



Argentina  
programa  
4.0

# Introducción a Algoritmos y Java Parte I

---

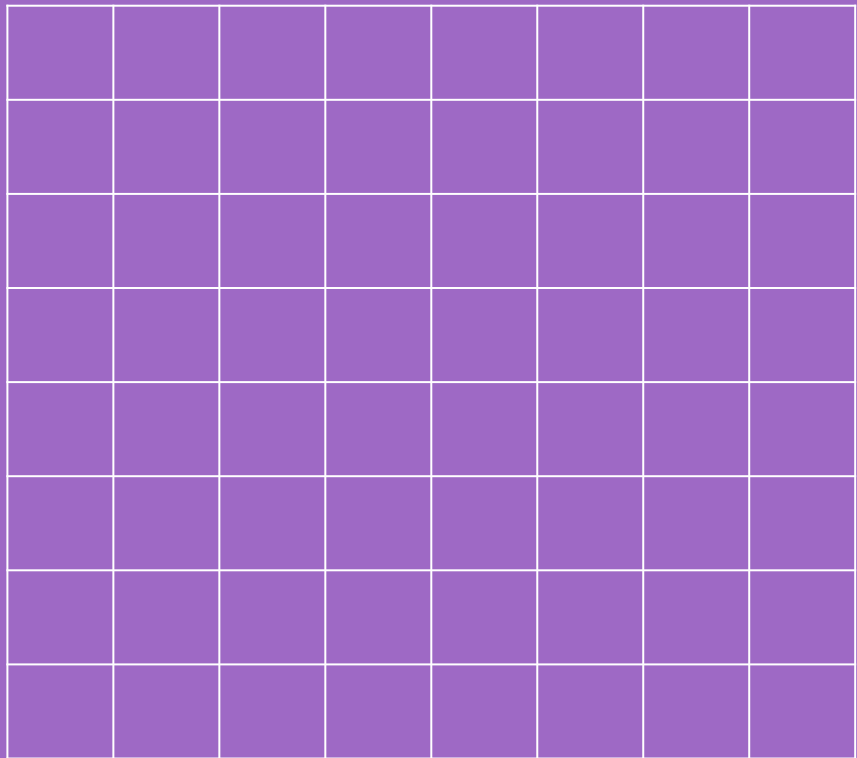
“Desarrollador Java Inicial”

# Agenda

- Introducción
- Java - Características
- Elementos básicos Java
- Ejemplos

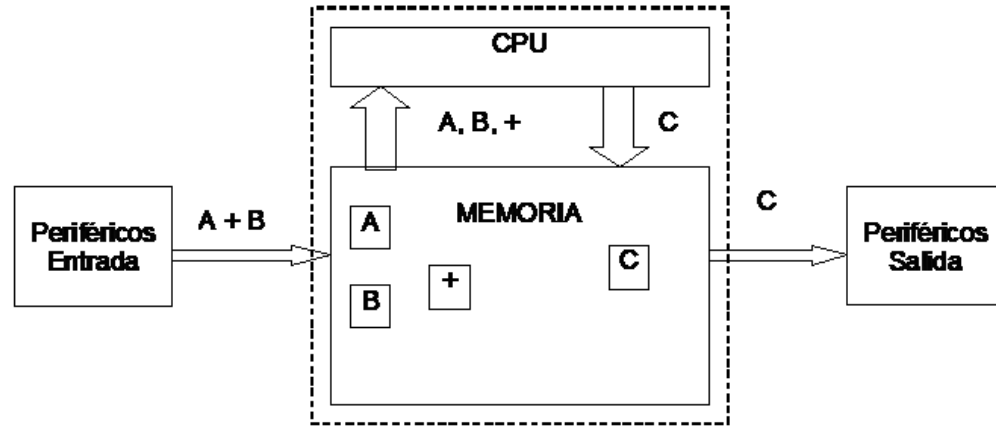
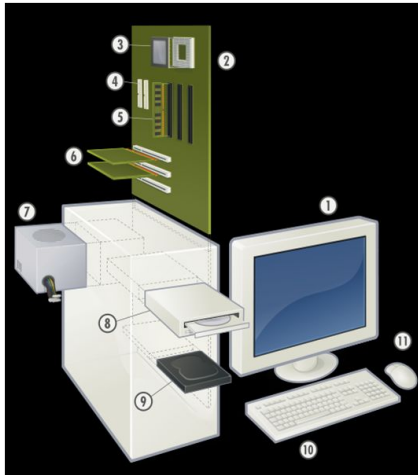


# Introducción



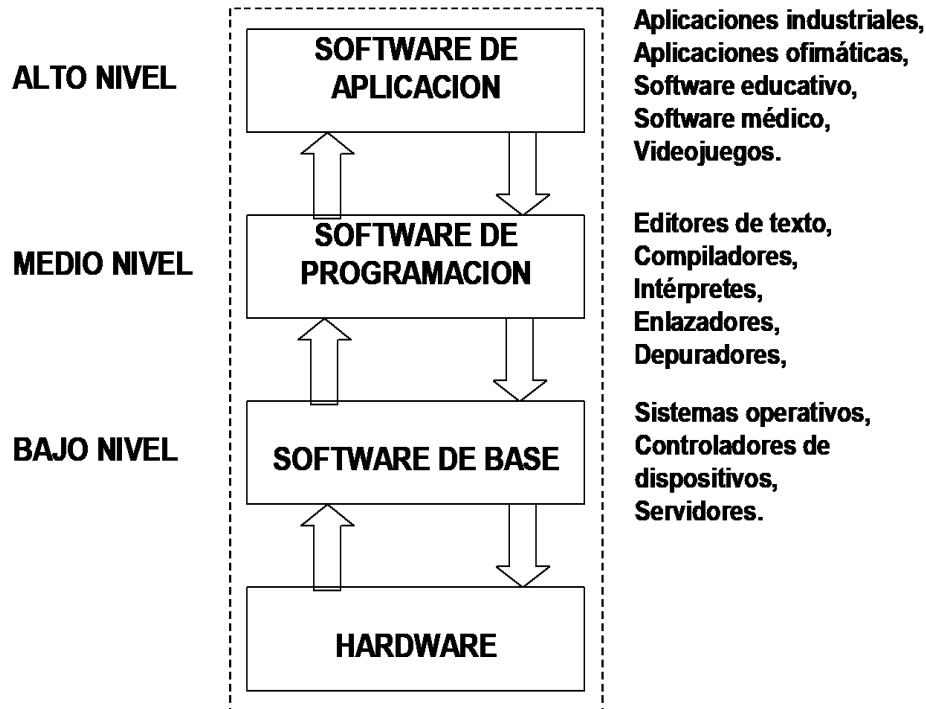
# Hardware

- Partes físicas: componentes eléctricos, electrónicos, electromecánicos, mecánicos, etc.



