JAVA DESDE CERO

**¿Qué es Java?**

Es un lenguaje de programación orientado a objetos multiplataforma.

**¿Qué se necesita para programar en JAVA?**

1. Descargar el Java Development Kit o (JDK)
2. Una Pc con recursos minimos.
3. Conocer el bloc de notas y Símbolo del sistema

Al no programar con editores profesionales o entornos de Desarrollo integrados (IDE), la retención de sintaxis es mayor y el desarrollo de la habilidad incrementa bastante al no tener ayuda.

(Entornos de desarrollo integrado (IDE) NetBeans – JC – Eclipse – Sublime Text - notepad)

**¿Qué se ha desarrollado en JAVA?**

Juego de **MINECRAFT:** en sus primeras versiones fue programado en java

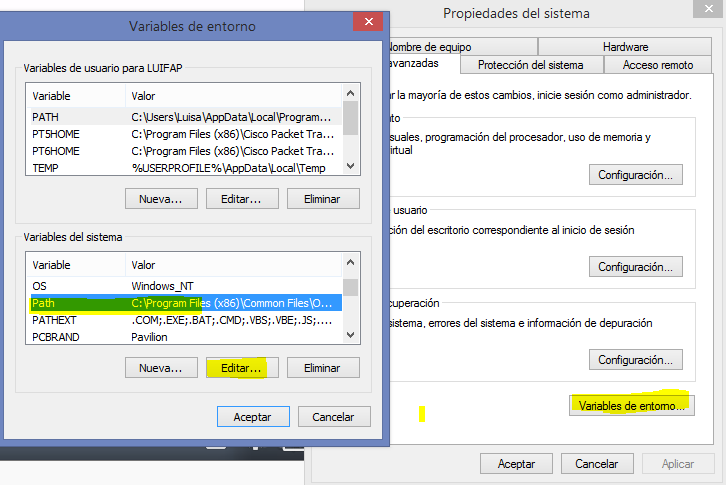
**OpenOffice.org:**Una copia de Microsoft office y es totalmente gratuito, es programado en Java

**Android:** Su base esta programada bajo el lenguaje de java y la mayoria de sus librerias es de java

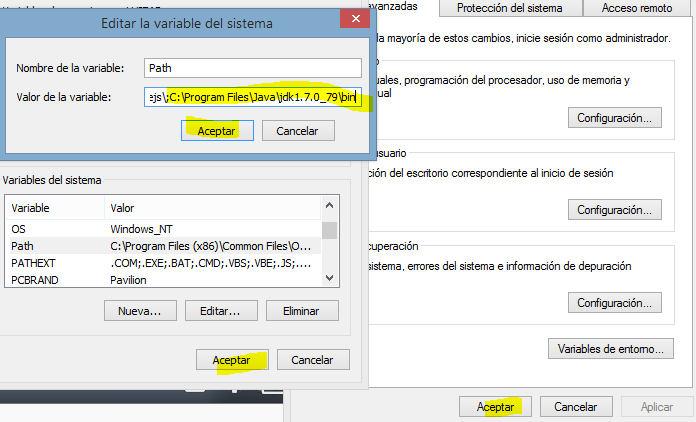
**PARTE PRACTICA:**

1. Se debe descargar el JDK de java en Oracle y se ejecuta.
2. configuración del PC para programar en Java:

* Copiamos la ruta de java: C:\Program Files\Java\jdk1.7.0\_79\bin
* Vamos al panel de control – sistema seguridad - sistema – configuración avanzada del sistema. Sale una ventana:

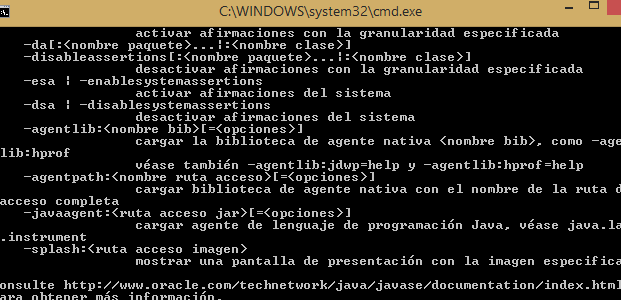


Al final de la línea de valor de variable agregamos un ; luego copiamos la ruta que habíamos tomado de java de jdk (C:\Program Files\Java\jdk1.7.0\_79\bin).



**Comprobación:**

Se abre el símbolo del sistema y se escribe **java** se da enter y si te sale toda una chorrera de información esta bien instalado y configurado java.



