Seguimiento 1

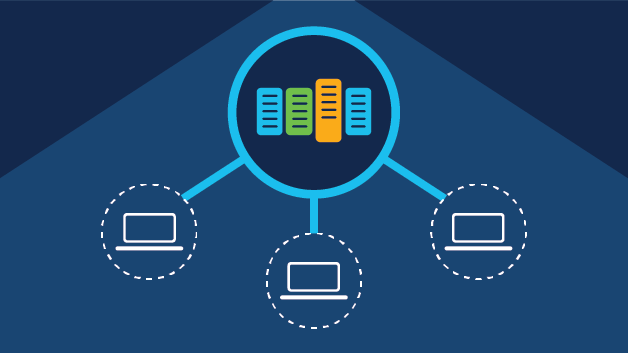
Sebastian Ordoñez Giraldo, Juan José Rodríguez

1.

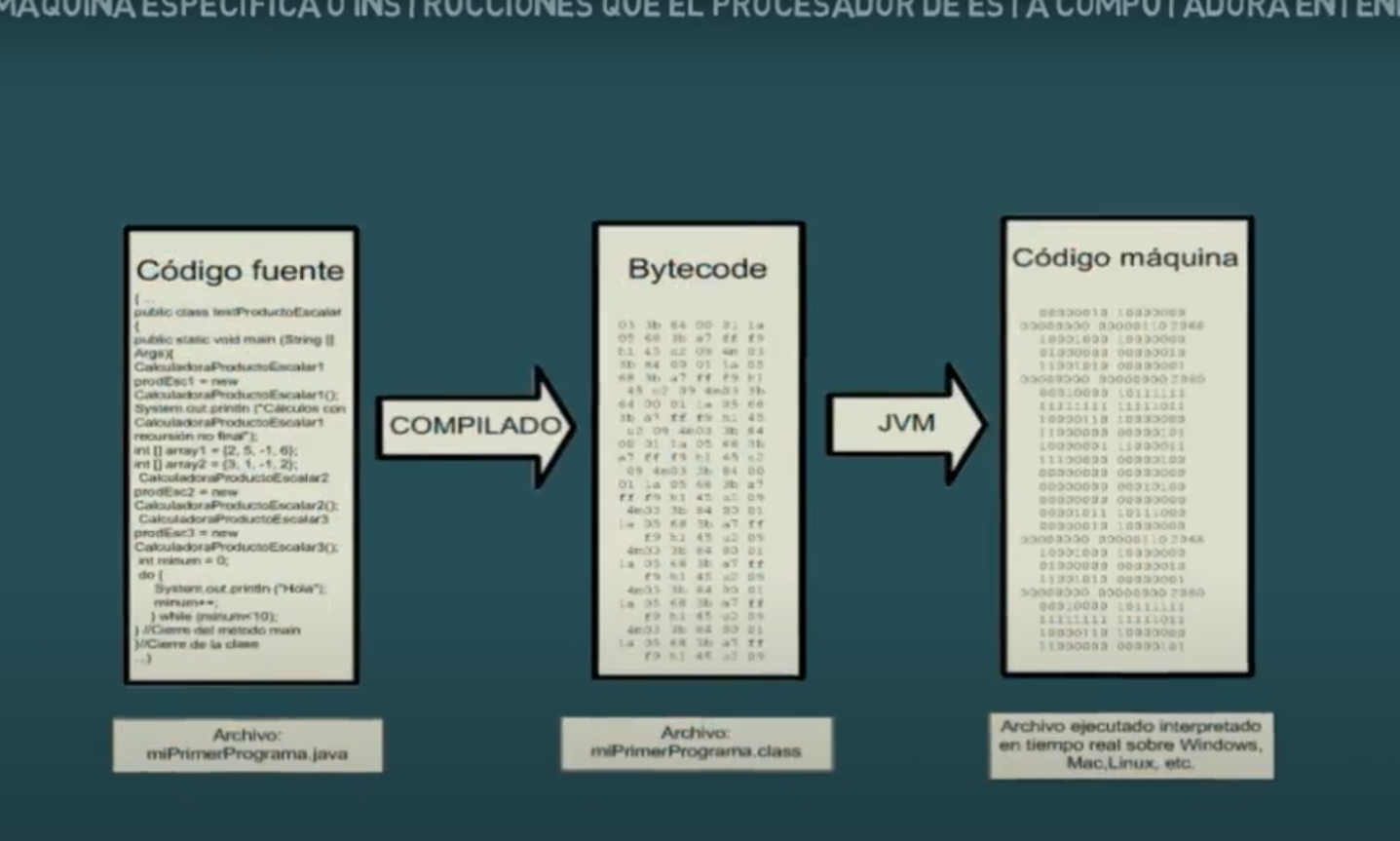
A. Las preguntas que plantean en el libro son 8:

* What is a virtual machine?

