

# RESUMEN TEMA 4 JAVA:

## 1.¿Que y como es java?

Java es un lenguaje sencillo parecido a C++ creado para solucionar problemas que otros programas de la época no solucionaban en el que se han eliminado elementos complicados y permite utilizar bibliotecas ya creadas evitando reescribir código existente.

Java es independiente del hardware ya que este pasa el código a una maquina virtual que luego traducirá a código específico.

C++ era el lenguaje mas usado hasta la llegada de Java pero cuando los programas en C aumentaban su volumen su manejo se complicaba hasta la llegada de la Programación Orientada a Objetos haciendo mas sencillo todo el proceso.

## 1.2.Breve historia

Java surgió en 1991 cuando un grupo de ingenieros trataron de diseñar un nuevo lenguaje de programación destinado a pequeños dispositivos electrónicos.

En 1995 se dio a conocer como Java y se presento al publico como un lenguaje para computadores y paso a ser un lenguaje independiente y potente.

## 1.3-La POO y Java

En Java, los datos y el código (funciones o métodos) se combinan en entidades llamadas objetos. El objeto tendrá un comportamiento (su código interno) y un estado (los datos). Los objetos permiten la reutilización del código y pueden considerarse, en sí mismos, como piezas reutilizables en múltiples proyectos distintos. Esta característica permite reducir el tiempo de desarrollo de software.

Al emplear los conceptos de la Programación Orientada a Objetos (POO), Java incorpora las tres características propias de este paradigma: encapsulación (En programación modular y más específicamente en programación orientada a objetos, se denomina así al ocultamiento de los datos y elementos internos de un objeto. Sólo se puede modificar un objeto a través de las operaciones definidas para éste.), herencia(Mecanismo que permite derivar una clase de otra, de manera que extienda su funcionalidad) y polimorfismo.(Capacidad para que varias clases derivadas de una antecesora utilicen un mismo método de forma diferente. Podemos crear dos clases distintas: Pez y Ave que heredan de la superclase Animal. La clase Animal tiene el método abstracto mover que se implementa de forma distinta en cada una de las subclases (peces y aves se mueven de forma distinta).) Los patrones o tipos de objetos se denominan clases(Es una construcción que se utiliza como un modelo (o plantilla) para crear objetos de ese tipo. El modelo describe el estado y el comportamiento que todos los objetos de la clase comparten) y los objetos que utilizan estos patrones o pertenecen a dichos tipos, se identifican con el nombre de instancias.

## 1.4-Independencia de la plataforma y trabajo en red

Independencia: Los programas escritos en Java pueden ser ejecutados en cualquier tipo de hardware. El código fuente es compilado, generándose el código conocido como Java Bytecode

Trabajo en red: Esta capacidad del lenguaje ofrece múltiples posibilidades para la comunicación vía TCP/IP. Para poder hacerlo existen librerías que permiten el acceso y la interacción con protocolos como http, ftp, etc., facilitando al programador las tareas del tratamiento de la información a través de redes.

### 1.5-Java y los Bytecodes

Un programa escrito en Java no es directamente ejecutable, es necesario que el código fuente sea interpretado por la Máquina Virtual Java. ¿Cuáles son los pasos que se siguen desde que se genera el

código fuente hasta que se ejecuta? A continuación, se detallan cada uno de ellos.

Una vez escrito el código fuente (archivos con extensión .Java), éste es precompilado generándose los códigos de bytes, Bytecodes o Java Bytecodes que serán interpretados directamente por la Máquina Virtual Java y traducidos a código nativo de la plataforma sobre la que se esté ejecutando el programa

### 2.-Programas en Java

Public class Clase\_Principal: Todos los programas han de incluir una clase como esta. Es una clase general en la que se incluyen todos los demás elementos del programa. Entre otras cosas, contiene el método o función main() que representa al programa principal, desde el que se llevará a cabo la ejecución del programa. Esta clase puede contener a su vez otras clases del usuario, pero sólo una puede ser public. El nombre del fichero .Java que contiene el código fuente de nuestro programa, coincidirá con el nombre de la clase que estamos describiendo en estas líneas.

#### 2.2- El entorno básico de desarrollo Java.

existen diferentes entornos para la creación de programas en Java que incluyen multitud de herramientas, pero por ahora nos centraremos en el entorno más básico, extendido y gratuito, el Java Development Kit (JDK). Según se indica en la propia página web de Oracle, JDK es un entorno de desarrollo para construir aplicaciones, applets y componentes utilizando el lenguaje de programación Java. Incluye herramientas útiles para el desarrollo y prueba de programas escritos en Java y ejecutados en la Plataforma Java.

#### 2.3-La API de Java

Junto con el kit de desarrollo que hemos descargado e instalado anteriormente, vienen incluidas gratuitamente todas las bibliotecas de la API (Application Programming Interface – Interfaz de programación de aplicaciones) de Java, es lo que se conoce como Bibliotecas de Clases Java. Este conjunto de bibliotecas proporciona al programador paquetes de clases útiles para la realización de múltiples tareas dentro de un programa. Está organizada en paquetes lógicos, donde cada paquete contiene un conjunto de clases relacionadas semánticamente (Referencia a los aspectos del significado, sentido o interpretación del significado de un determinado elemento, símbolo, palabra, expresión o representación formal. En principio cualquier medio de expresión (lenguaje formal o natural) admite una correspondencia entre expresiones de símbolos o palabras, y situaciones o conjuntos de cosas que se encuentran en el mundo físico o abstracto, que puede ser descrito por dicho medio de expresión).