# Software

- Componentes intangibles: conjunto de programas y procedimientos para que un hardware actúe.



# Recursos de una PC

- Memoria
- Procesador
- Entrada / Salida
  - Discos / USB
  - Red
  - Sonido
  - Video

Administrador de tareas

Archivo Opciones Vista

Procesos Rendimiento Historial de aplicaciones Inicio Usuarios Detalles Servicios

Nombre	Estado	9% CPU	43% Memoria	0% Disco	0% Red	7% GPU	Motor de GPU	Consumo de e
Aplicaciones (4)								
> Administrador de tareas		0,5%	26,2 MB	0 MB/s	0 Mbps	0%	GPU 0 - 3D	Muy baja
> Google Chrome (21)		1,1%	1.740,6 MB	0 MB/s	0 Mbps	0,1%		Muy baja
> Mail		0%	0 MB	0 MB/s	0 Mbps	0%		Muy baja
> Spotify (5)		0%	45,9 MB	0 MB/s	0 Mbps	0%		Muy baja
Procesos en segundo plano (1...								
AcroTray (32 bits)		0%	0,5 MB	0 MB/s	0 Mbps	0%		Muy baja
Adobe AcroCEF (32 bits)		0%	4,4 MB	0 MB/s	0 Mbps	0%		Muy baja
Adobe AcroCEF (32 bits)		0%	3,5 MB	0 MB/s	0 Mbps	0%		Muy baja
Adobe Collaboration Synchroni...		0%	0,9 MB	0 MB/s	0 Mbps	0%		Muy baja
Adobe Collaboration Synchroni...		0,1%	2,5 MB	0 MB/s	0 Mbps	0%		Muy baja
Adobe Genuine Software Integri...		0%	1,0 MB	0 MB/s	0 Mbps	0%		Muy baja
Adobe Genuine Software Servic...		0%	1,7 MB	0 MB/s	0 Mbps	0%		Muy baja
Adobe IPC Broker (32 bits)		0%	2,0 MB	0 MB/s	0 Mbps	0%		Muy baja
Aplicación de subsistema de cola		0%	1,9 MB	0 MB/s	0 Mbps	0%		Muy baja

< Menos detalles Finalizar tarea

# Sistema Operativo

## Objetivos

- Facilidad de uso
- Eficiencia / Administración de recursos

## Funciones

- Interface de usuario: CLI (Command Line Interface) - GUI (Graphical User Interface)
- Sistema de Archivos
- Ejecución de programas

# Algoritmos y Programas

- **Algoritmo:** conjunto finito de pasos para resolver un problema.



- **Programa:** conjunto de instrucciones que se ejecutan en un hardware.