En java para ejecutar un programa se requiere de clases

**Las clases son** una plantilla donde vamos a escribir un código. Por el momento en el block de notas vamos a escribir todo el código, el símbolo del sistema lo vamos a utilizar únicamente, para compilar los programas y ejecutarlos.

Todo el código siempre se debe de leer de arriba a bajo y de derecha a izquierda por que asi lo lee el pc.

Primero debemos de escribir el nombre de la clase y se escribe una llave (llave que se abre y llave que se cierra. Dentro de las llaves se coloca el código

Ejemplo: public class luisa{} listo como se debe leer de derecha a izquierda se dice que hay una clase de nombre luisa y que esta clase es pública.

Luego se debe escribir el método mein para darle inicio al programa (para que arranque).

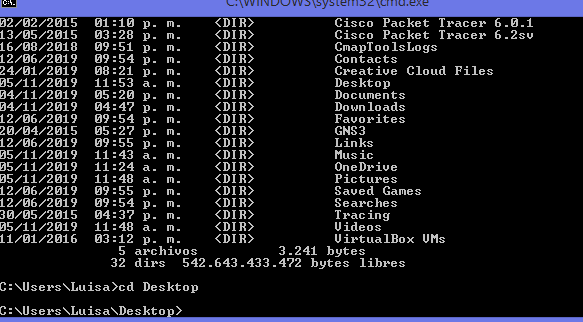
“el programador debe de mostrar un programa limpio y comprensible para eso se hace los inventados que es darle tabular o espacios en la siguiente línea de comandos.

Para mandar a imprimir en pantalla se debe de colocar **System.out.println(“”);**

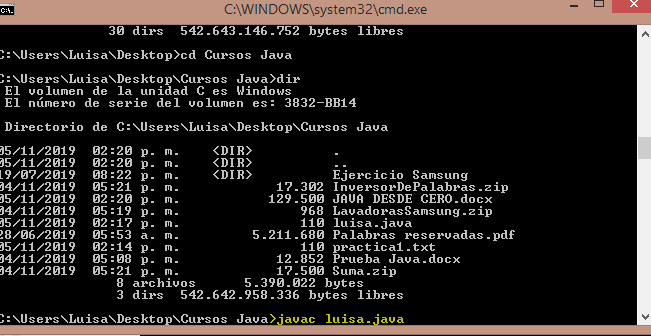
Siempre que se termine una línea de instrucción en llava se debe colocar un **;** el mensaje siempre debe ir entre comillas y entre paréntesis.

Luego se guarda el block de notas como un .txt para que no se nos borrre, luego de esto se va a guardar como un archivo java, se da en guardar como y se guarda con el mismo nombre de la clase. En este caso es luisa.java

Luego en el símbolo de sistema. Escribimos **dir,** para ver las carpetas que tenemos parecere la carpera del escritorio (desktop) entramos con **cd Desktop**



Java con la ayuda del complidaor traduce del lenguaje humano a lenguaje computadora hace una traducción bycots



Si aparece un error es por que estaba mal escrito. Y si no te aparecerá algo como:

Y te creara un archivo .class

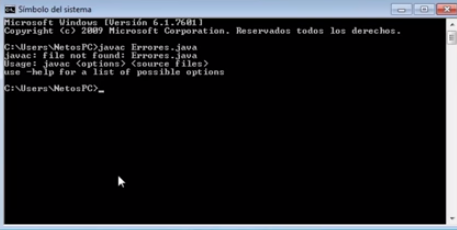


Errores sintácticos y errores lógicos

Nota: Las clases siempre empiezan con mayúscula ejemplo Practica – tampoco se debe dejar espacios se escribiría así: PracticaUno

**Error lógico:**

Se pueden causar por dos razones 1 por un error del código o un error lógico del programador. Ejemplo para ejecutar el programa escribimos javac Errores.java sin entrar al directorio donde se encuentra el archivo. Es más que todo por el programador



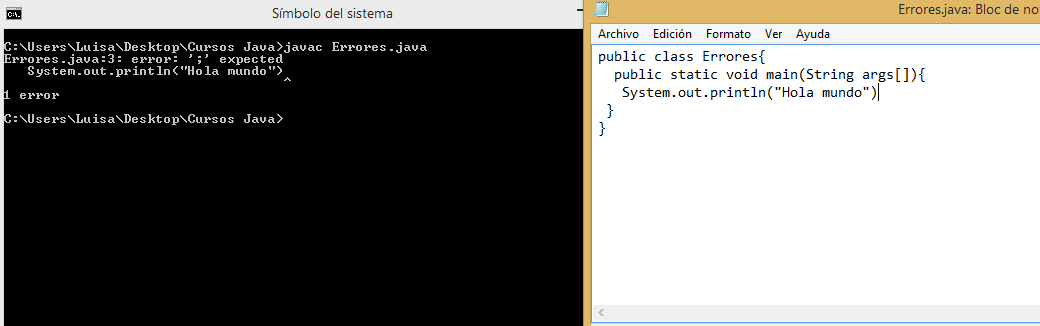
La segunda causa de error de lógica es más de lógica en el código y son más difíciles detectar, hay que tener mucho cuidado cuando se diseña el código.

