Si no se especifica, se usa la visibilidad por defecto a nivel de paquete (package).

## Métodos

En PHP, los métodos se definen dentro de las clases utilizando la palabra clave **function**. Por ejemplo:

Texto

Descripción generada automáticamente

En Java, los métodos también se definen dentro de las clases, pero debe escribirse la visibilidad, el retorno y el nombre de la función.

En ambos casos utilizan la sintaxis Camel Case, y el nombre de la función empieza en minúscula.

## Creación y uso de objetos

En ambos lenguajes, los objetos se crean utilizando la palabra clave ***new***. Por ejemplo:

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

## Herencia

Diferencias y Similitudes entre Java y PHP:

* Sintaxis: La sintaxis para definir y usar la herencia es similar en ambos lenguajes, utilizando la palabra clave **extends** en Java y PHP.
* Herencia Múltiple: En Java, una clase solo puede tener una superclase directa (herencia simple). En PHP, la herencia múltiple de clases tampoco es permitida. En ambos lenguajes una clase puede implementar múltiples interfaces, lo que proporciona cierta flexibilidad en la reutilización de código.
* Constructores: Ambos lenguajes permiten la herencia de constructores, lo que significa que una subclase puede llamar al constructor de su superclase utilizando la palabra clave **parent::\_\_construct()** en PHP o **super()** en Java.
* Polimorfismo: Ambos lenguajes admiten el polimorfismo, lo que significa que una subclase puede ser tratada como su superclase, lo que permite la flexibilidad en el diseño del código y la reutilización de clases.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

## Traits

Si bien PHP no admite herencia múltiple, se puede simular mediante el uso de **traits**. Los traits permiten la reutilización de métodos en varias clases, su objetivo es reducir las limitaciones propias de la herencia simple.

Los traits pueden tener atributos y métodos, pero no podemos tener un constructor y tampoco podemos heredar de otro trait o clase.

La ventaja que nos ofrece los Traits es la reutilización de código ya que podemos reutilizar métodos en múltiples clases y la flexibilidad de tener todos los métodos en una sola clase y combinarlos.

Texto

Descripción generada automáticamente

A la hora de utilizar un método del trait y necesitamos cambiar la visibilidad del método podemos utilizar la palabra reservada **as.**

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

### Resolución de conflictos

Si dos Traits insertan un método con el mismo nombre, se produce un error y para resolverlo se debe usar el operador **insteadof** para elegir unívocamente uno de los métodos conflictivos en la clase que intenta heredarlos.

Como esto solamente permite excluir métodos, se puede utilizar el operador **as** para añadir un alias a uno de los métodos.

Por ejemplo: Tenemos dos traits “Espaniol” e “Ingles”, ambos tienen un método saludar.

Texto

Descripción generada automáticamente

Luego, creamos una clase Persona que usa ambos traits, pero al tener mismo nombre de método, en la línea 16 elegimos Espaniol para conservar el nombre “saludar” y en la línea 17 renombramos el método saludar de Ingles.

Al hacer esto ya se nos admite llamar a ambos métodos desde la clase Persona.

Texto

Descripción generada automáticamente

## Herencia abstracta

* En ambos lenguajes, la palabra clave **abstract** se utiliza para definir clases y métodos abstractos.
* Ambos lenguajes permiten declarar métodos abstractos dentro de una clase abstracta, obligando a las subclases a proporcionar una implementación para estos métodos.

## Interfaces

Diferencias y Similitudes entre Java y PHP:

* Sintaxis: La sintaxis para definir interfaces es similar en ambos lenguajes, utilizando la palabra clave **interface**.
* Métodos Abstractos: Tanto en Java como en PHP, los métodos en una interfaz son abstractos y no tienen cuerpo.
* Implementación en Clases: En ambos lenguajes, una clase debe implementar todos los métodos definidos en una interfaz que implementa.
* Herencia de Interfaz: En Java, una interfaz puede extender otras interfaces, lo que permite *herencia múltiple de interfaces*. En PHP, sin embargo, una clase puede implementar múltiples interfaces, pero las *interfaces no pueden heredar de otras interfaces*.

**PHP**:

Texto

Descripción generada automáticamente

**Java:**

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

## Polimorfismo

En PHP, el polimorfismo se puede lograr principalmente a través de interfaces, clases abstractas y herencia. A diferencia de Java, al momento de “redefinir” algún método en las clases hijas, solo necesitamos llamar el método con el mismo nombre que su superclase. En java es opcional utilizar la notación @Override.

### Polimorfismo mediante Interfaces

Una interfaz define un conjunto de métodos que deben ser implementados por cualquier clase que implemente la interfaz. Esto permite que diferentes clases implementen los mismos métodos de diferentes maneras.

### Polimorfismo mediante Clases Abstractas

Las clases abstractas pueden definir métodos abstractos (sin implementación) que deben ser implementados por las subclases. Esto también permite el polimorfismo, ya que las subclases proporcionarán sus propias implementaciones de los métodos abstractos.

# Conclusión

En esta investigación, comparamos la implementación de la herencia y otros conceptos básicos de programación entre los lenguajes de programación orientados a objetos, Java y PHP. A lo largo de nuestro análisis, hemos identificado varias similitudes y diferencias en la forma en que estos lenguajes abordan la herencia.

Tanto Java como PHP ofrecen soporte para la herencia simple, así como para otros conceptos relacionados como la herencia múltiple, herencia abstracta, interfaces y polimorfismo. Sin embargo, hay diferencias en la sintaxis y las convenciones de nomenclatura entre los dos lenguajes, así como en la forma en que se implementan estos conceptos.

Ambos lenguajes tienen sus fortalezas y son adecuados para diferentes tipos de proyectos. La elección de uno de estos lenguajes para implementar una cierta solución depende de los requerimientos generales del proyecto. Mientras que Java ofrece una plataforma sólida para aplicaciones empresariales complejas, PHP proporciona una herramienta flexible y rápida para el desarrollo web.

# Referencias

<https://docs.oracle.com/en/java/javase/17/>

<https://www.php.net/manual/es/>

Traits: <https://www.php.net/manual/en/language.oop5.traits.php>