



El futuro digital  
es de todos

MinTIC



Vigilada Mineducación

# CICLO II: Programación Básica en Java

Misión  
TIC 2022





El futuro digital  
es de todos

MinTIC



Vigilada Mineducación

# Sesión 1: Introducción a Java

VARIABLES, OPERADORES Y PRECEDENCIA  
EXPRESIONES Y ALGORITMOS SECUENCIALES





# Objetivos de la sesión

Al finalizar esta sesión estarás en capacidad de:

1. Determinar las componentes mínimas de un programa escrito en Java.
2. Identificar y comprender los potenciales errores iniciales de un programa escrito en Java.
3. Describir y aplicar el método print y println.
4. Diferenciar Tipos de datos
5. Declarar variables
6. Manejo de Operadores aritméticos, relacionales y lógicos
7. Precedencia de operadores
8. Expresiones
9. Asignación
10. Manipular Tipos de datos Strings y Character.
11. Convertir números a strings y de strings a números para entrada.



# Objetivos de la sesión

Al finalizar esta sesión estarás en capacidad de:

12. Describir y aplicar el método Scanner.
13. Describir los operadores relacionales
14. Diseñar y desarrollar programas que incluyan condicionales If-else, if-else if-else simples y anidados.
15. Diseñar y desarrollar programas que incluyan el manejo de switch.
16. Diseñar y desarrollar programas que incluyan ciclo repetitivo while
17. Diseñar y desarrollar programas que incluyan ciclo repetitivo do-while
18. Operadores de asignación compuesto (operador=)
19. Operadores de incremento y decremento(++/--)
20. Diseñar y desarrollar programas que incluyan ciclo repetitivo for



El futuro digital  
es de todos

MinTIC

**UN** UNIVERSIDAD  
DEL NORTE

Vigilada Mineducación

# INTRODUCCIÓN A LA PROGRAMACIÓN

Misión  
TIC2022



# Variables

Una variable en programación es una unidad de datos que puede cambiar de valor. Es la forma más simple de almacenamiento, representando una zona de memoria donde se almacena un elemento de datos.

## Para recordar:

Una variable siempre debe estar identificada por un nombre y definida por un tipo.

## En Pseudocódigo

Tipos de variables:

1. Entero
2. Real
3. Carácter
4. Lógica

## Y En Java...

1. Entero □ **int**
2. Real □ **float, double**
3. Carácter □ **char**  
Cadena de caracteres □ **String**
4. Lógica □ **boolean**



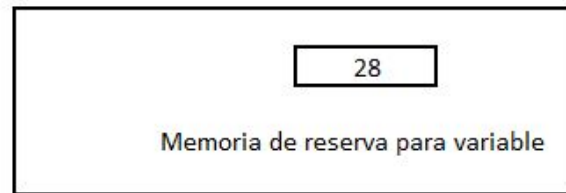
# Declaración de variables

- **tipo de dato:** tipo de datos que se pueden almacenar en esta variable.
- **nombre\_variable:** nombre dado a la variable.
- **valor:** es el valor inicial almacenado en la variable.

`int age = 28;`

Diagram illustrating the components of the variable declaration `int age = 28;`:

- tipo de dato:** Points to `int`.
- nombre\_variable:** Points to `age`.
- valor:** Points to `28`.



RAM

## Ejemplos:

```
float simpleInterest; //Declarando variable float
int time = 10, speed = 20; //Declarando e Inicializando la variable
integer
char var = 'h'; // Declarando e Inicializando la variable character
```



# Convertir String a Número o Número a String

String (cadena) a int (entero):

```
String enteroString = "5";  
int entero = Integer.parseInt(enteroString);
```

int(entero) a String:

```
int entero = 5;  
String enteroString = Integer.toString(entero);
```

String a double:

```
String DoubleString = "8342342";  
double aDouble = Double.parseDouble(aString);
```

double a String:

```
double d = 8342342;  
System.out.println(Double.toString(d));
```





El futuro digital  
es de todos

MinTIC



Vigilada Mineducación

# EXPRESIONES Y OPERADORES





# Operación de Asignación

## Recordemos:

Esta operación permite dar valor a variables.