Con el comando **cls**  limpias toda la pantalla del símbolo del sistema

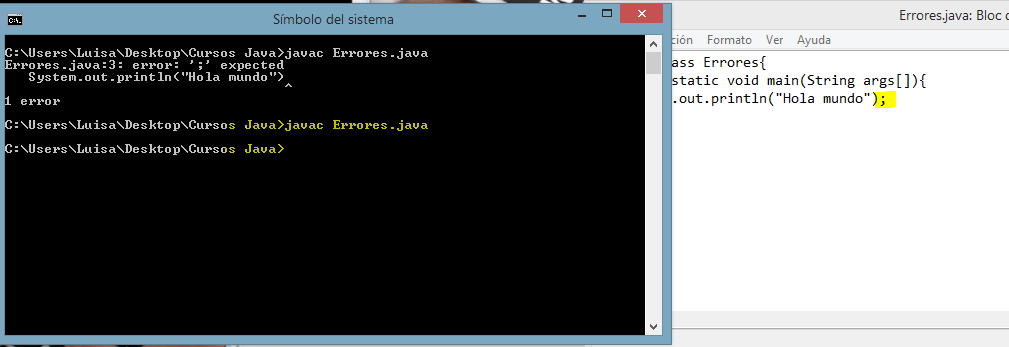
**Erros sintáctico:**

Es por ejemplo los ; o que te falto poner un ) o cerrar una { o escribiste mal un nombre. Un programa no se va a ejecutar si tiene alguno de estos errores.

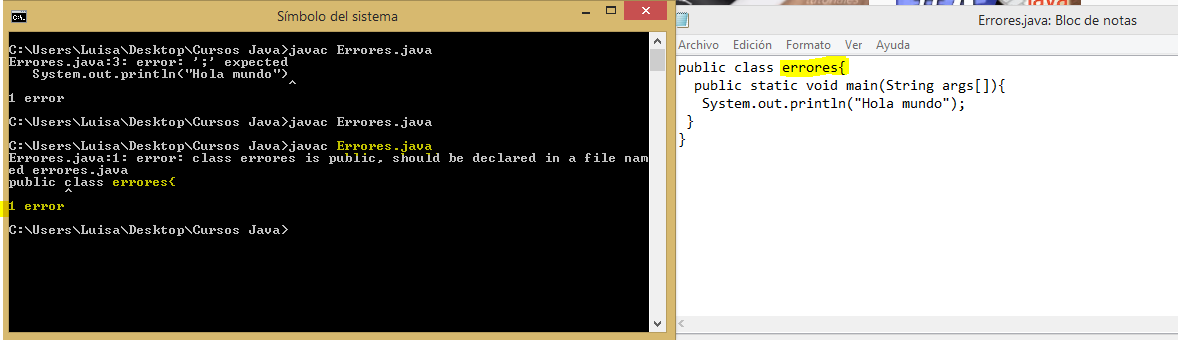
En este ejemplo cuando lo ejecutamos nos aparece un error y nos indica la línea en donde esta el error, y el error de este caso de sintaxis es el punto y coma;



Corregimos el error y compila bien



Este es otro ejemplo donde muestra que el nombre de la clase esta mal escrito debe ser en mayuscula mas no en miniscula



**LAS VARIABLES Y LOS TIPOS DE DATOS QUE EXISTEN EN JAVA**

**TRES TIPOS DE DATOS EN JAVA**

**Primitivos:**

**Enteros: (No pueden tener decimales)**

Byte(-128 hasta 127)

Short (-32, 768 hasta 32, 767)

Int(-2,147,483,648 hasta 2, 147, 483, 647)

Long (muy grande).

**Decimales:**

Float

Double

**Otros:**

Char (un solo carácter)

Boolean (true – false).

**Tipo Objeto**

**String (Cadena de texto)**

Permite crear cadena de texto todo lo que tenga que ver con texto y String empieza con Mayúscula.

