



El empleo
es de todos

Mintrabajo

Clase 1 java



@SENAcomunica

www.sena.edu.co

1 Instalación de JDK



1. visite el siguiente enlace para acceder al JDK (Java Development Kit)

<https://www.oracle.com/java/technologies/javase-downloads.html>

Java SE 16

Java SE 16 is the latest release for the Java SE Platform

- [Documentation](#)
- [Installation Instructions](#)
- [Release Notes](#)
- [Oracle License](#)
 - [Binary License](#)
 - [Documentation License](#)
- [Java SE Licensing Information User Manual](#)
 - [Includes Third Party Licenses](#)
- [Certified System Configurations](#)
- [Readme](#)

Java Platform, Standard Edition

Dar clic aquí y luego seleccione el sistema operativo que usa.

Oracle JDK

↓ JDK Download

↓ Documentation Download

You must accept the [Oracle Technology Network License Agreement for Oracle Java SE](#) to download this software.

☒ I reviewed and accept the Oracle Technology Network License Agreement for Oracle Java SE

Download jdk-16_windows-x64_bin.zip ↓

Windows x64 Compressed Archive

168.74 MB

Archivo comprimido

↓ jdk-16_windows-x64_bin.zip

El archivo que se descarga, debe ponerlo en una carpeta "java", o en archivos del programa en caso de trabajar con sistema operativo Windows.



C:\Program Files\Java\jdk-16\bin

▼ hoy (1)

jdk-16_windows-x64_bin

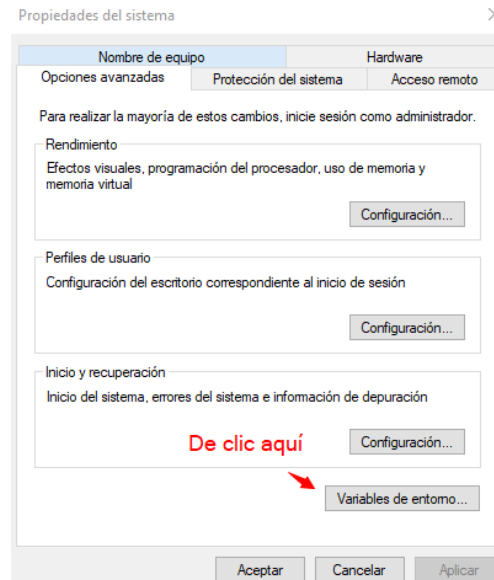
Configura las variables de entorno.

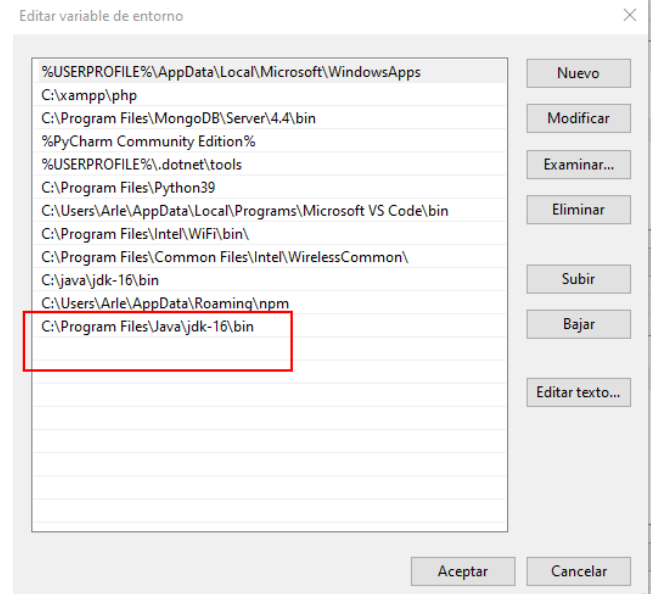
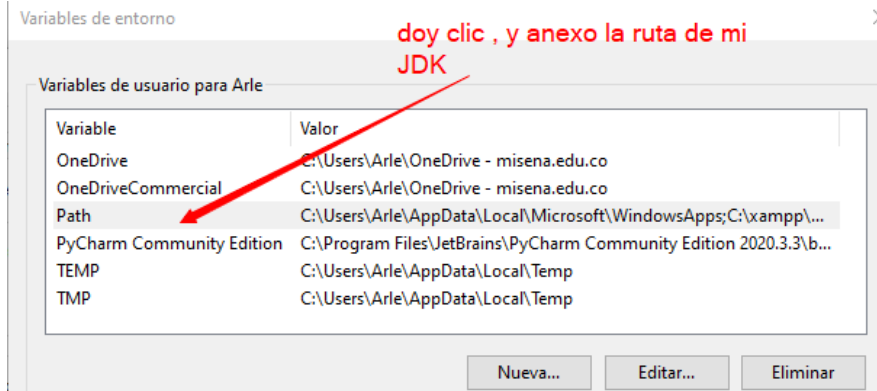


Editar las variables de entorno del sistema

Panel de control

 Abrir





En el CMD, podemos validar si quedo correctamente instalado.

```
C:\Users\Arle>java --version
java 16 2021-03-16
Java(TM) SE Runtime Environment (build 16+36-2231)
Java HotSpot(TM) 64-Bit Server VM (build 16+36-2231, mixed mode, sharing)

C:\Users\Arle>javac --version
javac 16
```



<https://www.guru99.com/best-programming-language.html>

LANGUAGES



PYTHON



JAVA



C



PHP



C++



JAVASCRIPT



C#



RUBY



OBJECTIVE-C

2 VERSIONES DE JSE JAVA



Java SE 16

Java SE 16 is the latest release for the Java SE Platform

Java SE 15

Java SE 15.0.2 is the latest release for the Java SE 15 Platform

Java SE 11 (LTS)

Java SE 11.0.10 is the latest release for the Java SE 11 Platform

Java SE 8

Java SE 8u281 is the latest release for the Java SE 8 Platform.