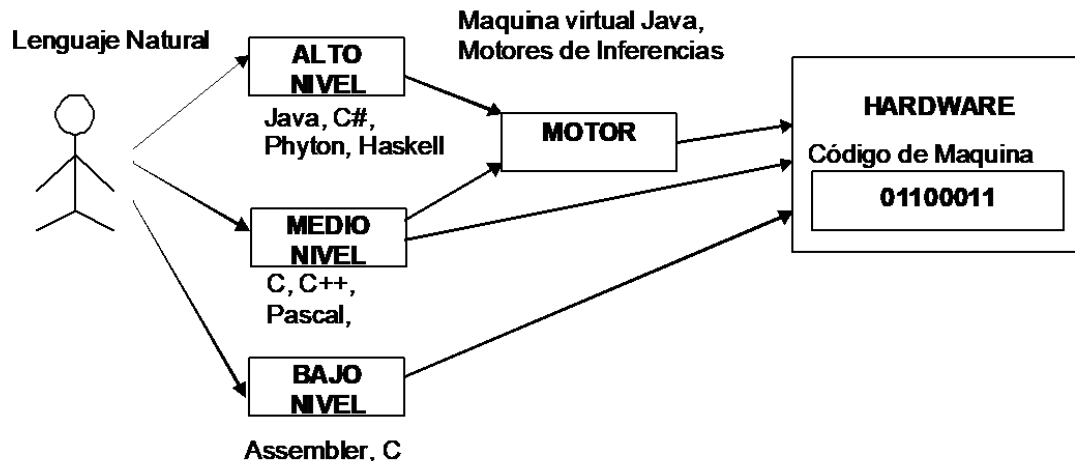




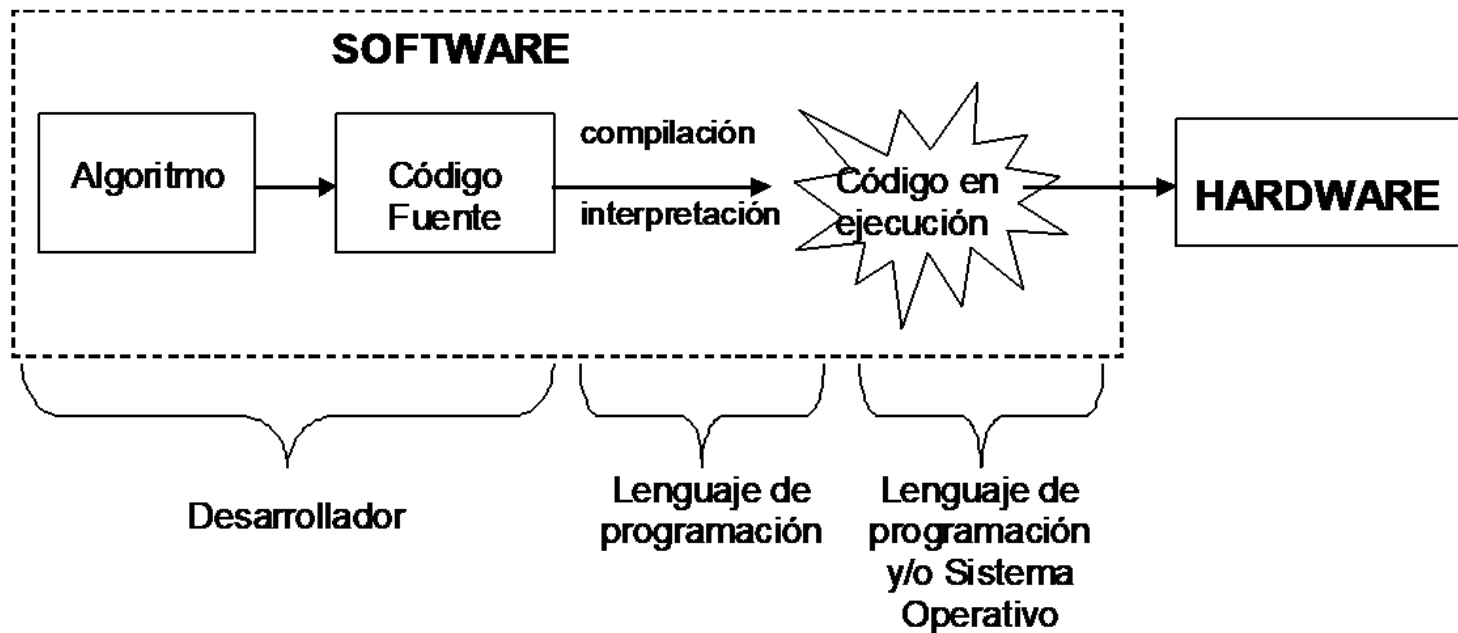
# Lenguajes de Programación

- Lenguaje formal que le permite al programador escribir un programa para que un hardware actúe.

## Niveles de abstracción de los lenguajes de programación



# Partes para ejecutar un programa



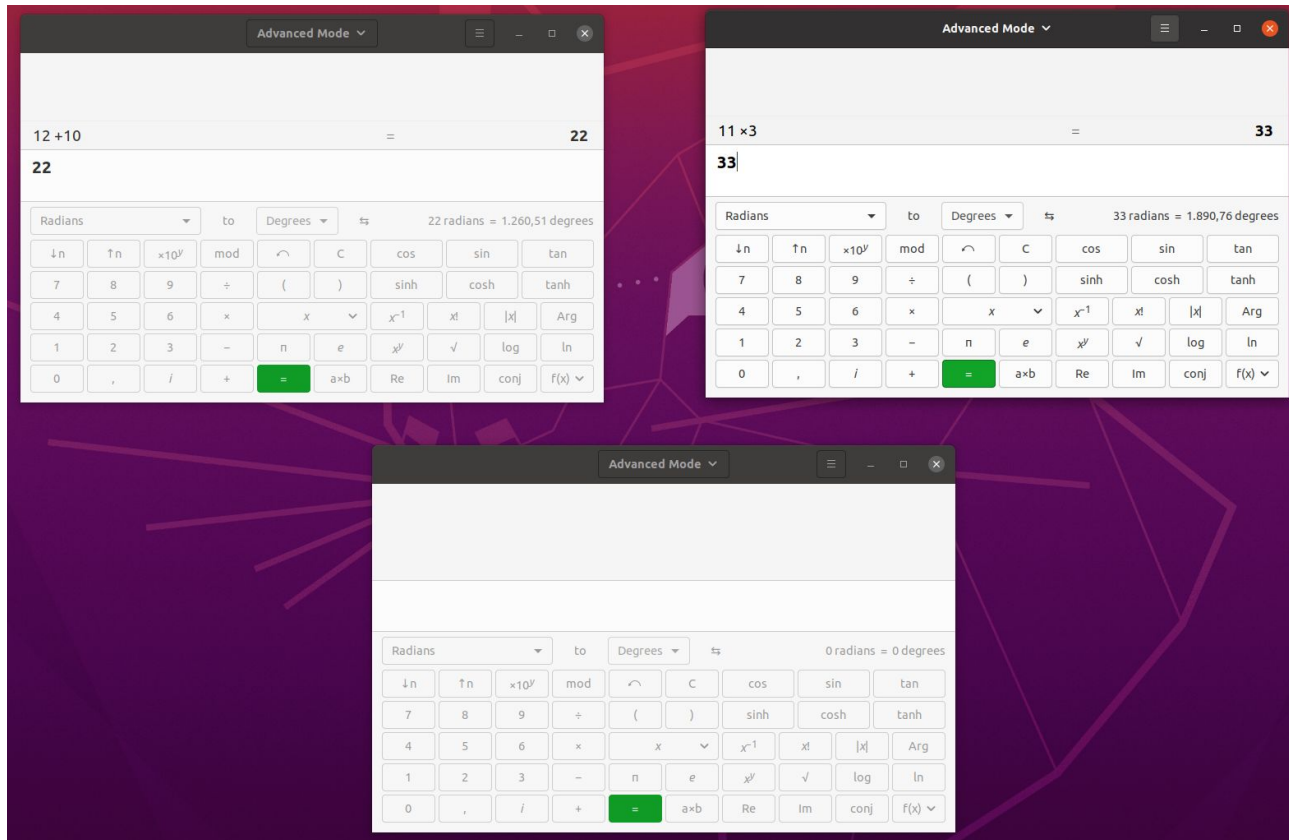
# Programas vs Procesos

*¿Cuál es la diferencia entre un programa y un proceso?*

## Proceso

- Unidad de asignación de recursos y planificación del SO
- Asignación de CPU = que decide cuando se ejecuta
- Recursos son:
  - Periféricos / GUI
  - Archivos
  - Memoria
  - Red (por ejemplo puertos)
  - etc...

