

1. Conceptos básicos de POO¹

Es muy importante conocer bien nuestro paradigma de programación y además el lenguaje de programación, ya que hay lenguajes y paradigmas que se adaptan mejor a ciertas tareas que otros.

Hay incluso ocasiones en las que para un mismo sistema, tengas que utilizar dos lenguajes de programación con distintos paradigmas. Lo importante es tener la sensibilidad para saber cuándo utilizar la POO es lo mejor para un programa. Comencemos.

Parte fundamental del desarrollo de *software* es independiente del lenguaje de programación, el paradigma o la tecnología seleccionada. Esta parte es la del análisis y diseño, es la etapa donde se estudia y analiza el problema a resolver, una vez hecho esto es cuando se pueden seleccionar las herramientas. El análisis y diseño no es sencillo, ya que se requiere de mucha experiencia y conocimiento profundo del problema para tomar la elección.

En la etapa de diseño, antes de comenzar con la programación, es muy importante delimitar qué partes del problema real deben ser pasados al plano virtual. Imagina que vas a escribir un programa para llevar la asistencia de los alumnos en una clase, necesitamos tener información de alumnos, pero sólo la necesaria para lo que necesitamos, es decir: nombre, número de lista, materias inscritas, etc. En este caso, no necesitamos otro detalle como su género, edad, nombre de mascota o fecha de cumpleaños. A esto se llama abstracción.

La ventaja de la POO es su capacidad para modelar situaciones reales a programas de computadora, ya que así como existe un *caballo* en la realidad, podemos abstraer el concepto y pasarlo a un programa.

Los beneficios que se buscan con la POO son:

- **Reutilización de código:** esto es, poder escribir programas (o partes de uno) para resolver un problema, que después se puedan usar para resolver otros diferentes.
- **Modularidad:** el poder agregar o quitar componentes a nuestros programas de una forma sencilla y clara.

Una industria que utiliza mucho la POO es la de los videojuegos, ya que por el tipo de aplicación que es se adapta muy bien el uso de lenguajes como C++, JAVA o C#.

¹ Documento elaborado por el Ing. Arturo Nereu Núñez.

2 ¿Qué es la programación orientada a objetos?

La Programación Orientada a Objetos (POO) como se define formalmente en el libro de texto:

Es un método de implementación en el que los programas se organizan como colecciones cooperativas de objetos, cada uno de los cuales representa una instancia de una clase, y cuyas clases son, todas ellas, miembros de una jerarquía de clases unidas mediante relaciones de herencia. [1]

Los cuatro principios de la POO

Ya vimos que las metas de la POO son lograr código reutilizable y modular, ahora vamos a exponer con qué principios vamos a apoyarnos para alcanzar las metas [2]:

- **Abstracción:** es descomponer sistemas complejos en componentes más sencillos.
- **Encapsulación:** se refiere a dar a los usuarios una forma de interactuar con objetos mediante métodos que oculten su funcionamiento interno.
- **Herencia:** es la habilidad de derivar nuevas clases de otras ya existentes. Una clase derivada hereda los componentes y funciones de una clase base y puede agregar nueva funcionalidad.
- **Polimorfismo:** la habilidad de definir múltiples clases con funcionalidad diferente pero nombres de funciones y propiedades idénticos que pueden ser utilizadas de forma intercambiable.

3. Características de una aplicación básica de programación orientada a objetos

La principal característica de un programa orientado a objetos es que se cuenta con una estructura de relación entre objetos, por ejemplo, si definimos una estructura de animales donde sabemos que:

Un perro **es un** (is a) animal.

Un gato **es un** (is a) animal.

Por otro lado, si tenemos una veterinaria modelada, entonces:

Una veterinaria **tiene un** (has a) dog.

Una veterinaria **tiene un** (has a) cat.

Aunque no es objetivo del curso aprender sobre diagramas UML, el revisar la siguiente presentación te

ayudará en tu diseño de programas orientados a objetos [3]: <http://www.slideshare.net/e1da4/diagramas-uml>

4. Ejecución de una aplicación en programación orientada a objetos

El lenguaje de programación Java es un lenguaje basado en clases, orientado a objetos y diseñado específicamente para tener la menor cantidad de dependencias de implementación posibles. Esto significa que un programa hecho en Java puede correr en la JVM (máquina virtual de Java) en cualquier sistema operativo como puede ser Windows, Mac o Linux, sin necesidad de hacer cambios al código.

