Programación en Java

Servicio Nacional de Derechos Intelectuales (Senadi)
GYE - 0097898

SERVICIO NACIONAL DE DERECHOS INTELECTUALES

Autor
© Ediciones Espinosa

Dirección Editorial
© Ediciones Espinosa

Diseño e Ilustración
© Ediciones Espinosa

Nueva Edición
Año 2025

ISBN
978-9942-36-147-9

Guayaquil - Ecuador

EDICIONES ESPINOSA

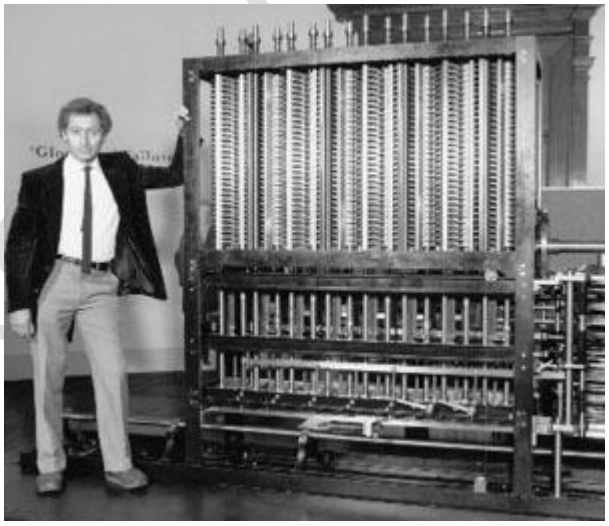
Lo mejor de enseñar es aprender

Historia de la programación

Gottfried Wilheml Von Leibniz (1646-1716), quien aprendió matemáticas de forma autodidacta (método no aconsejable en programación) construyó una máquina similar a la de Pascal, aunque algo más compleja, podía dividir, multiplicar y resolver raíces cuadradas.

Pero quien realmente influyó en el diseño de los primeros computadores fue **Charles Babbage** (1793-1871). Con la colaboración de la hija de Lord Byron, **Lady Ada Countess of Lovelace** (1815-1852), a la que debe su nombre el lenguaje ADA creado por el DoD (Departamento de defensa de Estados Unidos) en los años 70. Babbage diseñó y construyó la "máquina diferencial" para el cálculo de polinomios.

Más tarde diseñó la máquina analítica de propósito general, capaz de resolver cualquier operación matemática. Murió sin poder terminarla, debido al escepticismo de sus patrocinadores y a que la tecnología de la época no era lo suficientemente avanzada. Un equipo del Museo de las Ciencias de Londres, en 1991, consiguió construir la máquina analítica de Babbage, totalmente funcional, siguiendo sus dibujos y especificaciones.



Un hito importante en la historia de la informática fueron las tarjetas perforadas como medio para "alimentar" los computadores. Lady Ada Lovelace propuso la utilización de las tarjetas perforadas en la máquina de Babbage.

ÍNDICE

UNIDAD 1 – GENERALIDADES

UNIDAD 2 – ESTRUCTURAS SELECTIVAS

UNIDAD 3 – PROGRAMACIÓN ORIENTADA A OBJETOS

UNIDAD 4 – ESTRUCTURAS SELECTIVAS CON FORMULARIOS

UNIDAD 5 – INTRODUCCIÓN A LA PROGRAMACIÓN HTML

Unidad 1

GENERALIDADES

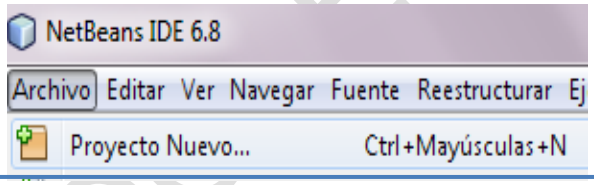
Historia de Java

Tal vez la contribución más importante a la fecha, por parte de la revolución del microprocesador, es que hizo posible el desarrollo de las computadoras personales, que ahora suman cientos de millones a nivel mundial.