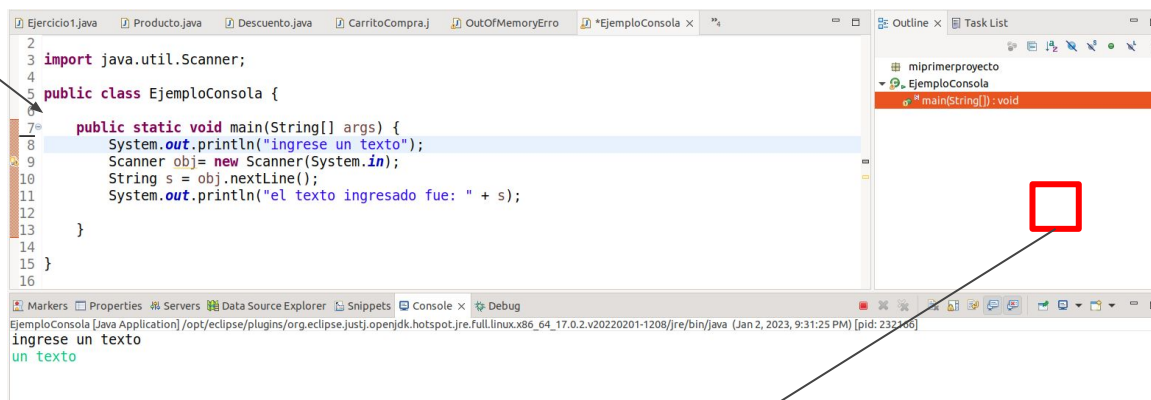
## Programas vs. Procesos - Ejemplo



Aca vemos 3  
**procesos**  
calculadora, los 3  
con datos distintos  
(ocupando distintas  
partes de la  
memoria). Pero  
**programa**  
calculadora hay **uno solo**.

# Programas vs. Procesos

Código fuente en el lenguaje de programación Java



The screenshot shows an IDE with a Java source file named 'EjemploConsola.java'. The code defines a class 'EjemploConsola' with a 'main' method that prompts the user to enter text and prints it. The console at the bottom shows the output 'ingrese un texto' and 'un texto'. A red square highlights a button in the IDE's toolbar, and an arrow points from the text 'Se pueden correr varios procesos del mismo código al mismo tiempo' to it.

```
2
3 import java.util.Scanner;
4
5 public class EjemploConsola {
6
7     public static void main(String[] args) {
8         System.out.println("ingrese un texto");
9         Scanner obj = new Scanner(System.in);
10        String s = obj.nextLine();
11        System.out.println("el texto ingresado fue: " + s);
12
13    }
14
15 }
16
```

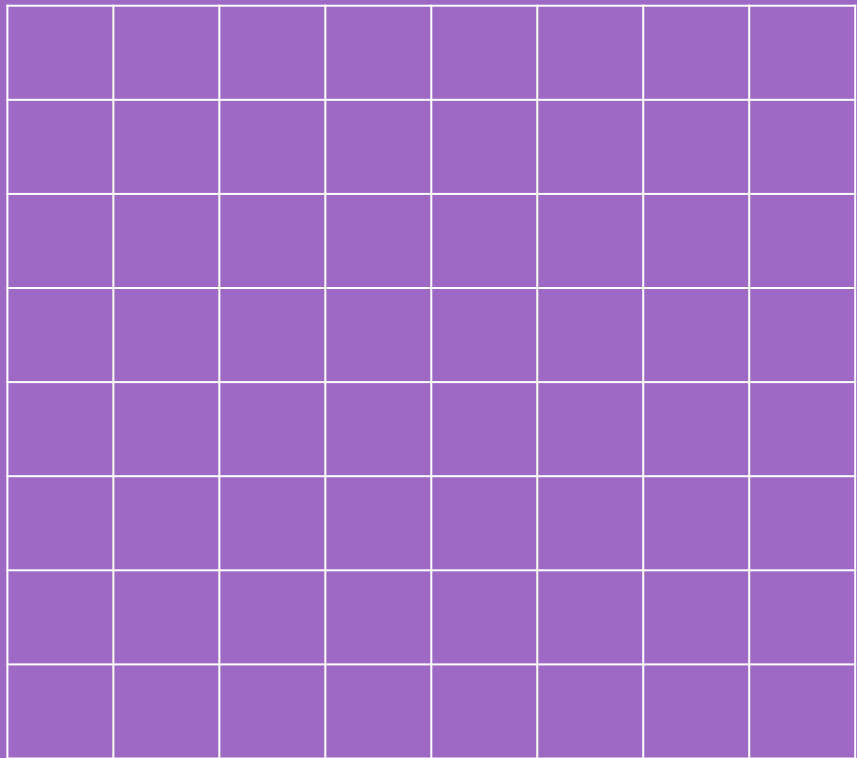
Console Output:

```
EjemploConsola [Java Application] /opt/eclipse/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.linux.x86_64_17.0.2.v20220201-1208/jre/bin/java (Jan 2, 2023, 9:31:25 PM) [pid: 232166]
ingrese un texto
un texto
```

Se pueden correr  
varios **procesos**  
del mismo código  
al mismo tiempo

- 1 EjemploConsola [Java Application] /opt/eclipse/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.linux.x86\_64\_17.0.2.v20220201-1208/jre/bin/java (Jan 2, 2023, 9:31:25 PM) [pid: 232166]
- 2 EjemploConsola [Java Application] /opt/eclipse/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.linux.x86\_64\_17.0.2.v20220201-1208/jre/bin/java (Jan 2, 2023, 9:31:32 PM) [pid: 232219]
- 3 EjemploConsola [Java Application] /opt/eclipse/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.linux.x86\_64\_17.0.2.v20220201-1208/jre/bin/java (Jan 2, 2023, 9:32:03 PM) [pid: 232502]
- 4 EjemploConsola [Java Application] /opt/eclipse/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.linux.x86\_64\_17.0.2.v20220201-1208/jre/bin/java (Jan 2, 2023, 9:32:58 PM) [pid: 232862]