Representación algorítmica:  $\leftarrow$

## Lo nuevo:

En java la operación de asignación se representa con el signo =

### Pseudocódigo

```
A  $\leftarrow$  5  
B  $\leftarrow$  A+2  
A  $\leftarrow$  7
```

### Java

```
A = 5  
B = A+2  
A = 7
```



# Expresiones - Recordemos

$$2*c^3+b*a*\sqrt{3*a+b^2}$$

- Una expresión consta de operadores y operandos.
- Según sea el tipo de datos que manipulan, se clasifican en:
  - Aritméticas
  - Relacionales
  - Lógicas



# Operadores Aritméticos

Operador	Nombre	Ejemplo	Java
+	Suma	$A + 4$	+
-	Resta	$A - 4$	-
*	Multiplicación	$B * 2$	*
/	División	$C / 3$	/
Mod	Modulo (residuo de la división entera)	$15 \bmod 2 = 1$	$15 \% 2$
=Div	Cociente de la división entera	$15 \text{ div } 2 = 7$	En Java no existe, se aplica el axioma de cerradura entre los números enteros: $15 / 2 = 7$
^	Potencia o	$B^3$	Math.pow(base,potencia) Math.pow(B,3)



# Operadores Relacionales

Operador	Nombre	Ejemplo	Java
>	Mayor que	4>2	>
<	Menor que	3<10	<
>=	Mayor o igual	5>=5	>=
<=	Menor o igual	7<=9	<=
=	Igual	3=2	==
<>	Diferente	9<>7	!=



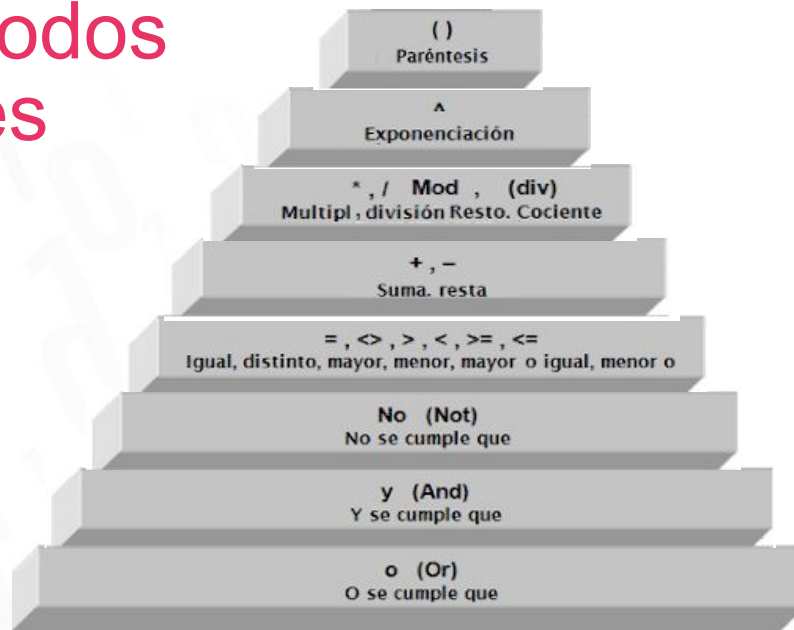
# Operadores Lógicos

Operador	Nombre	Ejemplo	Java
And, &	Y	$(3 > 5) \& (4 < 10)$	&&
Or, v,	O	$(3 < 5)   (4 < 10)$	
NO	Negación	$\sim(6 = 6)$	!

P	Q	$P \vee Q$	$P \wedge Q$
V	V	V	V
V	F	V	F
F	V	V	F
F	F	F	F



# Prioridad de todos los operadores





# Estructura de Salida - *Escritura*

## Pseudocódigo:

Escribir “El resultado es:”, R

Donde “El resultado es:” es un mensaje que se desea   aparezca y R es la variable que contiene un valor.

## Java:

- `System.out.print("El resultado es: "+ R);`(Muestra el mensaje entre paréntesis y deja el cursor en la misma línea después de mostrarlo)
- `System.out.println("El resultado es: "+ R);` (muestra el mensaje entre paréntesis y posiciona el cursor en una nueva línea)





# Estructuras de Entrada - *Lectura*

## Pseudocódigo:

Leer a, b;

Donde “a” y “b” son las variables tipo entero que recibirán los valores.

## Java:

Se debe importar la librería `java.util.Scanner`; luego se define una variable que permitirá realizar la captura de los datos por teclado.

```
Scanner leer = new Scanner(System.in);
```

Dependiendo del tipo de dato, estos se leen utilizando `nextTipo()` (donde Tipo puede ser: `Int`, `Float`, `Double`, `Line`) de la siguiente forma:

```
a=leer.nextInt();  
b=leer.nextInt();
```



# Errores comunes

1. Olvidar un punto y coma al final de una instrucción o sentencia.
2. No cerrar llaves de algún bloque de código, método, clase o en alguna estructura de control.
3. Colocar el mismo nombre a variables con diferente tipo.
4. Asignar a una variable otra con tipo de dato diferente: por ejemplo: una variable String y se le asigna una variable de tipo int, en este caso el compilador arroja un error de conversión de tipos.
5. Ingreso de valores diferentes a los que la aplicación recibe.
6. Acceder a una posición que no existe en un arreglo.
7. Almacenar cadenas donde se debe almacenar números.
8. Divisiones por cero.