Muchas personas creen que la siguiente área importante en la que los microprocesadores tendrán un profundo impacto, es en los dispositivos electrónicos para uso doméstico. Al reconocer esto, **Sun Microsystems** patrocinó en 1991 un proyecto interno de investigación denominado **Green**. El proyecto desembocó en el desarrollo de un lenguaje basado en **C++** al que su creador, **James Gosling**, llamó **Oak** debido a un árbol de roble que tenía a la vista desde su ventana en las oficinas de **Sun**.

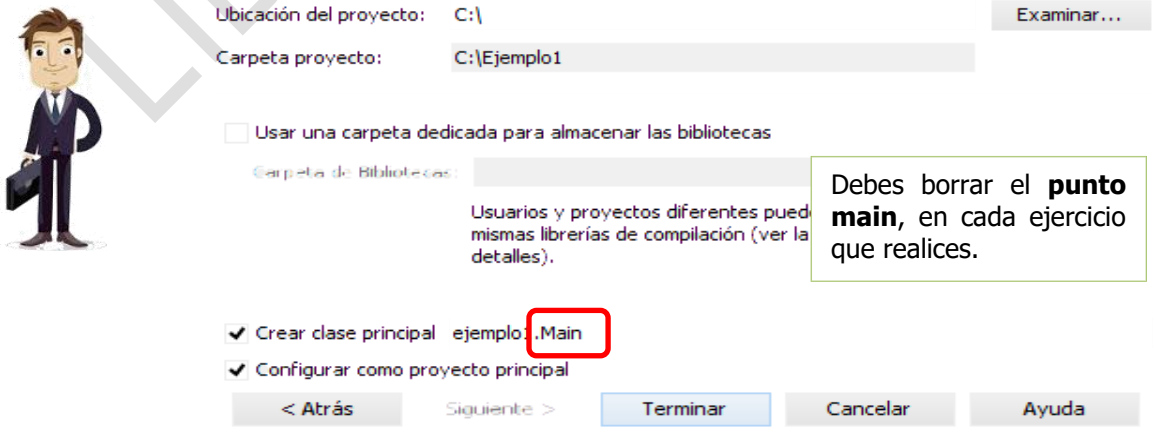
Posteriormente se descubrió que ya existía un lenguaje de programación con el mismo nombre. Fue entonces cuando un grupo de gente de **Sun** visitó una cafetería local y surgió el nombre **Java** (una variedad de café) y así quedó, como lo vemos en su ícono. Como lenguaje de programación para computadores, **Java** se introdujo a finales de 1995.

Pasos para crear un proyecto:
siga los siguientes pasos:
Archivo → Proyecto Nuevo



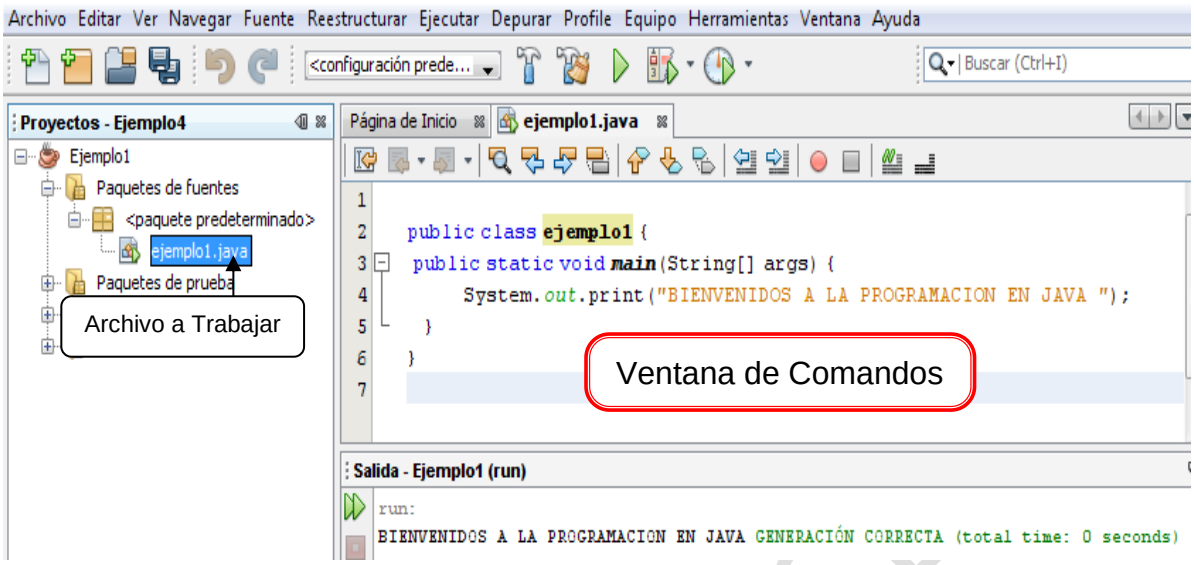
Se presenta: **Proyecto Nuevo**, **Seleccionar proyecto:**
Categorías: La carpeta **Java** y en **Proyectos**: **Aplicación Java**.