Ejemplo

“Hola Mundo”

**Variable:**

Una variable es un espacio en memoria donde nos permite ingresar o alojar información de tipo texto o numerico.

Una variable se representa de esta manera:

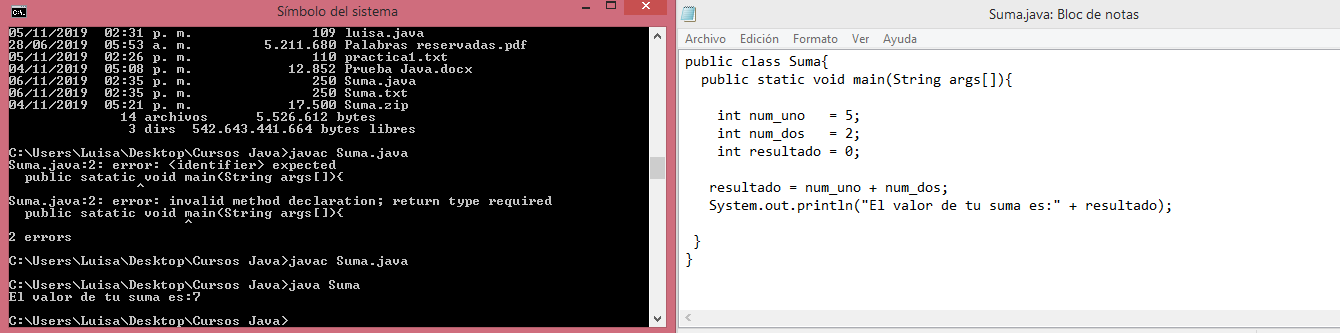
**Tipo:** int

20

**Nombre**: num\_uno

Este cuadro hace referencia al espacio en memoria que nosotros vamos a utilizar ya sea información o datos. En java necesitamos dos elementos un **Tipo** de dato (que tipo de información va a recibir) y un **nombre.**

**En este caso el tipo es Int (tipo numerico), y el nombre es num\_uno**

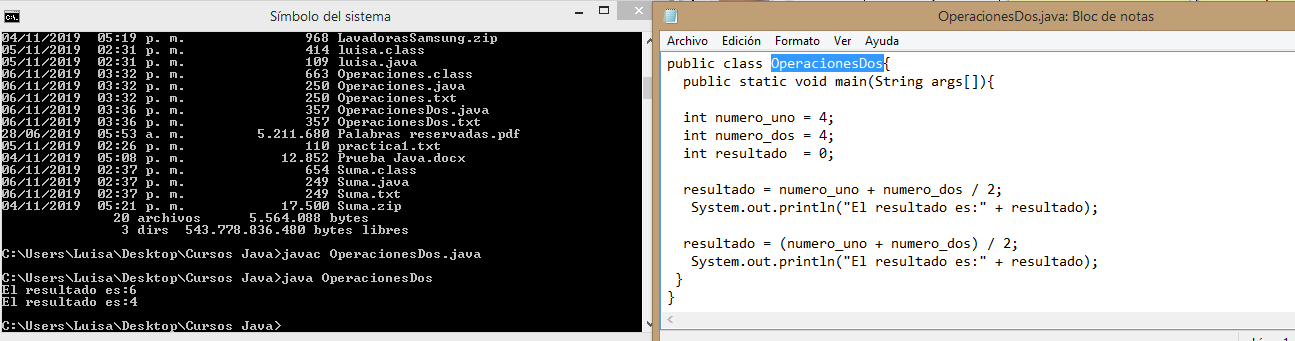


|  |  |  |  |
| --- | --- | --- | --- |
| **Operadores aritméticos & prioridad de los signos** | | | |
| **Signo** | **Nombre del signo** | **Tipo de operación** | **Código ASII** |
| **+** | Más | Suma | ALT + 43 |
| **-** | Menos | Resta | ALT + 45 |
| **\*** | Asterisco | Multiplicación | ALT +42 |
| **/** | Diagonal o Slash | División | ALT +47 |
| ^ | Acento Circunflejo | Potencia | ALT +94 |
| **(** | Paréntesis apertura | Prioridad entre operaciones | ALT +40 |
| **)** | Paréntesis cierre | Prioridad entre operaciones | ALT + 41 |



Primero se resuelve lo que hay entre paréntesis, despues resuelve la potencia, luego multiplicación y división y por ultimo la suma y la resta.

