## Universidad Autónoma de Ciudad Juárez

## División Multidisciplinaria Ciudad Universitaria

## 

## Cuadro comparativo de lenguajes de Programación Orientada a Objetos

## Programación II

## Docente: Alan Ponce

## 

## 

## Alumno: Jorge Brandon Garcia Bustamante 159799

## Licenciatura en Ingeniería de Software

## 08 de Marzo de 2019

# Introducción

El objetivo de este trabajo es hacer una sencilla comparación entre 2 diferentes paradigmas de programación, dando una definición de cada uno de los paradigmas y mostrar ejemplos para lograr también ver la diferencia en código.

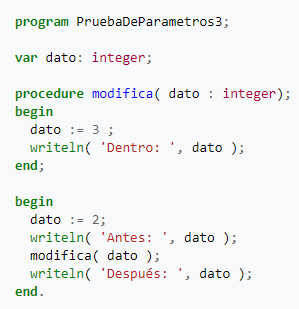
**Lenguaje procedural**

Este paradigma trabaja de una manera secuencial, utilizando una ruta predefinida con llamadas a procedimientos de ser necesarios, esto quiere decir que el programador deberá establecer todas la operaciones de acceso y llamados a operaciones.

**Pascal**

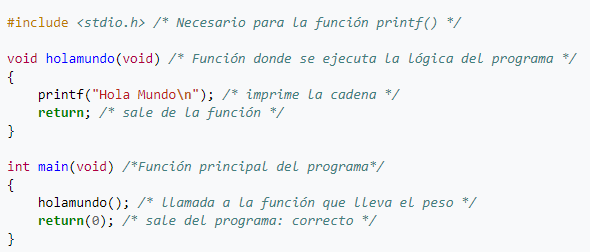
Pascal se caracteriza por ser un lenguaje de programación estructurado fuertemente tipado. Esto implica que:

* El código está dividido en porciones fácilmente legibles llamadas funciones o procedimientos. De esta forma Pascal facilita la utilización de la programación estructurada en oposición al antiguo estilo de programación monolítica.
* El tipo de dato de todas las variables debe ser declarado previamente para que su uso quede habilitado.



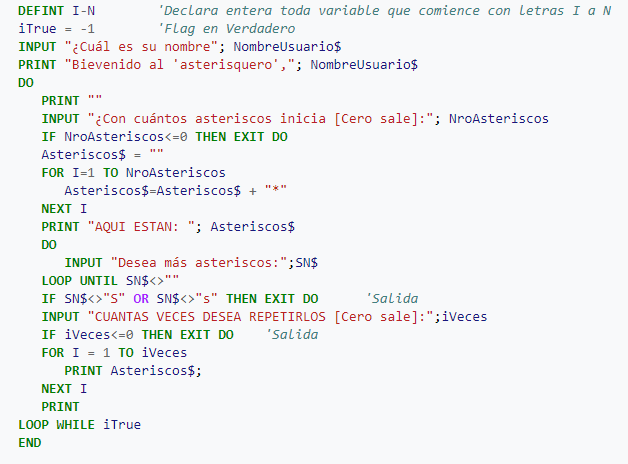
**C**

Se trata de un lenguaje de tipos de datos estáticos, débilmente tipificado, de medio nivel, ya que dispone de las estructuras típicas de los lenguajes de alto nivel, pero, a su vez, dispone de construcciones del lenguaje que permiten un control a muy bajo nivel. Los compiladores suelen ofrecer extensiones al lenguaje que posibilitan mezclar código en ensamblador con código C o acceder directamente a memoria o dispositivos periféricos.



**BASIC**

Los principios de BASIC son: que fuese fácil de usar por todos, crear un lenguaje de programación de propósito general, que se le pudiese incorporar características avanzadas y siguiese siendo de fácil uso para los principiantes, ser interactivo, que los mensajes de error fuesen claros, que respondiese rápidamente a los programas pequeños, que no fuese necesario tener conocimiento del hardware de la computadora, y que protegiese al usuario del sistema operativo.



**Lenguaje orientado a objetos**

Este paradigma busca innovar y facilitar la programación aprovechando las técnicas de herencia, cohesión, abstracción, polimorfismo, acoplamiento y encapsulamiento, así como los objetos que se pueden definir como un acercamiento a la realidad, con propiedades y acciones definidas por el usuario, y por ultimo clases, que permiten el uso de las técnicas anteriormente mencionadas y la agrupación de objetos.