Mucha gente cree que es un computador adentro de un computador, pero un (VM) es un entorno digital que emula un sistema informático completo.



* What is bytecode?

Es un código complejo de entender para los humanos, este es el lenguaje de máquina. 

* Is javac a compiler?

En el libro usan el termino compilador de código fuente para referirse a la producción de archivos de clase por javac.

* Why is it called “bytecode”?

El código de instrucción (opcode) es simplemente un byte único, por lo que solo hay 256 instrucciones posibles, en la practica algunas no se utilizan todos, solo se usan 200

* Is bytecode optimized?

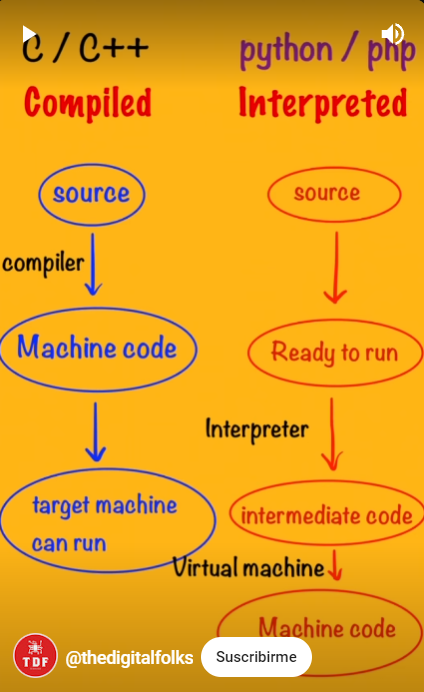
Antiguamente la plataforma javac producía bytecode muy optimizado, pero esto resulto en un error. Pero con la llegada de la compilación JUST IN TIME los métodos se volvieron mas rápidos. Un compilador JIT traduce un programa por partes al inicio

* Is bytecode really machine independent? What about things like endianness?

El formato de bytecode es siempre el mismo independientemente del tipo de máquina en la que se creó. Esto incluye el orden de los bytes (a veces llamado "endianness")

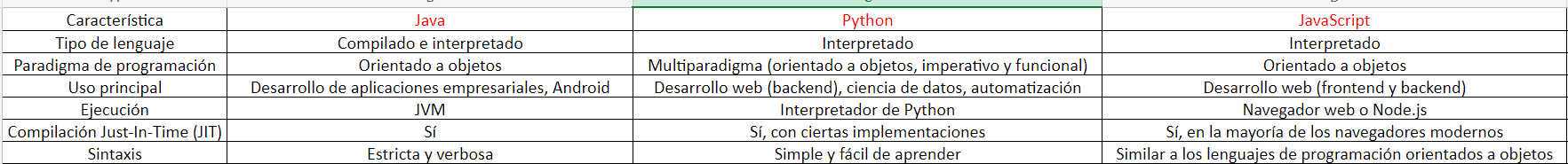
* Is Java an interpreted language?

Java es compilador e interprete, Así, dependiendo del entorno de ejecución, el bytecode de Java puede compilarse con antelación y ejecutarse como código nativo, compilarse justo a tiempo y ejecutarse, o interpretarse directamente1. Esta combinación única de compilación e interpretación es la razón por la que a menudo se hace referencia a Java como un "lenguaje compilador-intérprete"



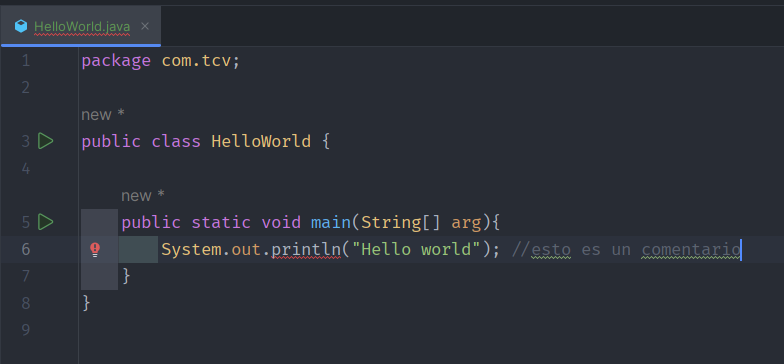
* Can other languages run on the JVM?