Visualizamos la **Nueva Aplicación de Java,** que sirve para almacenar los datos.
Nombre del proyecto: aquí va el nombre de nuestro (***Ejercicio1***).
Ubicación del Proyecto: es la ruta (***drive***) donde se guardará la clase.
Carpeta proyecto: que generalmente es el nombre de la carpeta donde se almacenará nuestra clase o ejercicio y clic en **Terminar.**



Debes borrar el **punto main**, en cada ejercicio que realices.

Al finalizar el editor (**Asistente**) genera un archivo, donde vamos a comenzar a escribir la codificación de nuestro proyecto:



Luego procedemos a escribir en la **Ventana de Comandos** la siguiente codificación:

System.out.print("BIENVENIDOS A LA PROGRAMACIÓN EN JAVA");

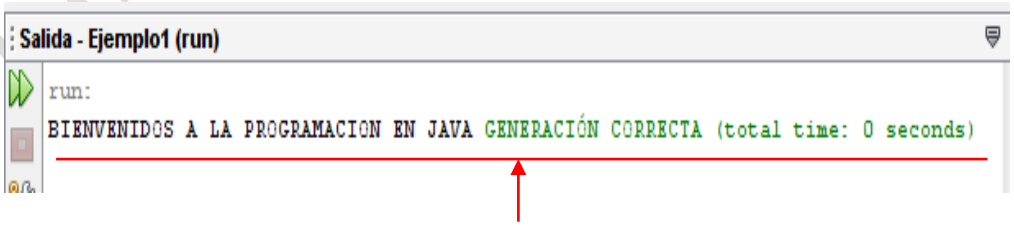
NOTA: Todo esto debe hacerlo entre las llaves.
Indica a la computadora que realice una acción, es decir, que presente, que muestre o visualice la **cadena** de caracteres contenida entre los caracteres de comillas dobles.

F6 es para ejecutar o el siguiente botón:

System.out. Se conoce como objeto de salida estándar y permite a las aplicaciones en Java mostrar conjuntos de caracteres en la ventana de comandos, desde la cual se ejecuta la aplicación en Java.

El método **System.out.print** muestra, presenta, visualiza o imprime una línea de texto en la ventana de comandos

El mismo que presenta un mensaje por **consola**, que es lo que está entre comillas, recuerda que todo **Mensaje** debe ir entre comillas dobles.