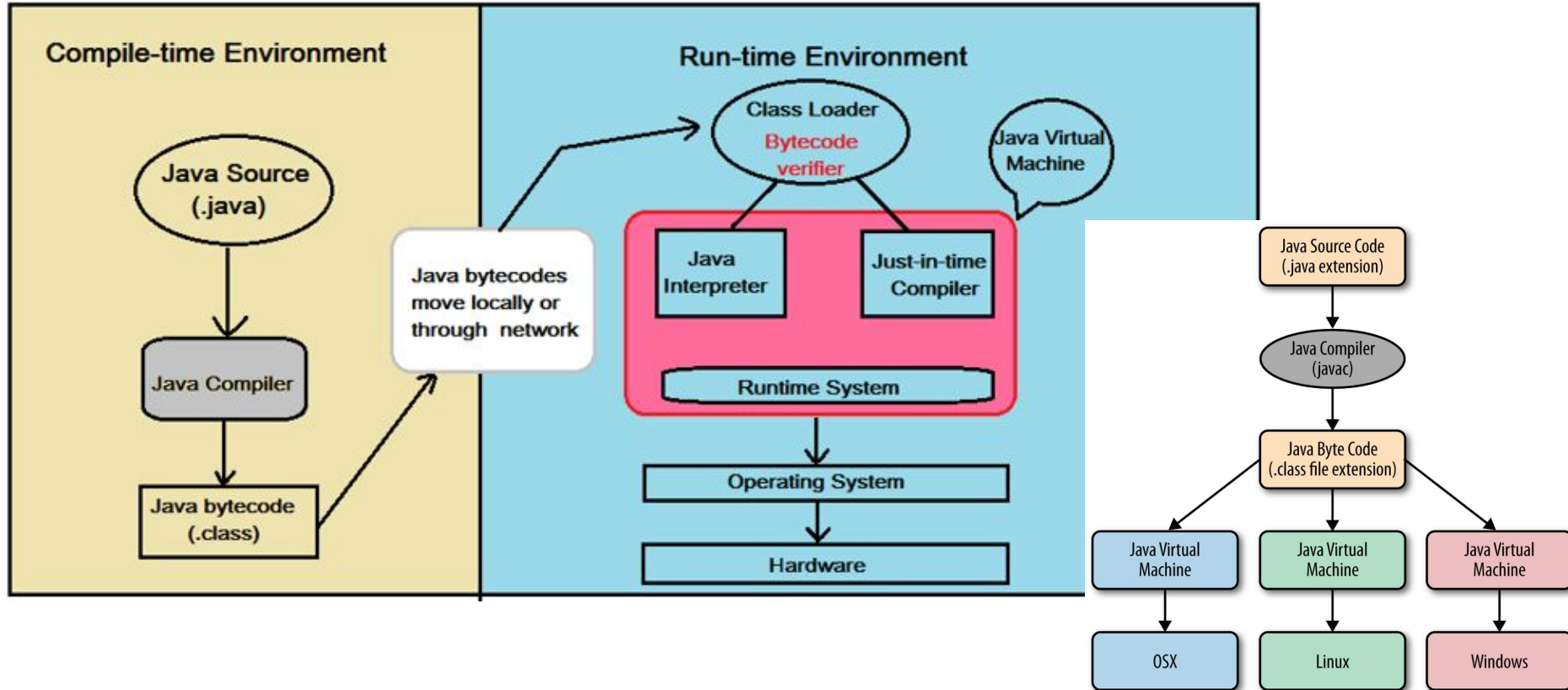
# JAVA



# Java

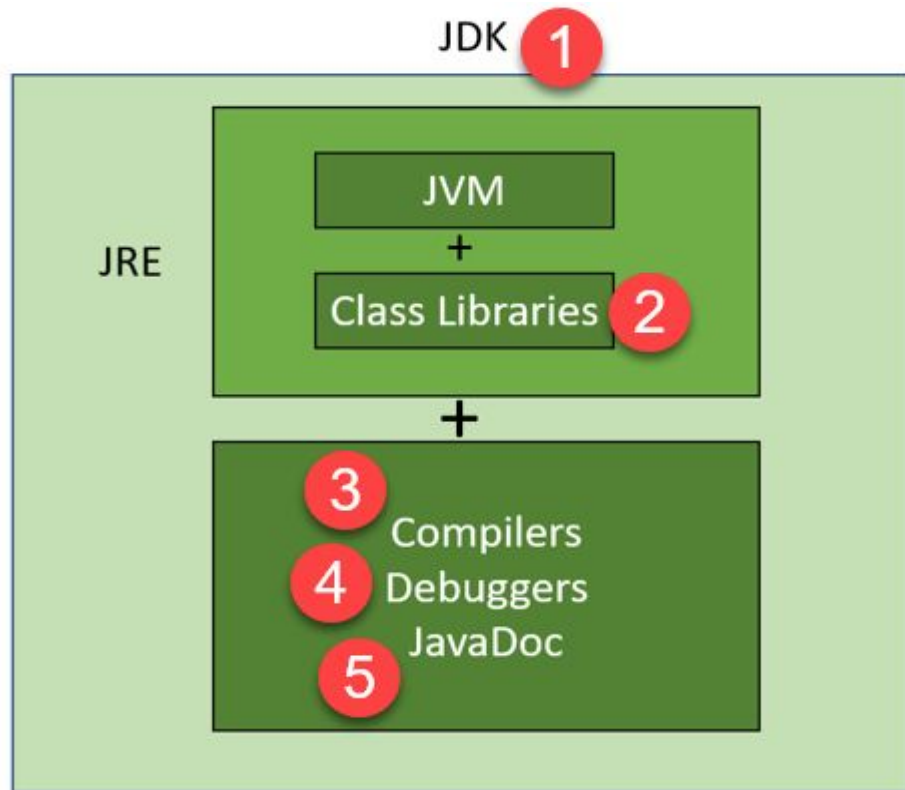
- Lenguaje de desarrollo de propósito general
- Portable: Funciona en todos los sistemas operativos
- Gran Comunidad y base de un gran número de proyectos
- Maduro
- Orientado a Objetos
- Compilado