Si de hecho muchos pueden corren Java Virtual Machine(JVM)

B.

C.

-Comentario se utiliza el //



-Una palabra reservada es una palabra que forma parte de la sintaxis del lenguaje y no pueden utilizarse para nombrar variables, clases, etc.).

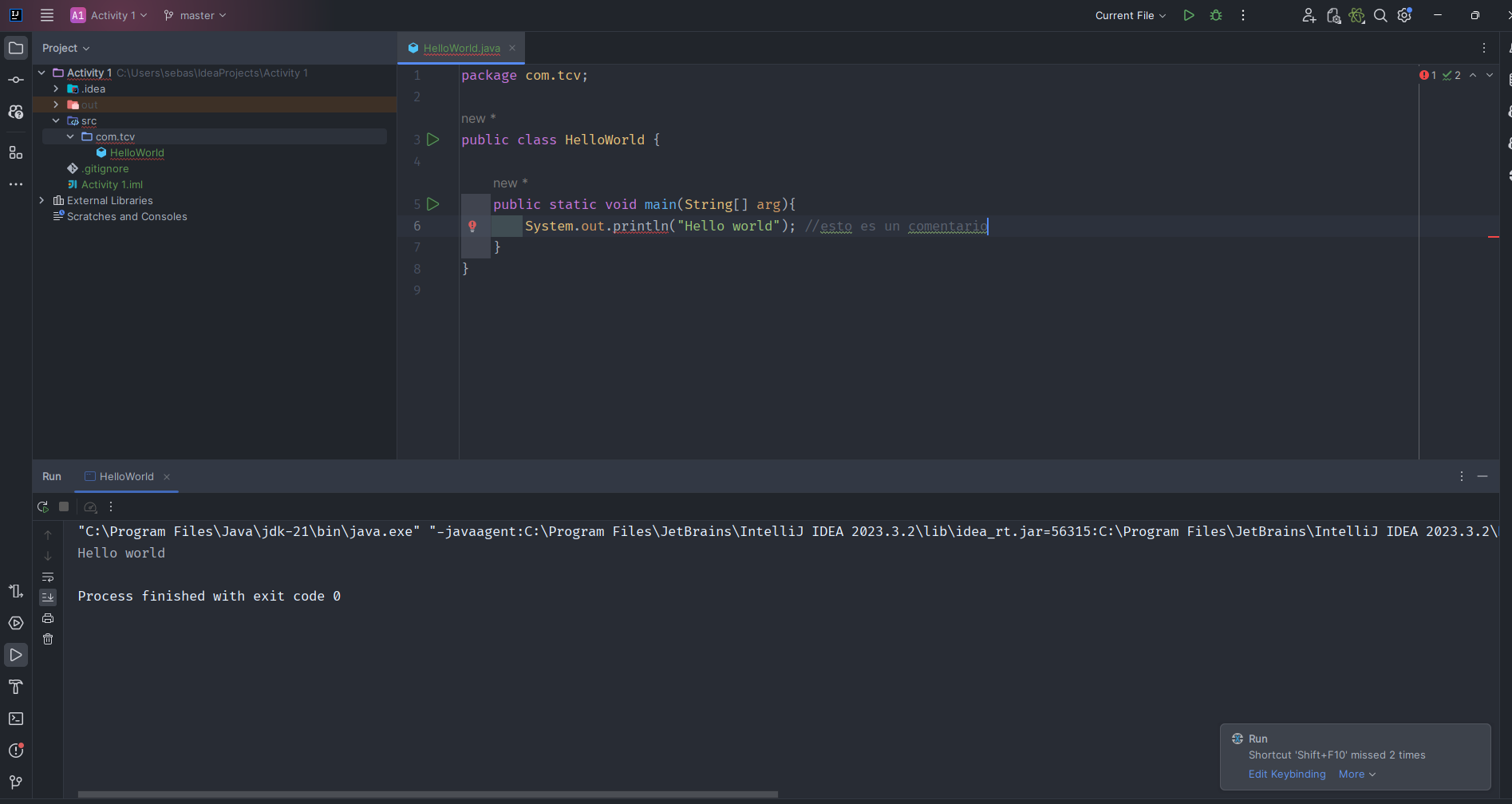
-

En programación Java, un identificador es un nombre que se da a un elemento del programa, como una variable, una clase, un método, etc. Los identificadores son esenciales porque permiten a los programadores referirse a estos elementos de forma coherente y sistemática en todo el programa.

* Un identificador es una secuencia de caracteres, guiones bajos (\_), signos de dólar ($) o dígitos.
* Un identificador puede comenzar con cualquier carácter, incluidos los signos de dólar y guiones bajos, excepto números o dígitos.
* Un identificador no puede ser una palabra reservada de un idioma.
* Un identificador no puede ser verdadero, falso o nulo.
* Un identificador puede tener cualquier longitud.

- En Java, un literal es un valor constante que se puede asignar a una variable. Los literales representan valores fijos en el código. aquí algunos ejemplos de literales en Java:

* Literales enteros: Son números sin decimales. Pueden ser de tipo byte, short, int o long. Por ejemplo, int número = 100;.
* Literales de coma flotante: Son números con decimales y pueden ser de tipo float o double. Por ejemplo, double number = 10.5;.
* Literales de carácter: Son caracteres simples de tipo char. Se definen entre comillas simples. Por ejemplo, char letra = 'a';.
* Literales de cadena: Son secuencias de caracteres y son de tipo String. Se definen entre comillas dobles. Por ejemplo, String saludo = "Hola Mundo";.