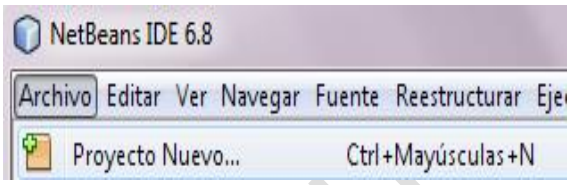
Como se dará cuenta el **print**, se ***concatena*** con otro texto, para que esto no ocurra trabaje con la instrucción **pritnln.**

Print. Esta función presenta, visualiza, muestra o imprime una expresión. Una expresión puede ser un mensaje, una variable, una función o cualquier combinación de ellas. Si la expresión es un **mensaje** debe ir entre comillas, ejemplo:

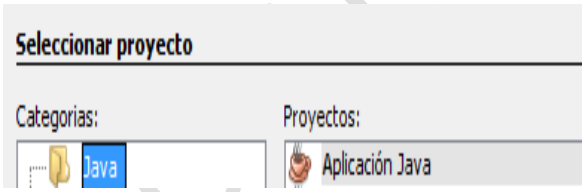
(“BIENVENIDOS A LA PROGRAMACIÓN EN JAVA”);

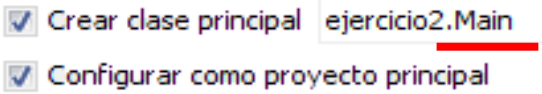
Punto y coma. La utilidad de la instrucción punto y coma **(;)** al final de cada una de estas líneas es la de separar dos instrucciones consecutivas; si usted no la incluye al momento de ejecutar el programa presentará error.

Ejercicio de Aplicación 1
A continuación, un proyecto que presenta varios mensajes por consola.
Archivo → Proyecto Nuevo:



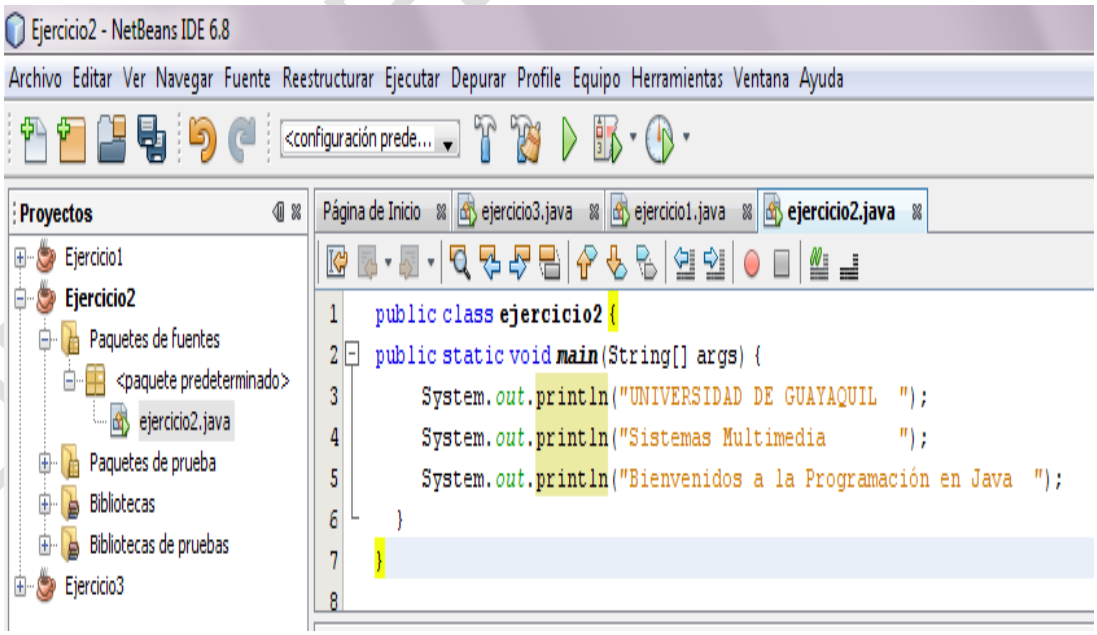
Se presenta el **Proyecto Nuevo**, **Seleccionar proyecto:**
Categorías: Carpeta **Java** y en **Proyectos**: **Aplicación Java**





Nota: En cada ejercicio debes borrar el **punto** y el **main**

Luego escriba las siguientes líneas:



F6 →para ejecutar el proyecto, y se visualizará por consola los siguientes mensajes:

```
Salida - Ejercicio2 (run)
run:
UNIVERSIDAD DE GUAYAQUIL
Sistemas Multimedia
Bienvenidos a la Programación en Java
GENERACIÓN CORRECTA (total time: 1 second)
```

Println

Parecida a la instrucción **print**, pero con la diferencia de que, por cada **println**, los mensajes se presentarán en una línea distinta, es como que usted utilice la tecla **enter**.