#### 2.4- Codificación, compilación y ejecución de aplicaciones.

Una vez que la configuración del entorno Java está completada y tenemos el código fuente de

nuestro programa escrito en un archivo con extensión .Java, la compilación de aplicaciones se realiza mediante el programa Javac incluido en el software de desarrollo de Java.

Para llevar a cabo la compilación desde la línea de comandos, escribiremos:

javacarchivo.java Donde javac es el compilador de Java y archivo.java es nuestro código fuente. El resultado de la compilación será un archivo con el mismo nombre que el archivo Java pero con la extensión class. Esto ya es el archivo con el código en forma de bytecode. Es decir con el código precompilado. Si en el código fuente de nuestro programa figuraran más de una clase, veremos como al realizar la compilación se generarán tantos archivos con extensión .class como clases tengamos. Además, si estas clases tenían método main podremos ejecutar dichos archivos por separado para ver el funcionamiento de dichas clases. Para que el programa pueda ser ejecutado, siempre y cuando esté incluido en su interior el método main, podremos utilizar el intérprete incluido en el kit de desarrollo. La ejecución de nuestro programa desde la línea de comandos podremos hacerla escribiendo: javaarchivo.class Donde java es el intérprete y archivo.class es el archivo con el código precompilado.

### 2.5- Tipos de aplicaciones en Java.

La versatilidad del lenguaje de programación Java permite al programador crear distintos tipos de aplicaciones. A continuación, describiremos las características más relevantes de cada uno de ellos:

✓ Aplicaciones de consola:

→ Son programas independientes al igual que los creados con los lenguajes tradicionales.

→ Se componen como mínimo de un archivo .class que debe contar necesariamente con el método main.

→ No necesitan un navegador web y se ejecutan cuando invocamos el comando Java para iniciar la Máquina Virtual de Java (JVM). De no encontrarse el método main la aplicación no podrá ejecutarse.

→ Las aplicaciones de consola leen y escriben hacia y desde la entrada y salida estándar, sin ninguna interfaz gráfica de usuario.

✓ Aplicaciones gráficas:

→ Aquellas que utilizan las clases con capacidades gráficas, como Swing que es la biblioteca para la interfaz gráfica de usuario avanzada de la plataforma Java SE.

→ Incluyen las instrucciones import, que indican al compilador de Java que las clases del paquete javax.swing se incluyan en la compilación.

✓ Applets:

→ Son programas incrustados en otras aplicaciones, normalmente una página web que se muestra en un navegador. Cuando el navegador carga una web que contiene un applet, éste se descarga en el navegador web y comienza a ejecutarse. Esto nos permite crear programas que cualquier usuario puede ejecutar con tan solo cargar la página web en su navegador.

→ Se pueden descargar de Internet y se observan en un navegador. Los applets se descargan junto con una página HTML desde un servidor web y se ejecutan en la máquina cliente.

→ No tienen acceso a partes sensibles (por ejemplo: no pueden escribir archivos), a menos que uno mismo le dé los permisos necesarios en el sistema.

→ No tienen un método principal.

→ Son multiplataforma y pueden ejecutarse en cualquier navegador que soporte Java.

✓ Servlets:

→ Son componentes de la parte del servidor de Java EE, encargados de generar respuestas a las peticiones recibidas de los clientes.

→ Los servlets, al contrario de los applets, son programas que están pensados para trabajar en el lado del servidor y desarrollar aplicaciones Web que interactúen con los clientes.

✓ Midlets:

→ Son aplicaciones creadas en Java para su ejecución en sistemas de propósito simple o

dispositivos móviles. Los juegos Java creados para teléfonos móviles son midlets.

→ Son programas creados para dispositivos embebidos (se dedican a una sola actividad), más específicamente para la máquina virtual Java MicroEdition (Java ME).

→ Generalmente son juegos y aplicaciones que se ejecutan en teléfonos móviles.

### 3.- Entornos Integrados de Desarrollo (IDE).

En los comienzos de Java la utilización de la línea de comandos era algo habitual. El programador escribía el código fuente empleando un editor de texto básico, seguidamente, pasaba a utilizar un compilador y con él obtenía el código compilado. En un paso posterior, necesitaba emplear una tercera herramienta para el ensamblado del programa. Por último, podía probar a través de la línea de comandos el archivo ejecutable. El problema surgía cuando se producía algún error, lo que provocaba tener que volver a iniciar el proceso completo.

Para el lenguaje de programación Java existen múltiples alternativas, siendo los principales entornos

de desarrollo NetBeans (que cuenta con el apoyo de la empresa Sun), Eclipse y JCreator. Los dos primeros son gratuitos, con soporte de idiomas y multiplataforma (Windows, Linux, MacOS).