* Literales booleanos: Son los valores verdadero y falso.

2) 

3) Los datos primitivos en Java son los tipos de datos más básicos que proporciona el lenguaje. No son objetos y, por lo tanto, no tienen métodos ni atributos. Aquí están los ocho tipos de datos primitivos en Java:

byte: Es un tipo de dato entero de 8 bits con signo. Su valor mínimo es -128 y el máximo 127.

short: Es un tipo de dato entero con signo de 16 bits. Su valor mínimo es -32.768 y el máximo 32.767.

int: Es un tipo de dato entero con signo de 32 bits. Su valor mínimo es -2.147.483.648 y el máximo 2.147.483.647.

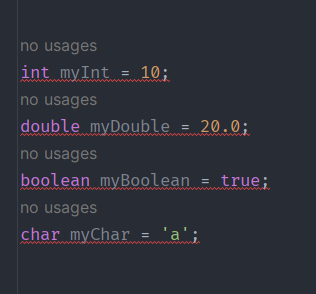
long: Es un tipo de dato entero con signo de 64 bits. Su valor mínimo es -9.223.372.036.854.775.808 y el máximo 9.223.372.036.854.775.807.

float: Tipo de datos de coma flotante de 32 bits y precisión simple.

doble: Un tipo de datos de coma flotante de doble precisión de 64 bits.

boolean: Sólo tiene dos valores posibles: falso y verdadero

char: Un tipo de datos de caracteres Unicode de 16 bits.



4) En Java, los operadores se utilizan para realizar diversas operaciones y manipulaciones sobre datos y variables. Los operadores en Java se pueden clasificar en los siguientes tipos

Operadores aritméticos: Se utilizan para realizar operaciones matemáticas, como sumas, restas, multiplicaciones, divisiones y módulos.

Operadores de asignación: Se utilizan para asignar un valor a una variable.

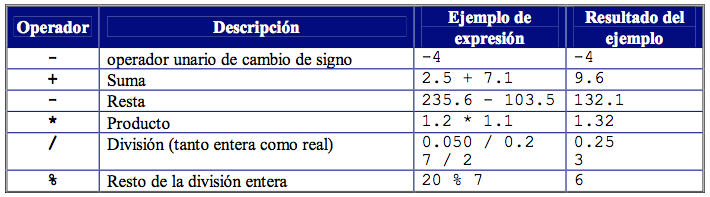
Operadores relacionales: Se utilizan para comparar dos valores. Incluyen operadores como igual a, no igual a, mayor que, menor que, mayor o igual que y menor o igual que.