Ventana de Comandos

Programa usado para escribir cualquier tipo de texto (sin formato, ej: bloc de notas), en este caso, una codificación en **"Java".**

Compiladores

Un grupo de programas que se encargan de revisar los errores de una codificación (escritura de un programa, en nuestro caso una clase).

Programa Fuente

Es la codificación original de un programa, en cualquier lenguaje de Programación, en nuestro caso, en **"Java",** el programa es una clase, y cada clase en un archivo**.**

Programa Objeto

Es el programa en **"Java",** listo para ejecutar libre de errores, es decir, es el resultado de la compilación de un programa Fuente.

TRABAJO AUTÓNOMO 1

1. ¿Qué es programación, de acuerdo con lo estudiado?

2. Cite tres ejemplos de programas.

3. Definición de programa.

4. ¿Qué es un lenguaje de programación, de acuerdo con lo estudiado?

5. Clasificación de los lenguajes de programación.

6. Cite como nace el nombre de Java.

7. ¿Cuál es la diferencia entre Programación Lineal y Programación Orientada a Objetos?

8. Escriba los pasos para crear un nuevo Proyecto

9. ¿Cuáles son los orígenes de Java?

10. ¿Qué características debe tener un computador para instalar Java?

11. Ejecute un programa con la nómina de tus profesores de tu curso.

12. Desarrolle una aplicación en Java que presente tus asignaturas.

13. Presenta los periféricos de un computador.

14. Visualice la primera estrofa del himno de tu colegio.

15. Desarrolla una aplicación con las versiones de Java.

16. Presente las partes internas del computador.

Librería, Biblioteca o Paquete.

Es un conjunto de archivos que contienen una serie de funciones que serán usadas en el presente programa, que generalmente deben ser importadas dentro del compilador. Una librería, biblioteca o paquete es una colección de clases de nombres, esto se lo conoce como ***bibliotecas de clase Java*** o la ***Interfaz de Programación de Aplicaciones de Java*** (**API** por sus siglas en inglés).

Las librerías o paquetes del **API** se dividen en básicos y opcionales. Los nombres de la mayoría de las librerías del **API** de Java comienzan, ya sea con **"java"** (paquetes básicos) o **"javax"** (paquetes opcionales), muchos de ellos se incluyen como parte del Kit de desarrollo del software para Java.

Uso del paquete javax.swing.*;

Este paquete proporciona dos de las herramientas más importantes para el desarrollo de entornos gráficos. Específicamente, para mostrar mensajes al usuario a través de ventanas, se emplea la siguiente sintaxis::

JOptionPane.showMessageDialog(null, "Mensaje");

En donde:

Null, es argumento que, SIMPRE lo pondremos en el método **MessageDialog** "**Mensaje**", es la cadena de caracteres que queremos presentar o visualizar.

Trabajar con JoptionPane

Es habitual que el usuario tenga que ingresar datos, confirmar una acción o pedirle algún dato sencillo, darle a elegir entre varias acciones o simplemente mostrarle un aviso. Por lo tanto, es necesario abrir una ventana secundaria, donde el usuario deba realizar alguna acción que se requiera y cerrarla.

Método JoptionPane

Proporciona cuadros de diálogos previamente empaquetados, los cuales permiten a los programadores mostrar ventanas que contengan mensajes para los usuarios.