Incluso, hay versiones de Java para teléfonos móviles que siguen esa filosofía.

El lenguaje Java cuenta con sintaxis muy parecida a la de lenguajes como C y C++ y es ampliamente utilizado en diferentes áreas del desarrollo de *software*, siendo la más popular la del desarrollo de aplicaciones cliente servidor para web.

El componente que le permite a Java no tener dependencias de plataformas es lo que se conoce como la máquina virtual de Java. La JVM es una máquina virtual, que también puede ser visto como un programa especial que puede ejecutar el bytecode de Java. Este bytecode es el resultado de compilar código desarrollado con Java.

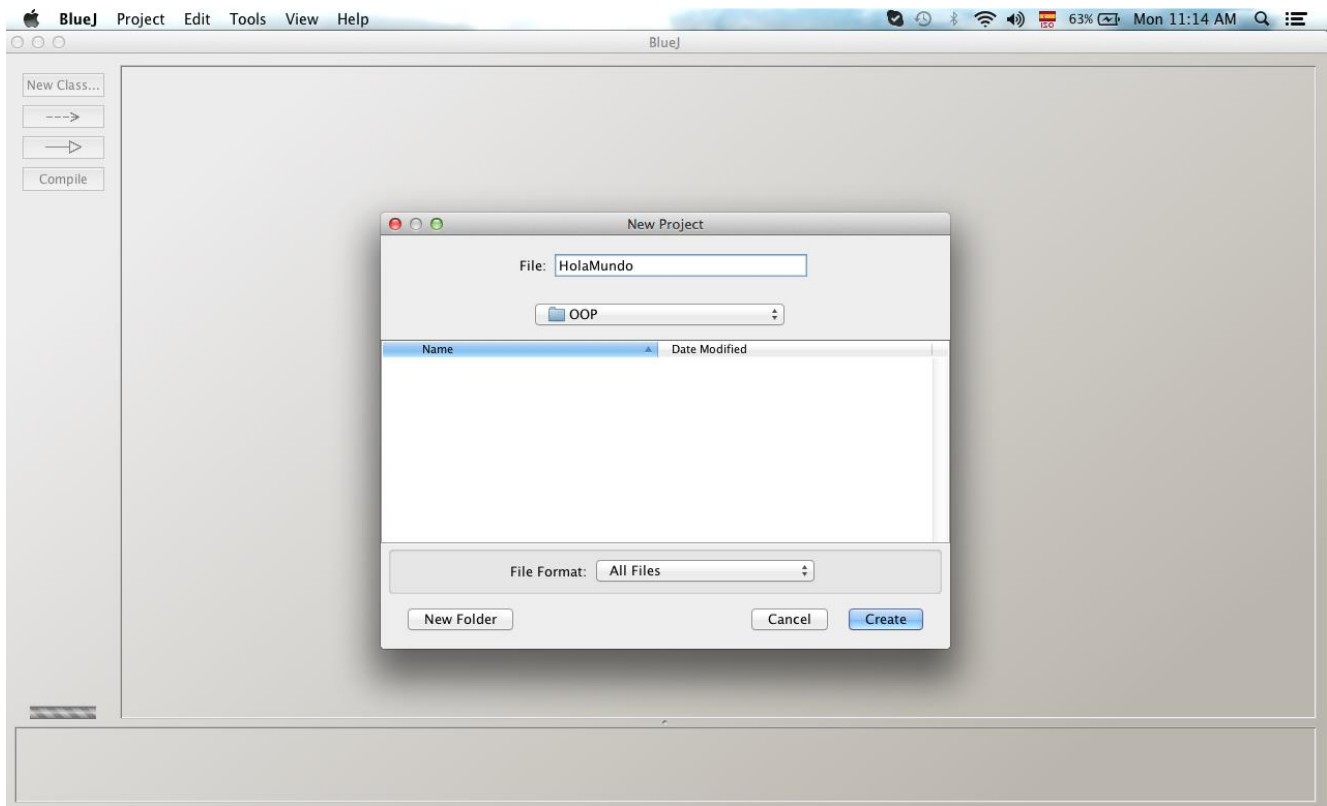
En Java, como en otros lenguajes de programación, existe un manejador de memoria llamado Garbage Collector (recolector de basura) que es el componente que se encarga de manejar la memoria de forma automática a diferencia de lenguajes como C++ donde el programador debe encargarse de solicitar y liberar la memoria del dispositivo en el que está corriendo el programa.