# JVM y compilación

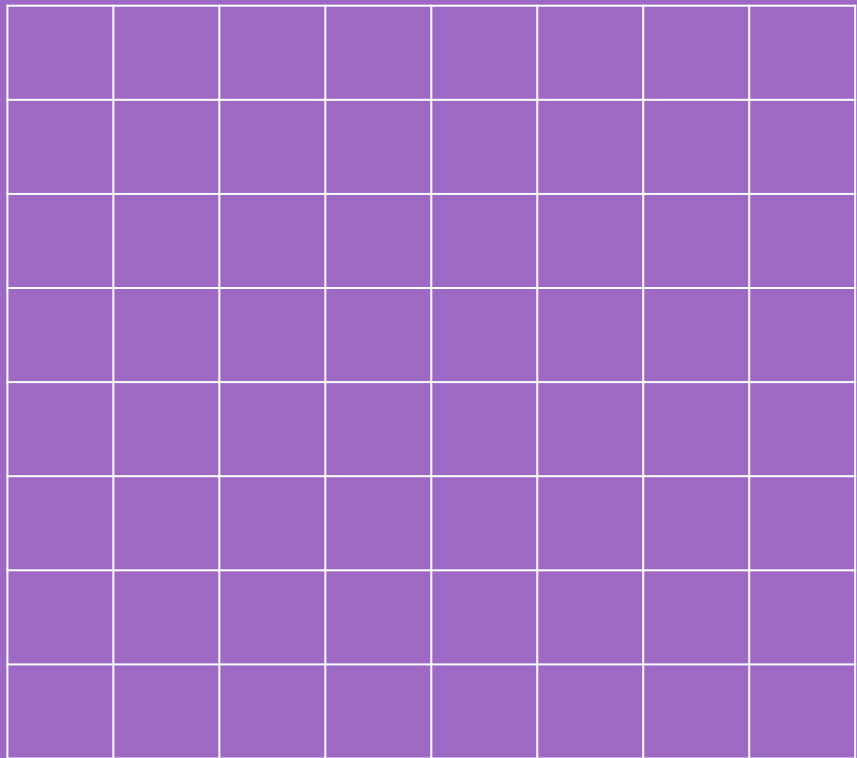




# Java - JVM - JDK - JRE



# ELEMENTOS BÁSICOS JAVA



# Elementos básicos

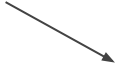
- Sintaxis.
- Tipos de datos.
- Variables.
- Operadores.
- Funciones de entrada y salida.

# Java - Sintaxis Básica

- Conjunto de reglas que deben seguirse al escribir el código fuente:
  - Comentarios: *//en línea, /\*Múltiples líneas\*/.*
  - Palabras reservadas: *palabras especiales del lenguaje.*
  - Todo programa se escribe dentro de una clase, el nombre de la clase debe coincidir con el nombre del archivo java, el nombre de clase con mayúsculas.
  - Uso de {} para delimitar bloques
  - Uso de ; para delimitar o finalizar sentencias
  - Identificadores propios de nombres: *comienzan con una letra, case sensitive, notación camelCase, no palabras reservadas.*

```
1  public class Programa
2  {
3      public static void main(String[] args)
4      {
5          System.out.println("Hola Mundo");
6      }
7  }
```

método especial que se reconoce como punto de inicio de un programa



# Tipos de datos

Tipo de Datos	Valor	Rango de Valores
boolean	Lógico (true o false)	1 byte
int	Entero	4 bytes, desde $-2^{15}$ a $+2^{15}-1$
long	Entero largo	8 bytes, desde $-2^{31}$ a $+2^{31}-1$
float	Real	4 bytes Precisión aprox. 7-8 dígitos
double	Real largo	8 bytes Precisión aprox. 18-19 dígitos
char	Caracter (Entre comillas simples)	2 bytes sin signo – Codificación Unicode en 32 bits
String (clase)	Cadena de caracteres (Entre comillas dobles)	Depende del largo de la cadena

```
boolean unValorBooleano = true;  
int miPrimerContador = 66;  
double unValor = 1.68;  
float otroNum = 2.344f;  
char unaLetra = 'a';  
String unNombre = "Juan";
```

# Variables

- Es un grupo de bytes asociado a un *nombre* o *identificador* y a un *tipo de dato*, tiene una *dirección* (ubicación) dentro de la memoria y puede *almacenar información* según el tipo de dato que tenga asociado.

## Asignación de Variables:

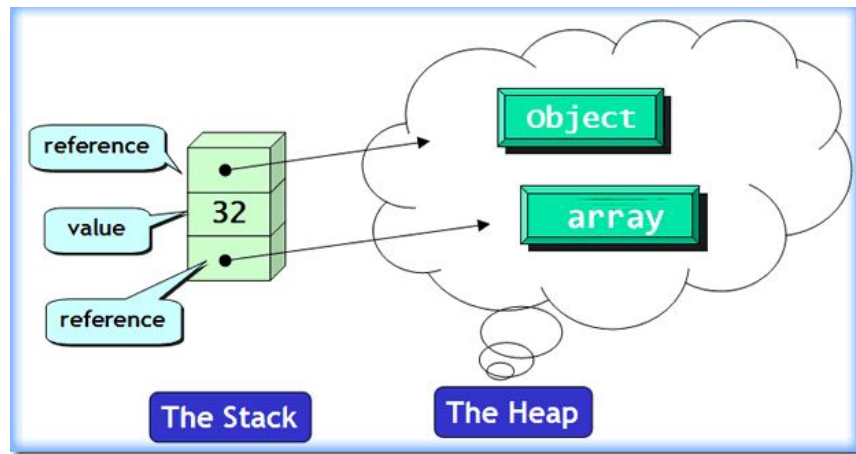
- Consiste en reemplazar el valor anterior o inicial de la variable con un valor nuevo.
- El operador de asignación es el signo “=”.

//Asigna el valor 32 a la variable num

```
int num = 32;
```

//Asigna el valor 7 a la variable num,  
esta pierde el valor anterior de 32, no hay  
que volver a definirla.

```
num = 7;
```



# Expresiones



- Fórmula que combina **operandos** (valores constantes o variables) y **operadores** (símbolos de operaciones suma, resta, etc.).
- Ejemplo de expresión aritmética, con asignación:

```
int a = 3;  
int b = 5;  
int c;  
int c = a + 2 * b;
```

# Operadores Aritméticos

Operador	Significado	Ejemplo de uso
+	suma	$a = b + c$
-	resta	$a = b - c$
*	producto	$a = b * c$
/	división de coma flotante	$a = b / c$
//	división entera	$a = b // c$
%	resto de una división	$a = b \% c$
**	potencia	$a = b ** c$

```
int miPrimerContador = 66;
```

```
double unValor = 1.68;
```

```
miPrimerContador + 20
```

```
15 - 12
```

```
10 * 3
```

```
unValor / 3
```

```
8 % 3
```



# Operadores Relacionales

Opera dor	Significado	Ejempl o	Observaciones
==	igual que	$a == b$	retorna <i>True</i> si <i>a</i> es igual que <i>b</i> , o <i>False</i> en caso contrario
!=	distinto de	$a != b$	retorna <i>True</i> si <i>a</i> es distinto de <i>b</i> , o <i>False</i> en caso contrario
>	mayor que	$a > b$	retorna <i>True</i> si <i>a</i> es mayor que <i>b</i> , o <i>False</i> en caso contrario
<	menor que	$a < b$	retorna <i>True</i> si <i>a</i> es menor que <i>b</i> , o <i>False</i> en caso contrario
>=	mayor o igual que	$a >= b$	retorna <i>True</i> si <i>a</i> es mayor o igual que <i>b</i> , o <i>False</i> en caso contrario
<=	menor o igual que	$a <= b$	retorna <i>True</i> si <i>a</i> es menor o igual que <i>b</i> , o <i>False</i> en caso contrario

```
int miPrimerContador = 66;  
double unValor = 1.68;  
double otroValor = 1.67;
```

```
unValor == (otroValor + 0.01)
```