Operadores lógicos: Se utilizan para realizar operaciones lógicas, como AND, OR y NOT.

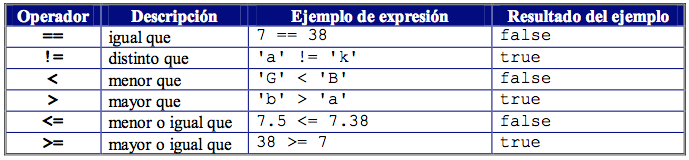
Operadores bit a bit: Se utilizan para manipular datos a nivel de bits.

Operadores varios: Incluyen operadores tales como operadores de incremento/decremento, operador ternario, etc.

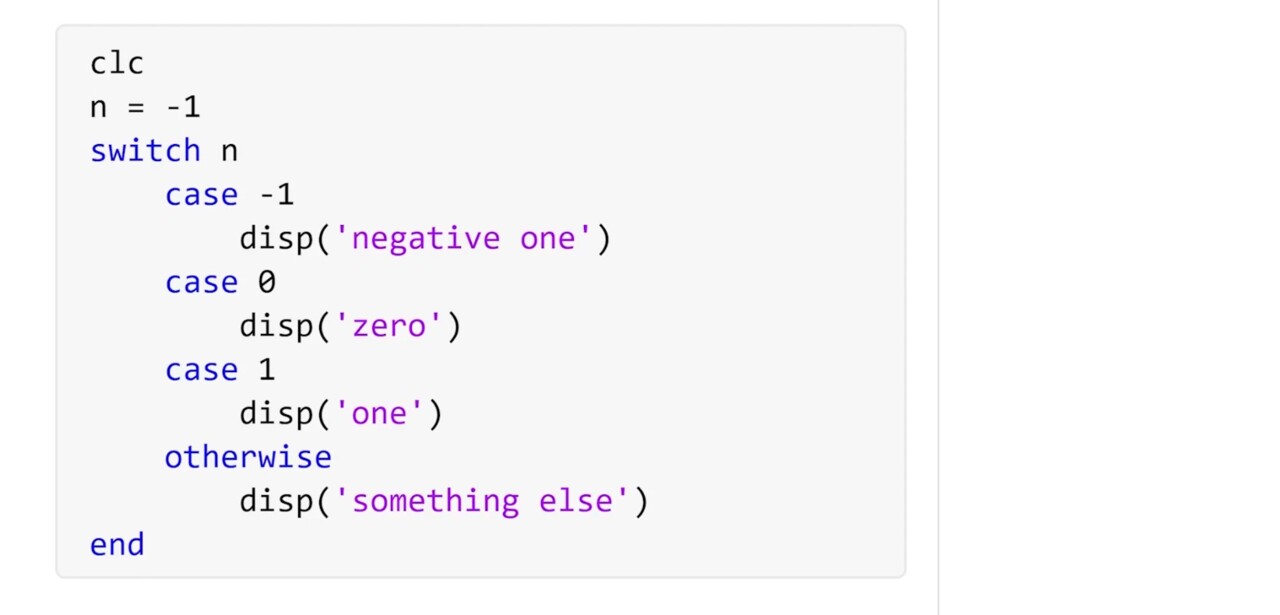
5) Los operadores de incremento (++) y decremento (--) en Java son útiles para modificar el valor de una variable en una unidad.



6) Los operadores de comparación en Java se utilizan para comparar dos valores. Éstos son los más comunes



9) La sentencia switch en Java es una sentencia de control de flujo que permite a un programa evaluar una variable y elegir una ruta de ejecución basada en el valor de esa variable.



12) Las sentencias break y continue en Java se utilizan para controlar el flujo de ciclos (for, while, do-while).

-Break: Cuando se ejecuta la sentencia break dentro de un bucle, éste se interrumpe inmediatamente y el control del programa salta a la siguiente instrucción después del bucle.

-Continue: La sentencia continue se salta el resto de instrucciones del bucle actual y pasa a la siguiente iteración del bucle.

13) 