**C++**

El motivo de la creación de C ++ fue expandir al lenguaje de programación C, utilizar mecanismos que permiten la manipulación de objetos, dicho eso, el C++ es un lenguaje híbrido.

El lenguaje de C++ puede ser un muy buen incio en la programación orientada a objetos si es que se viene de un lenguaje como C ya que permite una mejor adaptación gracias a su multiparadigma, esto da la posibilidad de trabajar como a lo que podrías estar acostumbrado e ir comenzando a experimentar con la nueva tool box que ofrece el paradigma orientado a objetos, además desde su existencia, la comunidad a trabajado lo suficiente con el como para proporcionar feedback para que se realicen mejoras en el lenguaje además de creaciones de API por parte de los usuarios.

**Java**

Fue diseñado especialmente para tener pocas dependencias de implementación como fuera posible. Su objetivo es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo, como su lema lo dice: “write once, run anywhere”, lo que quiere decir que el código que es ejecutado en una plataforma no tiene que ser recompilado para correr en otra.

Se podría decir que Java es uno de los lenguajes que más a evolucionado desde su lanzamiento, ya que a tenido bastantes actualizaciones a lo largo de su vida, y sigue siendo uno de los lenguajes de programación más utilizados por las compañías, tanto grandes como independientes, gracias a su previamente mencionada compatibilidad entre dispositivos, que da la posibilidad de no tener que reescribir códigos para cada uno, además a lo largo de cada una de sus actualizaciones se le dan más herramientas a los desarrolladores para la facilidad de desarrollo de aplicaciones, agregando también nuevas bibliotecas como cambios importantes a la API.

**Python**

Es uno de los lenguajes de programación dinámicos más populares que existen entre los que se encuentran Perl, Tcl, PHP y Ruby. Realmente es un lenguaje de propósito general. En la actualidad, Python es usado para todo, desde simples "scripts", hasta grandes servidores web. También es utilizado para la programación de interfaces gráficas y bases de datos.

Python está basado en varios lenguajes de programación para aprovechar sus funciones y características, también a lo largo de su vida a cambiado de compañía dueña del mismo lenguaje, cambiando así sus licencias de uso y licencias libres. Una de sus características más llamativas es que tiene múltiples versiones funcionando y siendo actualizadas al día de hoy, funcionando de manera distinta, cada una con sus ventajas y características que la hacen diferente a las demás, permitiendo así que se pueda utilizar como un lenguaje de propósito general.

**C#**

Es un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET, C# es uno de los lenguajes de programación diseñados para la infraestructura de lenguaje común.

Su sintaxis básica deriva de C/C++ y utiliza el modelo de objetos de la plataforma .NET, similar al de Java, aunque incluye mejoras derivadas de otros lenguajes.

Cuando C# fue lanzado era muy parecido a Java, incluso fue llamado una copia de Java, pero una versión muy temprana, pero con sus actualizaciones agregaron nuevas funciones y operadores que hacían del lenguaje una verdadera competencia directa con Java, pero a lo largo del tiempo decidieron mejor enfocarse en hacer un mejor lenguaje para los usuarios y algo más limpio en que programar en lugar de hacerle competencia directa a Java, haciendo que C# pudiera utilizarse de múltiples maneras, agregando en sus últimas actualizaciones bastantes herramientas para que a los programadores C# se vuelva una opción a elegir sin dudar.

**Cuadro comparativo**

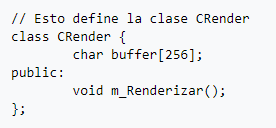
|  |  |  |
| --- | --- | --- |
| **Paradigma** | **Paradigma Procedural** | **Paradigma Orientada a Objetos** |
| **Características** | * Estado modificable. * Ejecución de sentencias. | * Abstracción * Encapsulación * Polimorfismo * Herencia |
| **Ventajas** | * Su relativa simplicidad y facilidad de implementación de compiladores. * Una forma fácil de seguir el código. * Necesita poca memoria. | * Los componentes se pueden reutilizar. * Se proporciona un buen marco que facilita la creación de interfaz gráfica de usuario para aplicaciones. * Se acopla bien a la utilización de bases de datos, debido a la correspondencia entre las estructuras. |
| **Desventajas** | * Los datos están expuestos, así que no tienen seguridad. * Es difícil crear nuevos tipos de datos, reduce la extensibilidad. | * No hay una forma única de resolver los problemas. Esto puede llevar a que diferentes interpretaciones de la solución planteada emerjan. * Se requiere una documentación amplia para determinar la solución planteada. |