El futuro digital  
es de todos

MinTIC

**UN** UNIVERSIDAD  
DEL NORTE

Vigilada Mineducación

# ESTRUCTURAS ALGORÍTMICAS CONDICIONALES

Mision  
TIC2022



# Estructuras Condicionales Simples

## PSEUDOCÓDIGO

---

Si (condición) entonces

Acción(es)

Fin-si

## JAVA

---

if (condición){

acciones

}



# Estructuras Condicionales Dobles:

## PSEUDOCÓDIGO

---

Si (condición) entonces

Acción(es)\_1

si no

Acción(es)\_2

Fin-si

## JAVA

---

```
if (condición){
```

```
    acciones 1
```

```
}else{
```

```
    acciones 2
```

```
}
```



# Estructuras Condicionales Múltiples

## PSEUDOCÓDIGO

```
Si (condición_1) entonces
    Acción(es)_1
sino
    Si (condición_2) entonces
        Acción(es)_2
    sino
        } Varias condiciones
fini
fini
```

## JAVA

```
if (condición_1) {
    Acción(es)_1
} else if (condición_2) {
    Acción(es)_2
} else if (condición_3) {
    Acción(es)_3
} else {
    } Varias condiciones
}
```



# Estructuras Condicionales Anidados

## PSEUDOCÓDIGO

```
Si (condición_1) entonces
  Si (condición_2) entonces
    Acción(es)_1
  si no
    Acción(es)_2
  Fin-si
sino
  } Varias condiciones
finsi
```

## JAVA

```
if (condición_1) {
  if (condición_2) {
    Accion(es)_1
  } else {
    Accion(es)_2
  }
} else {
  } Varias condiciones
}
```



El futuro digital  
es de todos

MinTIC

**UN** UNIVERSIDAD  
DEL NORTE

Vigilada Mineducación

# ESTRUCTURA ALGORÍTMICA DEPENDIENDO DE

Mision  
TIC2022





# Dependiendo De (DD) o Según

## Pseudocódigo

Dependiendo_De [Expresión] Hacer	Según [Expresión] Hacer
Caso 1: Instrucción 1 Instrucción 2	Caso 1: Instrucción 1 Instrucción 2
Caso 2: Instrucción 3 Instrucción 4 . . .	Caso 2: Instrucción 3 Instrucción 4 . . .
Caso n: Instrucción k SiNo Instrucción m	Caso n: Instrucción k SiNo Instrucción m
<b>Fin Dependiendo_De</b>	<b>FinSegún</b>

## Java

```
switch (Expresion) {  
    case 1:  
        instrucciones 1;  
        break;  
    case 2:  
        instrucciones 2;  
        break;  
    .  
    .  
    .  
    case n:  
        instrucciones k;  
        break;  
    default:  
        instrucción m;  
}
```



El futuro digital  
es de todos

MinTIC

**UN** UNIVERSIDAD  
DEL NORTE

Vigilada Mineducación

# ESTRUCTURAS ALGORÍTMICAS CÍCLICAS O REPETITIVAS

Mision  
TIC2022



# Estructuras cíclicas

Ciclos con un Número Indeterminado de Iteraciones  
(Mientras Que y Hacer Hasta o repetir hasta)

Mientras (condición) hacer

Accion1

Accion2

.

.

AccionN

Fin-mientras

Hacer

Accion1

Accion2

.

.

.

AccionN

Hasta (condición)

Ciclo con un Número Determinado de  
Iteraciones (Para)

para c  $\square$  VI hasta VF con inc I hacer

Accion1

Accion2

.

.

.

AccionN

Fin-para

Dónde:  
c: variable control  
VI: Valor Inicial  
VF: Valor Final  
I: Incremento



# Estructuras cíclicas en Java

## Mientras Que

```
while (condicion) {  
    Acción 1  
    Acción 2  
    :  
    :  
}
```

## Hacer Mientras

```
do{  
    Acción 1  
    Acción 2  
    :  
    :  
}while (condición);
```

## Para

```
for (c= vi; c<= vf; c=c+i) {  
    Acción 1  
    Acción 2  
    :  
    :  
}
```



# Operadores de asignación compuestos

Operador	Descripción	Ejemplo de expresión	Resultado del ejemplo
<b>+=</b>	Suma combinada	$a+=b$	$a=a+b$
<b>-=</b>	Resta combinada	$a-=b$	$a=a-b$
<b>*=</b>	Producto combinado	$a*=b$	$a=a*b$
<b>/=</b>	División combinada	$a/=b$	$a=a/b$
<b>%=</b>	Resto combinado	$a\%=b$	$a=a\%b$