Para poner manos a la obra, vamos a instalar nuestro IDE para editar y ejecutar código: BlueJ, para eso apóyate del siguiente video: http://youtu.be/21nbc_Y7_t4

La descarga directa la puedes hacer en: www.bluej.org

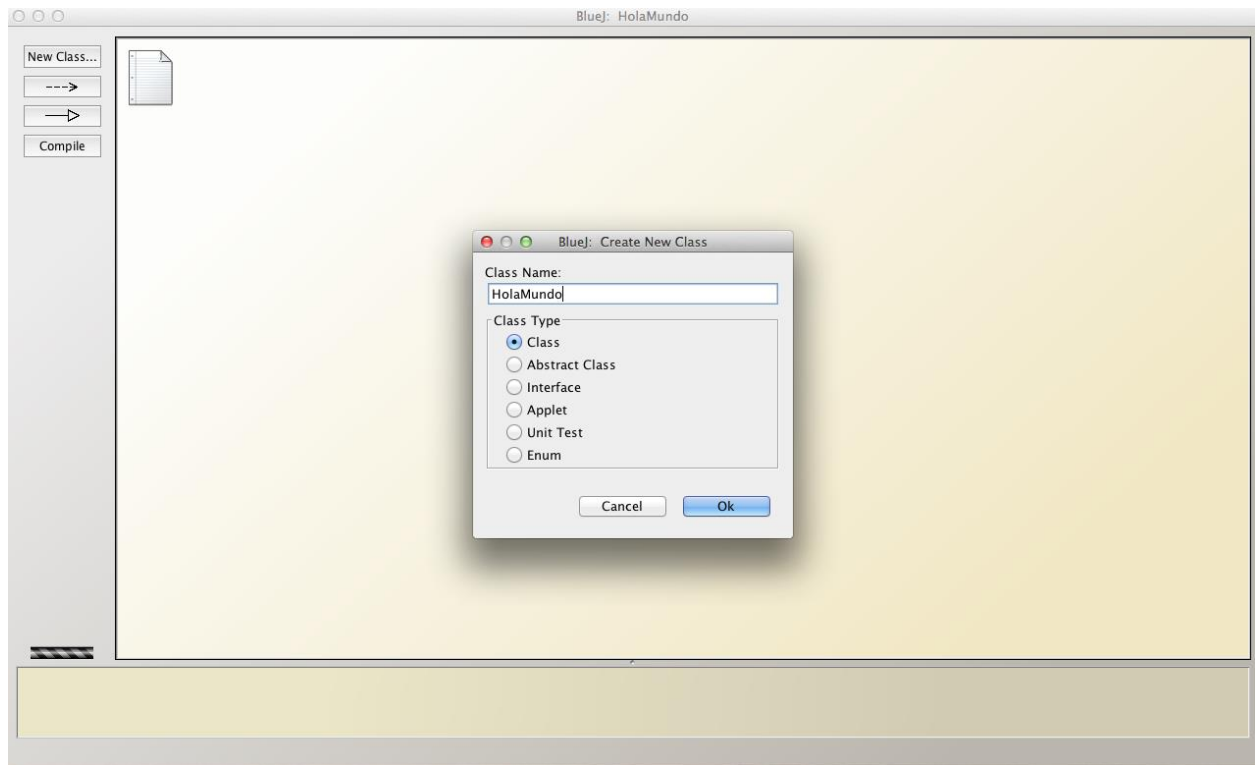
Una vez instalado, vamos a crear nuestro primer programa:

- 1) Abre BlueJ.
- 2) Crea un nuevo proyecto: Project -> New Project
- 3) Como nombre, ponle: HolaMundo