**Ejemplo que primero se resuelve lo que esta entre paréntesis:**



**ESTRUCTURAS CONDICIONALES CON if – else**

**Condiciones:**

**Situación:** ¿Mamá puedo ir a la fiesta?

**Condición:** Si quieres ir, debes hacer tu tarea

**ESSTRUCTURAS CONDICIONALES SIMPLES**

FIN

Te deja ir y te pide que regreses temprano.

Mamá revisa la tarea

Si o Verdadero

No o Falso

¿Hiciste tu tarea?

**Estructura condicionales Compuestas**

**(Instrucciones por las dos partes)**

Si o Verdadero

FIN

Te deja ir y te pide que regreses temprano.

No te deja ir y te pone hacer la tarea

Mamá revisa la tarea

No o Falso

¿Hiciste tu tarea?

**PRACTICA:**

Realizar un programa que calcule el promedio final de tres materias y que el programa nos indique si el alumno aprobó o reprobó.

Las Calificaciones son:

* Matemáticas: 5
* Biología: 8
* Química: 7

Inicio

Matemáticas = 5

Biología = 8

**Promedio = (Matemáticas + Biología + Química) / 3**

Promedio = 0

Química = 7

**NO**

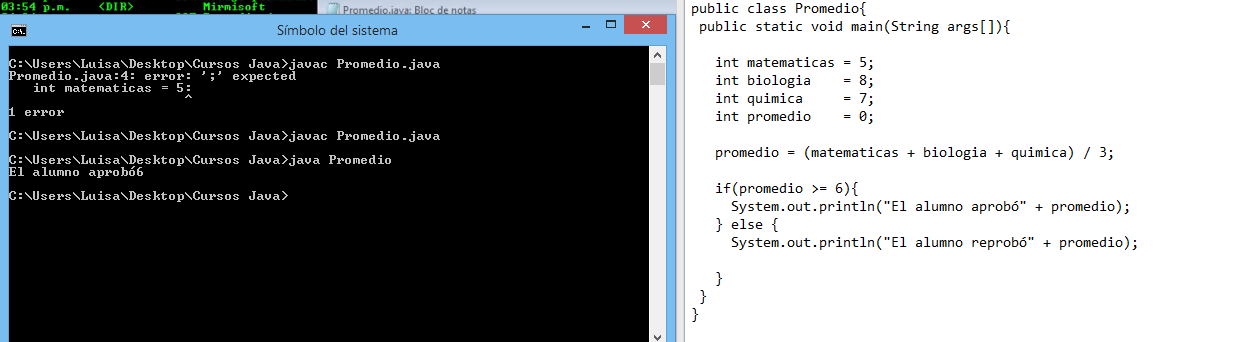
**SI**

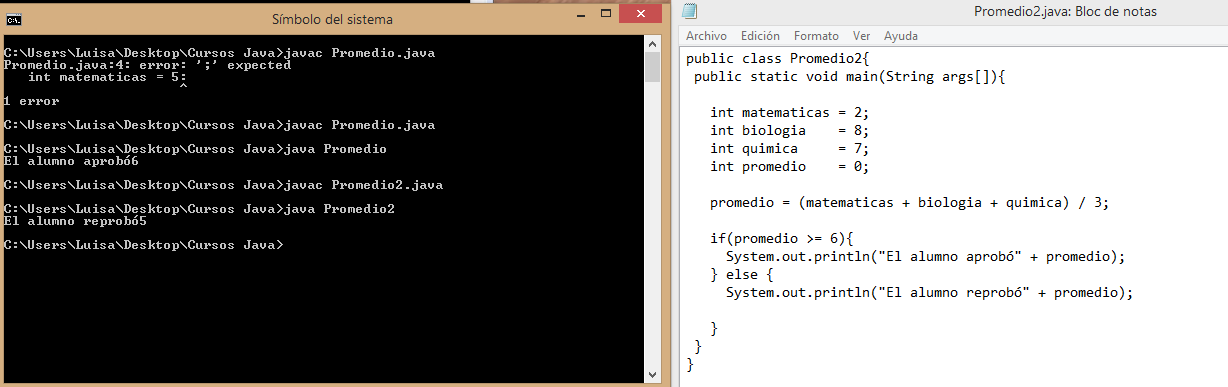
Promedio >=6

FIN

El alumno aprobó

El alumno reprobó





**ESTRUCTURAS CONDICIONALES ANIDADAS CON if – else**

Realizar un programa que pueda realizar una suma, una resta, una multiplicación o una división de dos números enteros, dependiendo de la decisión del usuario:

* Si operación es igual a 1, realizar una suma
* Si operación es igual a 2, realizar una resta
* Si operación es igual a 3, realizar una multiplicación
* Si operación es igual a 4, realizar una división