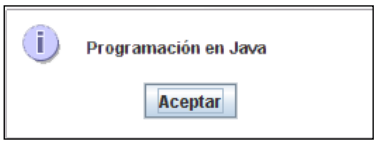
Comentarios //

Los comentarios no afectan la ejecución del programa. Son opcionales. Todo comentario comienza con dos barras (***//***), y no es necesario finalizarlas. Cualquier carácter entre estas marcas es ignorado por el compilador de **"Java".**

Ejercicio de Aplicación 2

A continuación, un proyecto que presenta un mensaje en un cuadro de diálogo.

```
1   //Llamamos a los Paquetes de java
2   import javax.swing.JOptionPane; //Vamos a trabajar con JOptionPa
3   public class ejercicio3 {
4    public static void main(String[] args) {
5   JOptionPane.showMessageDialog(null, "Programación en Java ");
6     }
```



AULA INVERTIDA

Escanea el código QR para ver el video y responde cinco preguntas que te ayudarán para la siguiente clase.

1. ¿Cuál es el propósito principal del método showMessageDialog?
   A) Guardar datos en un archivo.
   B) Mostrar un mensaje en una ventana emergente.
   C) Imprimir en la consola.
   D) Mostrar un cuadro de texto.

2. ¿Qué sucede si no se incluyen argumentos al invocar el método showMessageDialog?
   A) No se mostrará la ventana emergente.
   B) El programa se ejecuta normalmente.
   C) Se genera un error de compilación.
   D) Se generará un error de compilación.

3. ¿Cuál es la librería necesaria para usar el método showMessageDialog en Java?
   A) java.lang.String.
   B) java.awt.Graphics.
   C) javax.swing.JOptionPane.
   D) java.util.Scanner.

4. ¿Qué argumento es necesario incluir al usar el método showMessageDialog?
   A) Un número entero.
   B) Una cadena vacía.
   C) Null.
   D) Un objeto gráfico.

5. Al ejecutar la función showMessageDialog con un mensaje personalizado, ¿qué se espera que ocurra?
   A) Se mostrará un mensaje en la consola.
   B) El programa se cerrará.
   C) No se mostrará nada.
   D) Se mostrará una ventana emergente con el mensaje.

Línea 1, → Es un comentario.

Línea 2, → **import javax.swing. JOptionPane**;
Indica que el programa importará la librería **javax.swing** que utilizaremos en la clase **JOptionPane.**

Línea 3, → Comienza la declaración de la **clase ejercicio3**; el nombre de archivo para esta clase **public** debe ser **ejercicio3**

Línea 4, → Es el punto de toda aplicación en Java. Los paréntesis después de ***main*** indican que este es un bloque de construcción del programa, al cual se llama *método*. La palabra clave ***void*** indica que es un método y que realizará una tarea (mostrará un texto).

(String args[ ]) es una parte requerida de la declaración del método ***main***.

Línea 5, → Llama al método ***showMessageDialog*** de la clase ***JOptionPane*** para mostrar un cuadro de diálogo que contiene un mensaje y requiere de dos argumentos, el primero siempre será la palabra clave ***null*** que debe ir separado por una coma (**,**) y el segundo es el **mensaje** a mostrar en el cuadro de diálogo.

Método showMessageDialog

Presenta una ventana de aviso, la misma que se cierra hasta que el usuario realice una acción en la misma.

Ejercicio de Aplicación 3
El siguiente proyecto, visualiza un cuadro de diálogo con un mensaje en varias líneas.

```
1  import javax.swing.JOptionPane;
2  public class ejercicio5 {
3  public static void main(String[] args) {
4  JOptionPane.showMessageDialog(null,"Universidad de Guayaquil \nCarrera Multimedia \nProgramacion Java");
5   }
```

Manipulador de Formato

Existe una serie de caracteres especiales que no podríamos representar con letras o números, ejemplo: un salto de línea, o un tabulador. Esto es lo que se conoce como **Manipulador de Formato:**

\n	salto de línea
\r	tabulador horizontal

Palabras Reservadas

Existe una serie de ***palabras reservadas*** las cuales tienen un significado especial para ***Java*** y por lo tanto no se pueden utilizar como nombres de variables. Dichas palabras son:

abstract	boolean	Break	byte	case	catch
char	const	Class	continue	default	do
double	else	Extends	final	finally	float
for	goto	If	implements	import	instanceof
int	interface	Long	native	new	null
throws	transient	Try	void	volatile	while

