

• Guía número uno.

Parte 1.

- **¿Qué es la programación orientada a objetos (POO)?**

La **programación orientada a objetos** (**POO**, u **OOP** según sus siglas en inglés) es un paradigma de programación que usa objetos en sus interacciones, para diseñar aplicaciones y programas informáticos. Está basada en varias técnicas, incluyendo herencia, cohesión, abstracción, polimorfismo, acoplamiento y encapsulamiento.

La programación orientada a Objetos básicamente define una serie de conceptos y técnicas de programación para representar acciones o cosas de la vida real basada en objetos.

- **¿Cuáles son los conceptos y nociones básicas de la POO?**

- **Clases.**

Las clases son uno de los principales componentes de un lenguaje de programación, pues en ellas ocurren todos los procesos lógicos requeridos para un sistema, en si podemos definirlas como estructuras que representan objetos del mundo real, tomando como objetos a personas, lugares o cosas, en general las clases poseen propiedades, comportamientos y relaciones con otras clases del sistema.

Una clase se compone por tres partes fundamentales:

Nombre: Contiene el Nombre de la Clase.

Atributos: Representan las propiedades que caracterizan la clase.

Métodos: Representan el comportamiento u operaciones, la forma como interactúa la clase con su entorno.

- **Objeto.**

Los objetos representan una entidad concreta o abstracta del mundo real, en programación básicamente se le conoce como la instancia de una clase en si es lo que da el sentido a estas.

Al igual que las clases se componen de tres partes fundamentales:

Estado: Representa los atributos o características con valores concretos del objeto.

Comportamiento: Se define por los métodos u operaciones que se pueden realizar con él.

Identidad: Es la propiedad única que representa al objeto y lo diferencia del resto.

En java se representa creando una instancia de la clase por medio de la palabra **new** al hacer eso creamos el objeto de la clase y podemos hacer uso de los métodos o atributos de esta (dependiendo de la visibilidad de los mismos) por medio de un punto (.)

- **Explique de manera clara que es:**

- **Polimorfismo.**

Este tal vez sea uno de los conceptos de la programación orientada a objetos más usados pero muchas veces sin saber que se aplica ya que el concepto inicialmente puede ser un poco confuso, básicamente mediante el polimorfismo programamos de forma general en lugar de hacerlo de forma específica, se usa cuando se trabajan con la herencia y objetos de características comunes los cuales comparten la misma superClase y árbol jerárquico, al trabajar con este concepto optimizamos y simplificamos en gran medida nuestro trabajo.

Básicamente podemos definirlo como la capacidad que tienen los objetos de comportarse de múltiples formas sin olvidar que para esto se requiere de la herencia, en si consiste en hacer referencia a objetos de una clase que puedan tomar comportamientos de objetos descendientes de esta.

- **Encapsulamiento.**

Este concepto es uno de los más importantes en términos de seguridad dentro de nuestra aplicación, la encapsulación es la forma de proteger nuestros datos dentro del sistema, estableciendo básicamente los permisos o niveles de visibilidad o acceso de nuestros datos.

Se representa por 3 niveles:

Público: Se puede acceder a todos los atributos o métodos de la clase. **Protegido:** Se puede acceder a los atributos o métodos solo en la misma jerarquía de herencia.

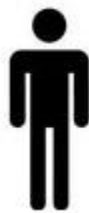
Privado: Solo se puede acceder a los atributos o métodos de la clase en la que se encuentran.

- **HERENCIA**

Un tipo de objeto de alto nivel puede especializarse en tipos de objeto de bajo nivel. Un tipo de objeto puede tener subtipos. Por ejemplo, el tipo de objeto persona puede tener subtipos estudiante y empleado. A su vez, el tipo de objeto estudiante puede tener como subtipo estudiante de pregrado y estudiante de postgrado, mientras que empleado puede tener como subtipo a académico y administrativo. Existe de este modo una jerarquía de tipos, subtipos, subsubtipos, etc.