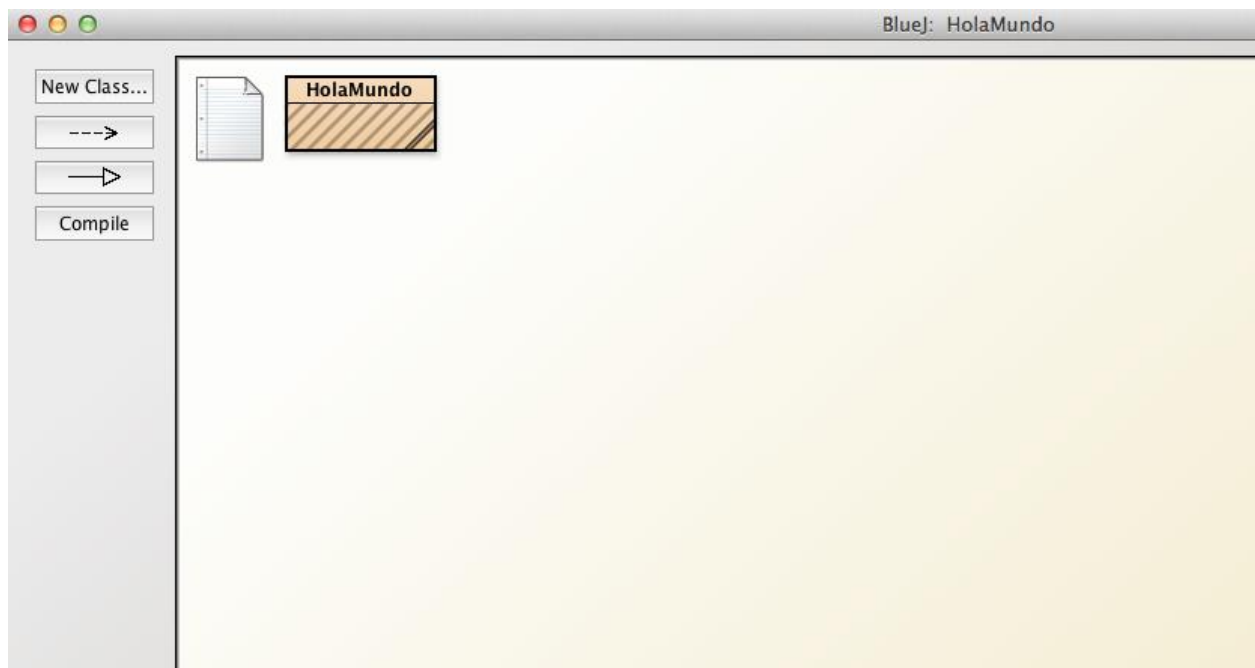


4) Vamos a crear una nueva clase (después veremos más a detalle qué es esto), por ahora simplemente lo veremos como un archivo .java.

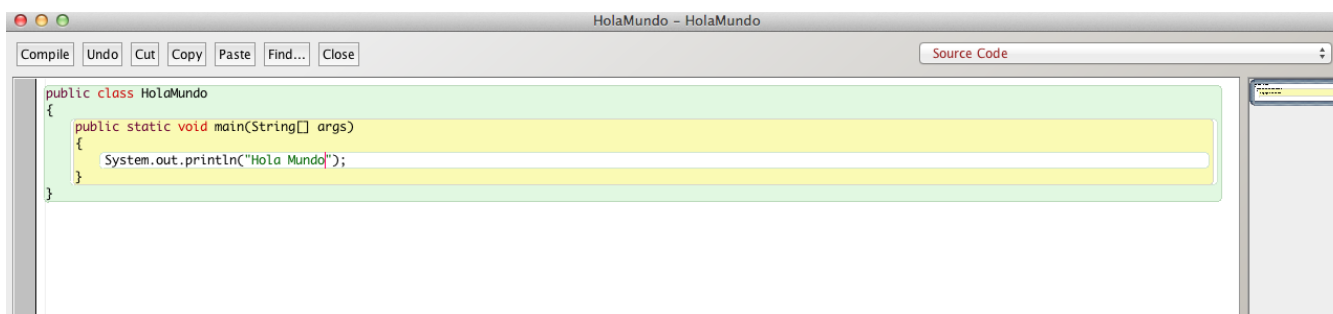
Da click en New Class... y ponle de nombre al archivo: HolaMundo



5) Ahora, el nuevo archivo HolaMundo está creado, dale doble click y vamos a abrirlo para editar.