Variable

Una variable es un ***casillero de memoria*** en el cual podemos almacenar datos; estos datos pueden variar dependiendo de la ejecución del programa. Toda variable debe tener un nombre. Los nombres de las variables deben cumplir con las siguientes reglas:

- Deben de empezar siempre con una ***letra***, (minúscula) y pueden ser una ***combinación*** de ***letras*** y ***números***. Por ejemplo, nombres correctos de variables: **suma, xx, m1, m2, xyz123, sueldo,** etc.…

- **No** pueden tener ***espacios en blanco*** ni símbolos especiales intermedios. Para nombres de variables largos, use abreviaturas (sin puntos). Por ejemplo: incorrecto es usar **sueldo por hora**, correcto sería **suelxhor**.

- Deben ser ***nombres cortos*** y significativos. Evite usar nombres largos y difíciles. Por ejemplo, en lugar de usar la variable **promedio** use **prom**.

Cada nombre de variable debe ser ***único***, es decir, no puede existir otra variable con el mismo nombre, en la codificación del programa.

Tipos de Variables

Todas las variables en el lenguaje **Java** deben tener un tipo de dato.

El tipo de la variable determina los valores que la variable puede contener y las operaciones que se pueden realizar con ella.

TIPO	RANGO
String	Cadena de Caracteres
Int	Número real, cantidades pequeñas
Float	-32768…..32767 números enteros
Doublé	Flotante doble

Ejemplo, para el uso de los tipos de variables:

String	Nombres, apellidos, dirección, estado civil, cargo, es decir, toda una cadena de caracteres.
Int	Edad, cantidad, días, meses, años, número de hijos y cantidades pequeñas, pero enteras.
Float	Estatura, IVA, subtotal, resultado de una división, cantidades que contengan números con decimales o números enteros grandes
Doublé	Para cantidades de más de 10 cifras, ejemplo: número de población, extensión de un territorio, etc.

Operadores Aritméticos

La mayoría de los programas realizan cálculos aritméticos que permiten efectuar ***procesos*** o cálculos ***matemáticos elementales*****.** Los operadores aritméticos son:

Operador	Operación	Ejemplo
*	MULTIPLICACIÓN	15 * 2 = 30
/	DIVISIÓN	15 / 2 = 7.5
%	RESIDUO DE DIVISIÓN	15 % 2 = 1
+	SUMA	15 + 2 =17
-	RESTA	15 – 2 = 13

En el caso de los operadores **"/"** y **%,** son exclusivamente para la división.
El operador **"/"**es usado para la división decimal, es decir, la división común y corriente. Por ejemplo:

6 / 3 = 2
6 / 4 = 1.5
7 / 2 = 3.5
4 / 3 = 1.33333

El operador **%** obtiene el residuo, es decir, lo que sobra de una división. Por ejemplo:

(9 **%** **2**); → 1 (sobra 1 al dividir 9 para 2)
(6 **%** 4); → 2 (divida 6 para 4 y le sobran 2)
(14 **%** 3); → 2 (porque 14 / 3 es 4 sobrando 2)
(10 **%** 5); → 0 (divida 10 para 5 y no le sobra nada)

Reglas de Prioridad

Dentro de una expresión matemática compleja siempre se efectúan, ***primero*** los ***paréntesis*** más internos, luego las divisiones y las multiplicaciones y al final se realizan las sumas y restas.

Por ejemplo: 5 + 4 / 2 – 1

Primero se efectúa 4 / 2, o sea 2 y luego 5 + 2 - 1. El resultado es 6.

Reglas de Prioridad

( )	**Prioridad Máxima**
/, %, *	**Prioridad Intermedia**
+, -	**Prioridad Mínima**

Por ejemplo:

5 + 2 * 4 - 3

Es 10; primero se efectúa 2*4 y luego se suma y se resta. Sin embargo, los ***paréntesis*** siempre se evalúan ***primeros***.
Por ejemplo:

(5 + 2) * 4 - 3

Es 25; primero se hace el paréntesis (5+2) y luego se multiplica por 4 y se resta 3.