- **Abstracción**

La abstracción permite resaltar la parte más representativa de algo, ignorando detalles para centrarse en lo principal.



La imagen es muy fácil de identificar, con base a ella podemos crear una clase persona, o la clase hombre, humano entre otras, pero obviamente vemos que la imagen no tiene elementos como ojos, nariz, boca, rostro en general, ni dedos, pies, manos o cuello..... pero entonces porque decimos que es una persona?.....Precisamente aquí estamos aplicando el concepto de abstracción, ya que nos fijamos en lo más representativo de algo, en este caso vemos que se tiene una cabeza, tronco, brazos y pies, con esto es suficiente para saber que es una persona sin fijarnos en los detalles mencionados anteriormente.

- **¿Por qué es preferible usar POO a programación estructurada?**

Las ventajas de un lenguaje orientado a objetos, son:

- Fomenta la reutilización y extensión del código.
- Permite crear sistemas más complejos.
- Relacionar el sistema al mundo real.
- Facilita la creación de programas visuales.
- Construcción de prototipos
- Agiliza el desarrollo de software
- Facilita el trabajo en equipo
- Facilita el mantenimiento del software.

Con la programación estructurada elaborar programas de computador sigue siendo un albor que demanda esfuerzo, creatividad, habilidad y cuidado.

- **¿Qué ventajas posee al usar JAVA para el desarrollo de aplicaciones?**
- **Lenguaje Simple.** Una de las cosas más importantes que debes saber de Java, es que no es para nada complejo. De hecho la curva de aprendizaje del lenguaje es realmente corta, por lo que de inmediato podrás familiarizarte con los términos y las funciones que el lenguaje utiliza.
- Eso mismo ocurre con las personas que ya tienen conocimientos en otros lenguajes de programación como C o C++. Pues la realidad es que estos lenguajes son muy similares, obviamente no son iguales, pero si ya tienes la lógica, que es una de las cosas más importantes, entonces no te costará ningún trabajo empezar a trabajar con Java y sobretodo sacarle provecho, que es una de las cosas que se buscan, se productivo y que valga la pena el tiempo invertido.
- **Lenguaje Orientado a Objetos.** Si no conoces mucho de programación, entonces te estarás preguntando, cómo a objetos. Bueno, pues los objetos se encargan de encapsular información, clases y funciones, las cuales se pueden manipular más adelante o se pueden agregar a distintos programas y existe lo que es la manipulación de datos entre objetos, por algo este tipo de lenguajes son mucho más potentes.
- No por nada, Java es uno de los lenguajes más utilizados en proyectos de gran tamaño, que a simple vista pueden parecer sumamente complejos.
- **Aplicaciones Distribuidas.** Seguramente has escuchado hablar del cómputo distribuido, lo que es y las ventajas de este, pues te cuento, con Java tienes la posibilidad de hacer aplicaciones distribuidas. Estas aplicaciones en red, lo que hacen son ejecutarse en una plataforma que se compone por una base de cómputo distribuido y funcionar perfectamente. Mantiene mucha estabilidad y el rendimiento se incrementa considerablemente, bueno, es solo un detalle por si no lo sabías.
- **Interpretado y Compilado.** Una de las principales ventajas de Java, definitivamente es su compilación. ¿Qué ocurre con la compilación de Java?. Bueno, pues la compilación es tan buena, que se llega a asimilar al lenguaje ensamblador, es decir, desde la base puede ser interpretado. Esto ayuda muchísimo a la ejecución de aplicaciones compiladas en Java, pues se puede ejecutar básicamente en cualquier lugar sin mayor problema.
- **Es Seguro.** Una de las virtudes de Java, posiblemente sea su seguridad, además de que es un lenguaje a código abierto, pero sus programas están compilados tan perfecta y originalmente, que no tendrás ningún problema con filtros de seguridad ni cosas por el estilo. Tendrás la comodidad de que incluso al hacer aplicaciones web con Java, la seguridad será máxima y no tendrás por qué inquietarte.
- **Desglose cada uno de los siguientes conceptos en el ámbito de POO:**

Clase: Cada **clase** es un modelo que define un conjunto de variables -el estado, y métodos apropiados para operar con dichos datos -el

comportamiento. Cada objeto creado a partir de la **clase** se denomina instancia de la **clase**. Las **clases** son un pilar fundamental de la **programación orientada a objetos**.