# Operadores de incremento y decremento

Operador	Descripción	Ejemplo de expresión	Resultado del ejemplo
<b>++</b>	<b>Incremento</b> i++ primero se utiliza la variable y luego se incrementa su valor ++i primero se incrementa el valor de la variable y luego se utiliza	4++ a=5; b=a++; a=5; b=++a;	5  a vale 6 y b vale 5  a vale 6 y b vale 6
<b>--</b>	<b>decremento</b>	4--	3



## Ejercicio

**Diseñar un algoritmo que calcule el promedio de notas del primer parcial de un curso de  $N$  estudiantes.**



# Solución

## Con Mientras Que

Inicio  
  real not, prom, sum,  
  entero c, n  
  leer n  
  sum  $\leftarrow$  0  
  c  $\leftarrow$  1  
  MQ (c  $\leq$  n) hacer  
    leer not  
    sum  $\leftarrow$  sum + not  
    c  $\leftarrow$  c + 1  
  Fin MQ  
  prom  $\leftarrow$  sum / n  
  Escribir prom  
Fin

## Con Hacer Mientras

Inicio  
  real not, prom, sum,  
  entero c, n  
  leer n  
  sum  $\leftarrow$  0  
  c  $\leftarrow$  1  
  Hacer  
    leer not  
    sum  $\leftarrow$  sum + not  
    c  $\leftarrow$  c + 1  
  Mientras (c > n)  
    prom  $\leftarrow$  sum / n  
    Escribir prom  
Fin

## Con Para

Inicio  
  real not, prom, sum,  
  entero c, n  
  leer n  
  sum  $\leftarrow$  0  
  para c: 1 hasta n inc 1 hacer  
    leer not  
    sum  $\leftarrow$  sum + not  
  Fin para  
  prom  $\leftarrow$  sum / n  
  Escribir prom  
Fin







El futuro digital  
es de todos

MinTIC



Vigilada Mineducación

# Codificación en Java de las estructuras algorítmicas.





## Ejemplo If:

```
int i = 10;
if (i > 15)
    System.out.println("10 es menor que 15");
// Esta declaración se ejecutará
// como si considerara una declaración por
defecto
System.out.println("No estoy en IF");
```

**Salida** No estoy en IF

## Ejemplo If-else:

package practica;

```
int i = 20;
if (i < 15)
    System.out.println("Soy menor que 15");
else
    System.out.println("Soy mayor de 15");
```

**Salida** Soy mayor de 15



## Ejemplo If anidado:

package practica;

```
int i = 10;
if (i == 10)
{
    // Primera declaración if
    if (i < 15)
        System.out.println("soy menor que 15");
    // Declaración if anidada
    // Solo se ejecutará si la instrucción anterior
    // es verdad
    if (i < 12)
        System.out.println("yo también soy menor que 12");
    else
        System.out.println("soy mayor que 15");
}
```

Salida soy menor que 15

yo soy menor que 12



## Ejemplo switch-case:

```
int i = 9;  
switch (i)  
{  
  case 0:  
    System.out.println("i es cero.");  
    break;  
  case 1:  
    System.out.println("i es uno.");  
    break;  
  case 2:  
    System.out.println("i es dos.");  
    break;  
  default:  
    System.out.println("i es mayor que 2.");  
}
```

Salida i es mayor que 2.



## Ejemplo while-if-for:

```
Scanner input = new Scanner(System.in);
int magic_square[][]=new int[3][3];
int counter = 0;
while(true){
    System.out.println("Enter the Magic Number: ");
    for(int i = 0; i < 3; i++)
    {
        for(int j = 0; j < 3; j++)
        {
            System.out.println("Number "+(i+1)*(j+1));
            magic_square[i][j] = input.nextInt();
            if (magic_square[i][j] < 1 || magic_square[i][j] > 9)
                System.out.println("ERROR! Input range is invalid.");
            else
                counter++;
        }
    }
}
```

Salida i es mayor que 2.



El futuro digital  
es de todos

MinTIC



Vigilada Mineducación

# Ejercicios para practicar





El futuro digital  
es de todos

MinTIC

**UN** UNIVERSIDAD  
**DEL NORTE**

Vigilada Mineducación

**¡GRACIAS**  
**POR SER PARTE DE**  
**ESTA EXPERIENCIA**  
**DE APRENDIZAJE!**



**Misión**  
**TIC 2022**