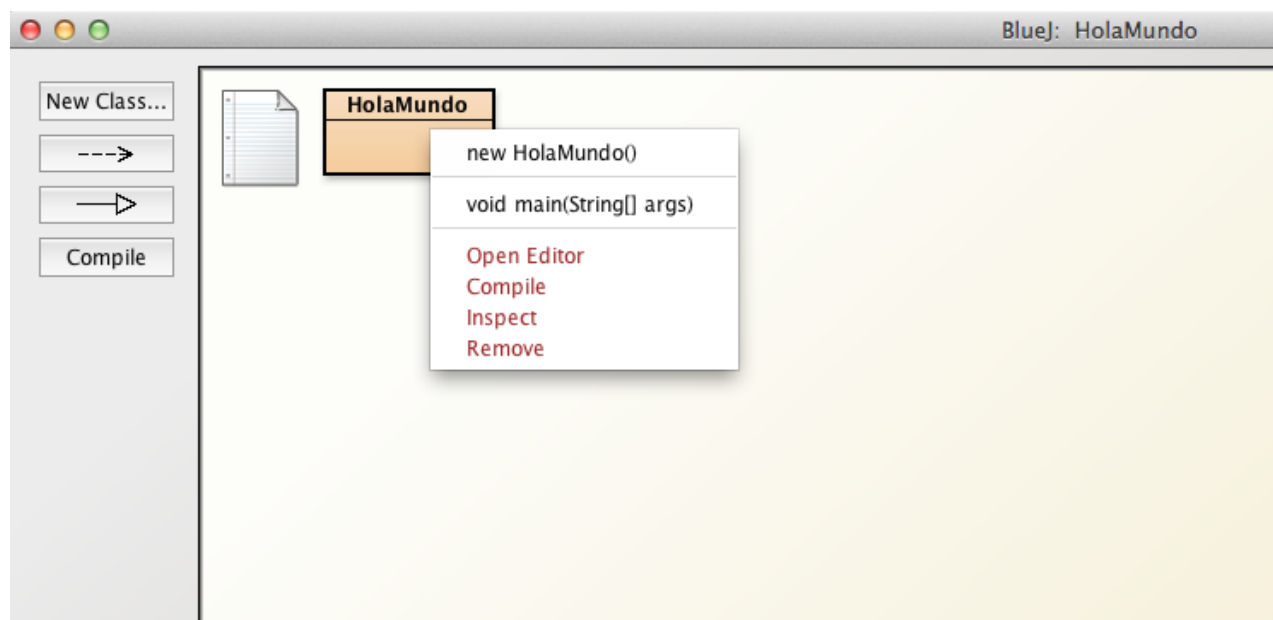


6) Una vez abierto, vamos a editar el código, borrando su contenido y dejando solamente lo siguiente:



7) Es momento de compilarlo, da click al botón *Compile* y en la parte de abajo de BlueJ no debe haber ningún error, si lo hay revisa que hayas escrito el código de la misma manera.

8) Cierra el archivo (Close) y ahora, da click en el escritorio de BlueJ para ejecutarlo con botón derecho:



8) Da click izquierdo en la opción **void main(String[] args)**. En la siguiente ventana sólo dale OK y ahora aparecerá una consola con el mensaje “Hola Mundo”.

Listo, ya hiciste tu primer programa orientado a objetos y tal vez sin que sea tan obvio. En el resto del curso vamos a ver cómo sacar el mayor provecho de Java en el uso de la POO.

Referencias

[1] Orientación a objetos. Teoría y Práctica (en biblioteca virtual).

[2] Principios de la Programación Orientada a Objetos (recurso en línea).

https://www.google.com.mx/search?q=oop+principles&og=oop&aqs=chrome.0.69i59j69i57j0j69i59j0l2.1239j0j7&sourceid=chrome&espv=2&es_sm=91&ie=UTF-8

[3] Diagramas UML (recurso en línea) <http://www.slideshare.net/e1da4/diagramas-uml>