Método: Es la implementación de un algoritmo que representa una operación o función que un objeto realiza. El conjunto de los **métodos** de un objeto determinan el comportamiento del objeto.

Constructor: es una sub rutina cuya misión es inicializar un objeto de una clase

Librería: conjunto de clases que poseen una serie de métodos y atributos, facilitando las operaciones

Registro: es una variable especial que tiene la capacidad de almacenar datos de diferentes tipos

Datos: representan información que queremos recordar, comparar o manipular

- ¿Cuáles son las características principales de los siguientes tipos de datos?
 - Int:32 bits, se utiliza para números enteros
 - Double:64 bits, se utiliza para los números decimales
 - String: Es para una cadena de texto
 - Char:16 bits. Unicode
 - Object:variable también puede hacer referencia a los **datos** de cualquier **tipo** de valor (numérico, Boolean , Char , Date , estructura o enumeración)
- ¿Que es la clase math en JAVA? Representa la librería matemática de java
- ¿Qué es una base de datos? son un almacén que nos permiten guardar grandes cantidades de información de forma organizada para que luego podamos encontrarla y utilizarla fácilmente

- **¿Qué es un vector dentro java?**
Posee la capacidad de definir un conjunto de variables del mismo tipo agrupadas todas ellas bajo un mismo nombre, y distinguiéndolas mediante un índice numérico

- **¿Qué es un array dentro de java?**
Un **array** es una estructura de datos que nos permite almacenar una gran cantidad de datos de un mismo tipo. El tamaño de los **arrays** se declara en un primer momento y no puede cambiar en tiempo de ejecución como puede producirse en otros lenguajes.

¿Cómo se diferencia un vector de un array?

el array es estático y el vector es dinámico quiere decir a la hora de declarar el array tienes que darle un valor.

-el array solo almacena un tipo de dato y el vector puede almacenar varios tipos de datos

-el array no tiene métodos ni clases

-al declarar el vector no se le tiene que dar un valor

-el vector tiene métodos

-el vector es más pesado porque contiene métodos, clases

-cada vez que ingresas un valor en el vector tienes que darle un id.

- ¿Qué utilidad concreta tiene cada una de estas estructuras de datos dentro de la programación?

Estas estructuras de datos son adecuadas para situaciones en las que el acceso a los datos se realice de forma aleatoria e impredecible. Por el contrario, si los elementos pueden estar ordenados y se va a utilizar acceso secuencial sería más adecuado utilizar una [lista](#), ya que esta estructura puede cambiar de tamaño fácilmente durante la ejecución de un programa.

- ¿Qué es el tipo de datos object en JAVA?

Todas las clases son en realidad subclases de una clase más amplia: la clase Object. Esta clase incluye todos los objetos (los lectores de archivos, las tortuga, los arreglos, los glyphs, etc.). Por lo tanto siempre es posible colocar cualquier objeto en donde se espera una expresión de tipo Object.

- **¿Qué es una matriz y q tipos de datos puede manejar?**

Un array en Java puede tener más de una dimensión. El caso más general son los arrays bidimensionales también llamados **matrices** o **tablas**.

La dimensión de un array la determina el número de índices necesarios para acceder a sus elementos.

Los vectores que hemos visto en otra entrada anterior son arrays unidimensionales porque solo utilizan un índice para acceder a cada elemento.

Una matriz necesita dos índices para acceder a sus elementos.