# Operadores lógicos

Operador	Significado
<b>&amp;&amp;</b>	conjunción lógica (y)
<b>  </b>	disyunción lógica (o)
<b>!</b>	negación lógica (not)

```
boolean unBooleano = true;  
boolean otroBooleano = false;
```

```
!unBooleano // false  
unBooleano && otroBooleano // false  
unBooleano || otroBooleano // true
```

```
unBooleano && (otroBooleano || True) //  
true
```

# Salida Básica - System Out



Todos los lenguajes de programación tienen una forma de enviar información al usuario, ya sea mediante ventanas o lo que llamamos la “consola”. En este curso, utilizaremos mucho mostrar contenidos de variables e información usando el comando:

```
System.out.println(...)
```

```
//Ejemplos
```

```
System.out.println(1) ;
```

```
System.out.println('a') ;
```

```
System.out.println(true) ;
```

```
int x = 14;
```

```
System.out.println(x);
```

Todavía falta para que comprendamos por qué hay puntos o que es “System”, pero rápidamente, hay algo denominado “Clase”, que representa conceptos de distinto tipo y ofrece funcionalidades varias.

Por ahora simplemente lo usaremos y en unas pocas clases entenderemos la estructura de lo que estamos usando.

# Entrada por consola



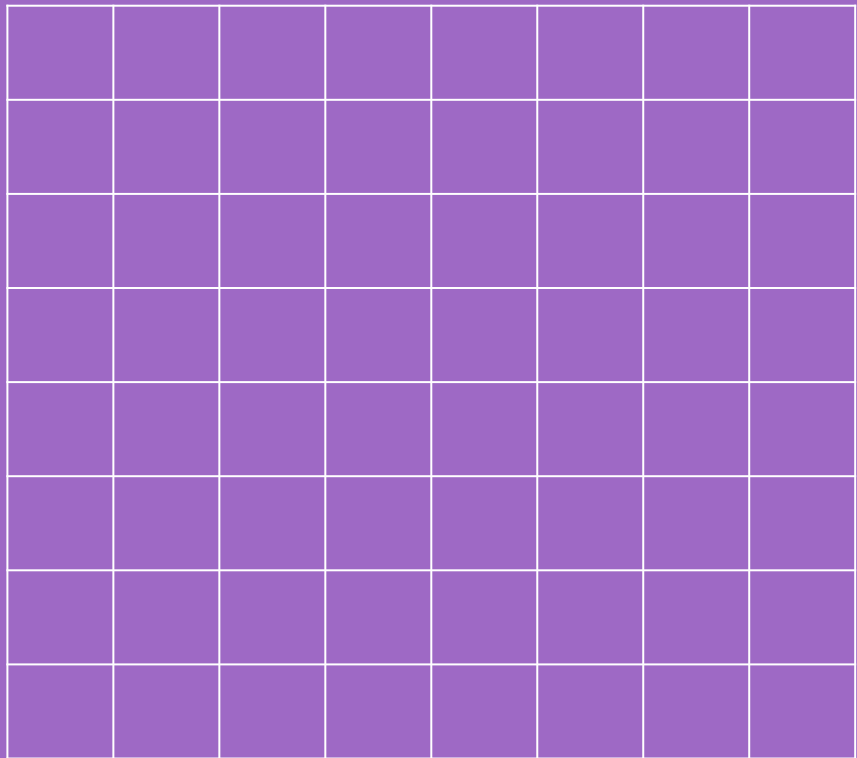
```
import java.util.Scanner;

public class Ejercicio1 {

    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        // Entrada de datos numéricos
        int edad = scn.nextInt();
        long celular = scn.nextLong();
        float promedio = scn.nextFloat();
        double precio = scn.nextDouble();
        // Entrada de una cadena
        String nombre = scn.nextLine();
        // entrada de un carácter
        char letra = scn.next().charAt(0);
    }
}
```

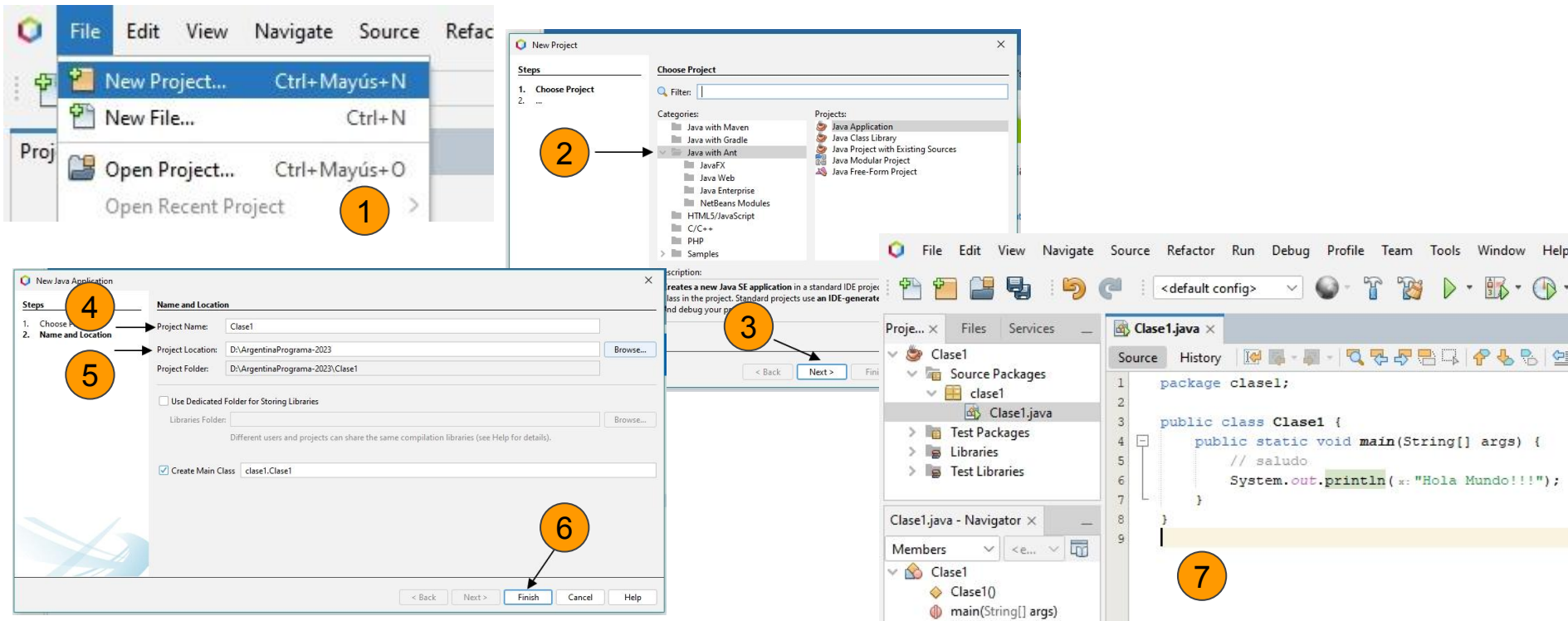
Una forma de ingresar datos a un programa es mediante la consola, en este ejemplo el programa nos pedirá que escribamos algo y apretemos la tecla “enter” para terminar, guardando el resultado en la variable indicada.