Veamos ahora los siguientes ejemplos:

A) 2 + 1 * 5 + 2
Es 9, porque primero es **1 * 5 = 5**; luego **2 + 5 + 2 = 9**

B) (5 % 2) + 2 / 2
Primero es **% (5 % 2)** → **1** y **2 / 2** → **1**; entonces **1 + 1 = 2**

C) 5 + 2+3 / 2
Primero **3 / 2 = 1.5**; más **5 + 2 + 1.5 = 8.5**

Fórmulas

Una fórmula es una expresión que se utiliza para ***obtener*** un resultado. Por ejemplo:

su = pn + st

Es la fórmula que representa la suma de dos valores, donde la variable **"pn"** representa el primer valor, la variable **"st"** el segundo valor y la **"su"** la suma.

Las fórmulas siempre se evalúan de ***derecha*** a ***izquierda***, es decir, colocando la expresión a calcular al lado derecho **del igual** y la variable que guarda el resultado del **lado izquierdo**.

La siguiente fórmula representa el promedio de tres calificaciones de un estudiante:

Expresión a calcular

Resultado a Obtener → **pro** = (pn + sn + tn) / 3

Método showInputDialog()

Permite crear un campo de texto en la ventana para que el usuario pueda ingresar datos libremente, y estos son almacenados en una variable de memoria.

Ejercicio de Aplicación 4

El siguiente proyecto solicita el ingreso de su nombre y edad, y luego presenta los datos ingresados, en la misma línea.

```
1  //Lamamos a los paquetes de java
2  import javax.swing.JOptionPane;
3  public class ejemplo4 {
4  public static void main(String[] args) {
5    //Declaramos a las variables que almacenaran lo que vamos a ingresar
6    String ed,nom;
7    //nom = almacena los datos a ingresar y showInputDialog, crea un campo de texto
8    nom=JOptionPane.showInputDialog(" Ingrese su nombre -->>");
9    //ed = alamacena los datos ingresados y showInputDialog, crea un campo de texto
10   ed=JOptionPane.showInputDialog(" Ingrese su edad   -->>");
11   //showMessageDialog presenta un mensaje y el conrenido de las variables nom y ed
12 JOptionPane.showMessageDialog(null,"Tu nombre es  "+nom+" y tienes  "+ed+" años ");
13 }
```

Línea 2,

import javax.swing.JOptionPane;

Indica que el programa importara la librería de **javax.swing** que vamos a trabajar, en este caso a **JOptionPane**, que sirve para los cuadros de diálogos.

Línea 6,

String ed, nom; declaramos las variables de tipo cadena de caracteres, que sirven para almacenar los datos ingresados.

Línea 8 y 10,

Las variables **ed y nom** almacenarán los datos ingresados y **showInputDialog** crea un campo de texto, el mismo que utilizaremos para ingresar los datos.

Línea 12,

JOptionPane.showMessageDialog, permite presentar un cuadro de diálogo, acompañado de un mensaje y lo almacenado en sus variables respectivas.

Ejercicio de Aplicación 5

Realice un proyecto que ingrese dos números y obtenga las cuatro operaciones matemáticas.

```
1  import javax.swing.JOptionPane;
2  public class ejercicio1 {
3   public static void main(String[] args) {
4      int pn,sn, su, re, mu, di;
5   pn=Integer.parseInt(JOptionPane.showInputDialog("Primer  Numero "));
6   sn=Integer.parseInt(JOptionPane.showInputDialog("Segundo Numero "));
7   su=pn+sn;
8   re=pn-sn;
9   mu=pn*sn;
10  di=pn/sn;
11
12  JOptionPane.showMessageDialog(null,"    La  Suma  es      " + su +
13                                    "\n  La  Resta es      " + re +
14                                    "\n La Multiplicacion " + mu+
15                                    "\n La Division       " + di);
```