# Lenguajes de Programación Orientada a Objetos

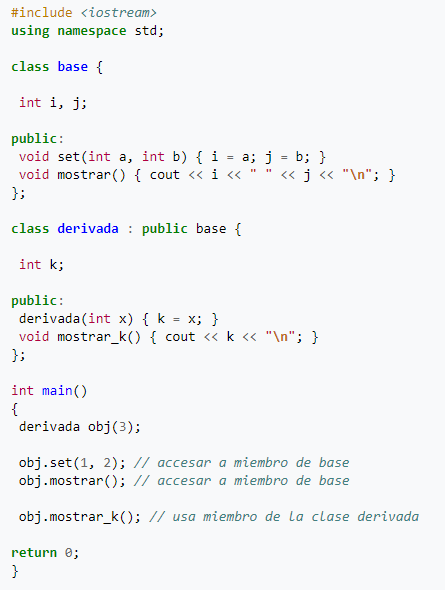
### **C++**

Incluir codigo de como implementar:

Una clase



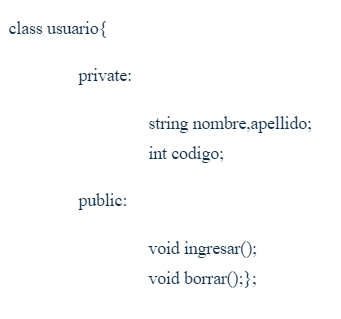
Herencia



Poliformismo



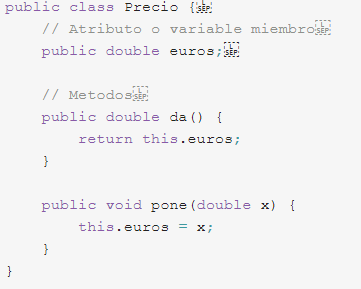
Encapsulamiento



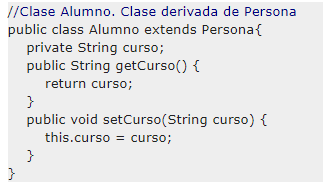
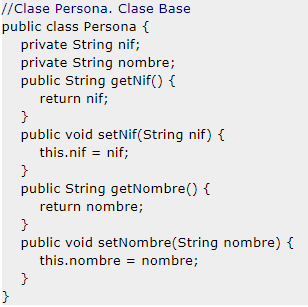
### **Java**

Incluir codigo de como implementar:

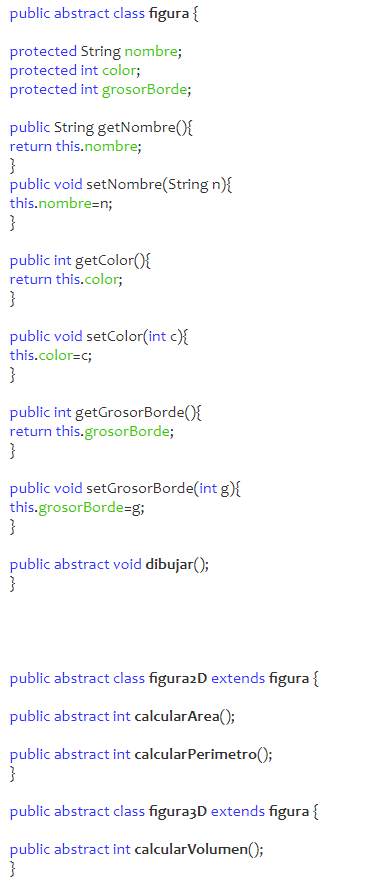
Una clase



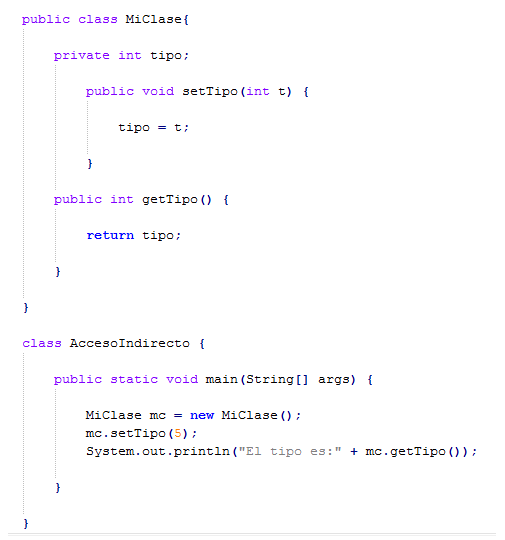
Herencia



Poliformismo



Encapsulamiento



### 

### 

### 

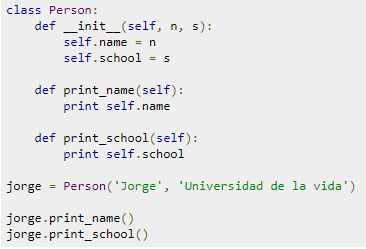
### 

### 

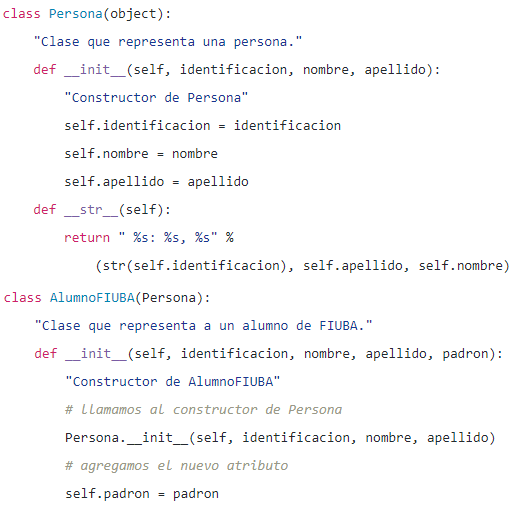
### **Python**

Incluir codigo de como implementar:

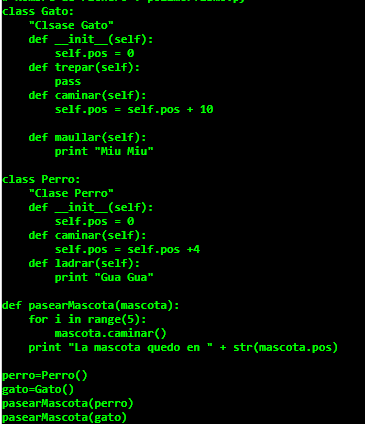
Una clase



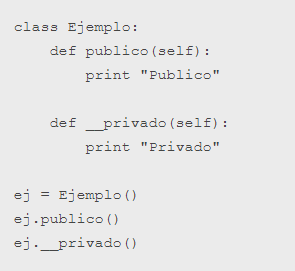
Herencia



Polifirmismo



Encapsulamiento

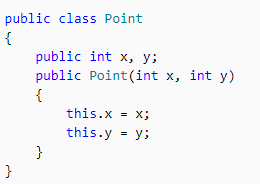


### 

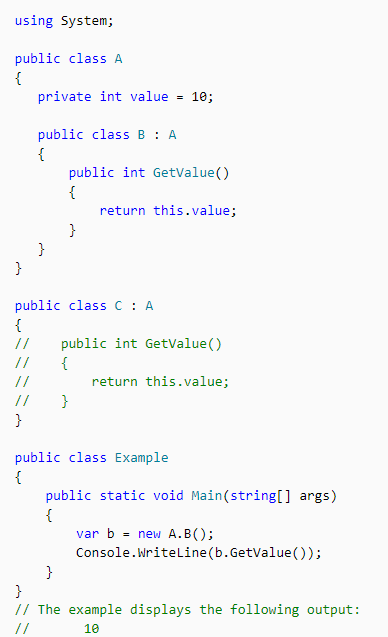
### **C#**

Incluir codigo de como implementar:

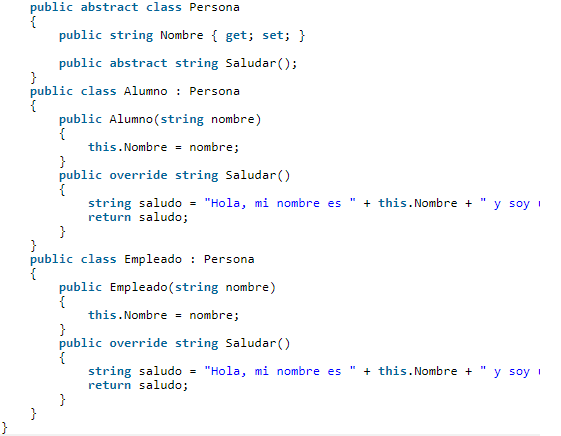
Una clase



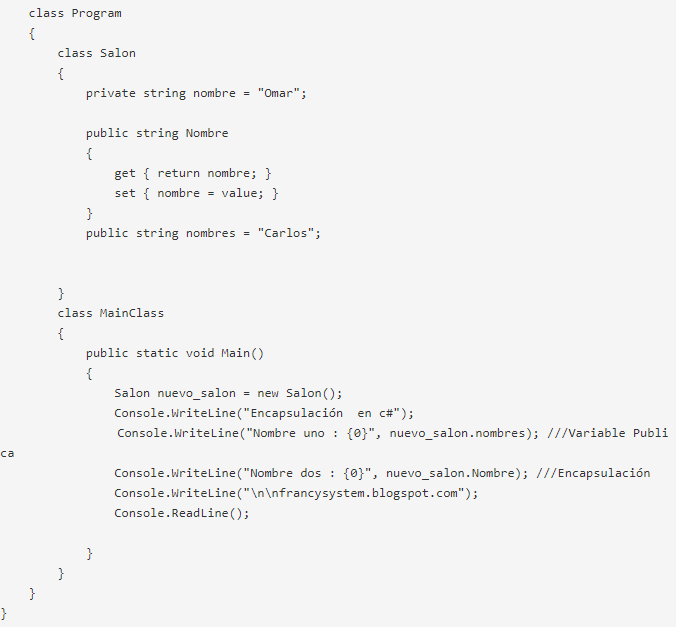
Herencia



Polifirmismo



Encapsulamiento



# Conclusiones

Gracias a este trabajo aprendí que hay lenguajes de programación para todo, desde simples funciones matemáticas, hasta trabajar con diseños de interfaces y bases de datos, personalmente me gustaría aprender de todo, ya que puede que en un futuro se me presenten diversas oportunidades y me gustaría poder estar preparado. Pienso que un lenguaje para comenzar a programar en OOP rápidamente es C++ por su simplicidad y la experiencia que tuve en C, pero me gusta estar trabajando en Java, ya que es muy utilizado hoy en día y además es parte de lo que quiero hacer, extenderme en varios lenguajes, y creo que puedo decir que no hay un mejor lenguaje de programación, tal vez lo haya más simples, pero no mejores ya que como dije cada uno tiene sus puntos fuertes, quiero seguir aprendiendo el paradigma de OOP y realizar proyectos.

**Bibliografía**

jaimemontoya. ( 2017). Tipos de Lenguajes de Programación. 07/03/2019, de monografias Sitio web: <https://www.monografias.com/trabajos38/tipos-lenguajes-programacion/tipos-lenguajes-programacion2.shtml>

ecured. (2016). Historia del Lenguaje C. 07/03/2019, de ecured Sitio web: <https://www.ecured.cu/Historia_del_Lenguaje_C>

‎ Edgar. (2002). C++. 07/03/2019, de wikipedia Sitio web: <https://es.wikipedia.org/wiki/C%2B%2B>

Cecilio Álvarez Caules . (2016). Las versiones de Java y su historia. 08/032019, de arquitecturajava Sitio web: <https://www.arquitecturajava.com/las-versiones-de-java/>

unmalnick. (2018). La historia de Python. 08/03/2019, de platzi Sitio web: <https://platzi.com/blog/historia-python/>

campusMVP. (2017). Historia del lenguaje C#: pasado, presente y evolución. 08/03/2019, de campusmvp Sitio web: <https://www.campusmvp.es/recursos/post/historia-del-lenguaje-c-sharp-pasado-presente-y-evolucion.aspx>

ManuelGR. (2004). Polimorfismo . 08/03/2019, de wikipedia Sitio web: <https://es.wikipedia.org/wiki/Polimorfismo_(inform%C3%A1tica)>

Enrique. (2014). Herencia en Java.. 08/03/2019, de puntocomnoesunlenguaje Sitio web: <http://puntocomnoesunlenguaje.blogspot.com/2014/07/herencia-en-java.html>

Ángel Roldán. (2017). Encapsulamiento en Java. 08/03/2019, de ciberaula Sitio web: <http://www.ciberaula.com/articulo/encapsulacion_java>

programacion. (2017). Cómo funcionan las clases y objetos en Python. 08/03/2019, de programacion Sitio web: <https://programacion.net/articulo/como_funcionan_las_clases_y_objetos_en_python_1505>