# EJERCICIOS



# Ejercicios

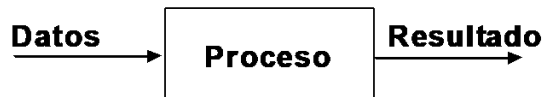
## Abrir un proyecto en NetBeans y ejecutar un “Hola Mundo”



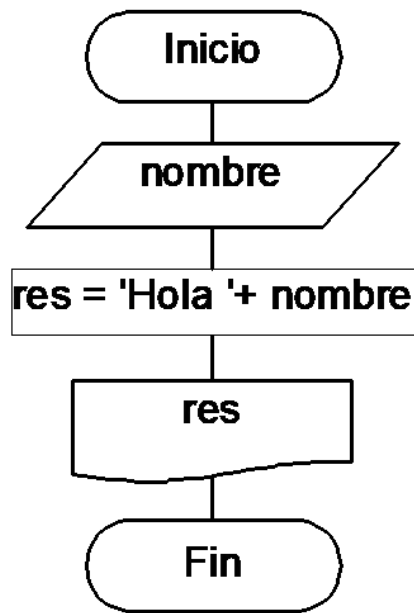
The following steps are illustrated in the screenshots:

1. Open the **File** menu and select **New Project...** (Ctrl+Mayús+N).
2. In the **Choose Project** dialog, select **Java with Ant** under the **Categories** list.
3. In the **New Java Application** wizard, click **Next >** to proceed to the next step.
4. In the **Name and Location** step, enter **Clase1** as the **Project Name**.
5. Select the **Project Location** (e.g., **D:\ArgentinaPrograma-2023**).
6. Click **Finish** to create the project.
7. Run the application. The output in the console is: `System.out.println( "Hola Mundo!!!" );`

# Ejercicios



1. Ingresar un nombre por teclado y mostrarlo por pantalla.



## Pseudocódigo:

- 1-Cargar **nombre**: nombre del usuario
- 2-Armar cadena de resultado: 'Hola' y **nombre**
- 3-Mostrar el resultado **res**

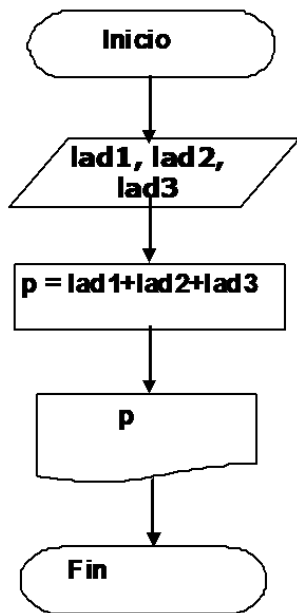
## Código Java

```
5 package clasel;
6 import java.util.Scanner;
7
8 public class Ejercicio1_1 {
9     public static void main(String[] args) {
10         Scanner sc = new Scanner(System.in);
11         // Solicita cargar un nombre
12         System.out.print("Ingrese un nombre: ");
13         // Entrada de una cadena
14         String nombre = sc.nextLine();
15         // Muestra el saludo con el nombre
16         System.out.println("Hola " + nombre);
17     }
18 }
```

# Ejercicios



2. Ingresar el valor de los tres lados de un triángulo, calcular y mostrar su perímetro.



```
7 import java.util.Scanner;
8
9 public class Ejercicio1_2 {
10     public static void main(String[] args) {
11         // declaración de variables
12         float lad1, lad2, lad3;
13         float perimetro;
14
15         Scanner sc = new Scanner( source: System.in);
16         // Ingreso de datos
17         System.out.print( ": "Longitud del primer lado: ");
18         lad1 = sc.nextFloat();
19         System.out.print( ": "Longitud del primer lado: ");
20         lad2 = sc.nextFloat();
21         System.out.print( ": "Longitud del primer lado: ");
22         lad3 = sc.nextFloat();
23
24         //Proceso
25         perimetro = lad1 + lad2 + lad3;
26
27         // Muestra el resultado
28         System.out.println("El perimetro es: " + perimetro);
29     }
30 }
```



# Ejercicios



3. Ingresar el valor de tres temperaturas, calcular y mostrar el promedio de las temperaturas.

```
4  import java.util.Scanner;
5
6  public class Ejercicio1_3 {
7      public static void main(String[] args) {
8          // declaración de variables
9          int temp1, temp2, temp3;
10         int suma;
11         float promedio;
12
13         Scanner sc = new Scanner(System.in);
14         // Ingreso de datos
15         System.out.print("Temperatura 1: ");
16         temp1 = sc.nextInt();
17         System.out.print("Temperatura 2: ");
18         temp2 = sc.nextInt();
19         System.out.print("Temperatura 3: ");
20         temp3 = sc.nextInt();
21
22         //Procesos
23         suma = temp1 + temp2 + temp3;
24         promedio = suma / 3;
25
26         // Muestra el resultado
27         System.out.println("El promedio de temperaturas es: " + promedio);
28     }
29 }
```



**Argentina  
programa  
4.0**

# Gracias!

---