Java SE 7

Updates for Java SE 7 released after April 2015 are only available to Oracle Customers through [My Oracle Support](#) (requires support login).

3 MAQUINA VIRTUAL DE JAVA



El archivo se compila con el comando `javac`, el cual genera un código conocido como Bytecode. Este archivo generado tiene la extensión `.class` y es el que podemos utilizar para ejecutar nuestro programa, el cual puede ser ejecutado en cualquier ambiente en el que este instalada la maquina virtual de java.

4 AMBIENTE DE EJECUCION DE JAVA



Si utilizamos un IDE como Netbeans, al guardar nuestro archivo estamos haciendo el proceso de compilación, por eso será muy transparente el proceso



Para descargar IntelliJ IDEA

<https://www.jetbrains.com/es-es/idea/download/#section=windows>

Descargar IntelliJ IDEA

Windows

macOS

Linux

Ultimate

Para desarrollo web y empresarial

Descargar

.exe ▼

Prueba gratis de 30 días

Community

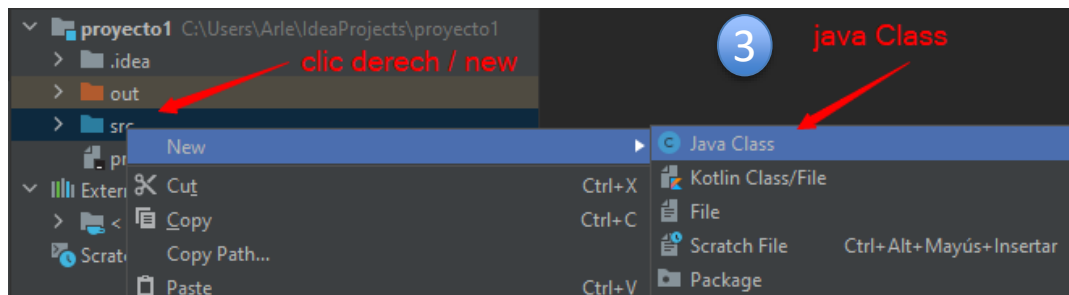
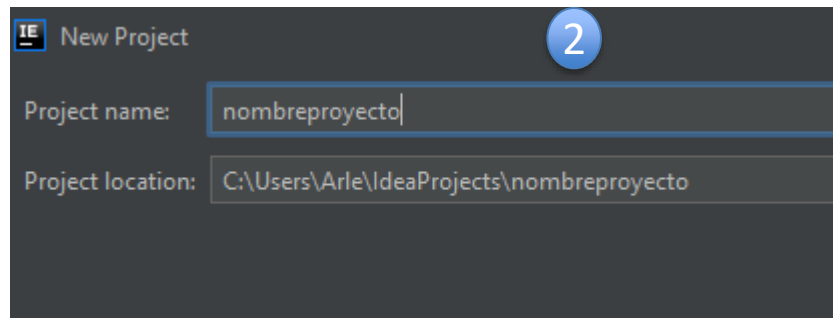
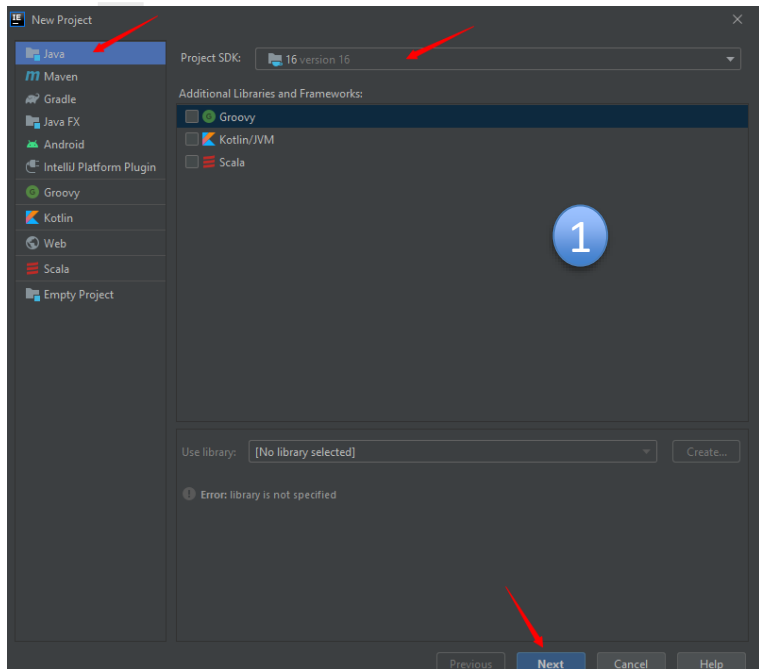
Para desarrollo de JVM y Android

Descargar

.exe ▼

Gratis, código abierto

5 CREAR UN PROYECTO EN INTELIJ IDEA



6 HolaMundo



nombre de la clase

```
1 ► public class HolaMundo {  
2 ►     public static void main(String[] args) {  
3         System.out.println("hola mundo");  
4     }  
5  
6 }
```

3

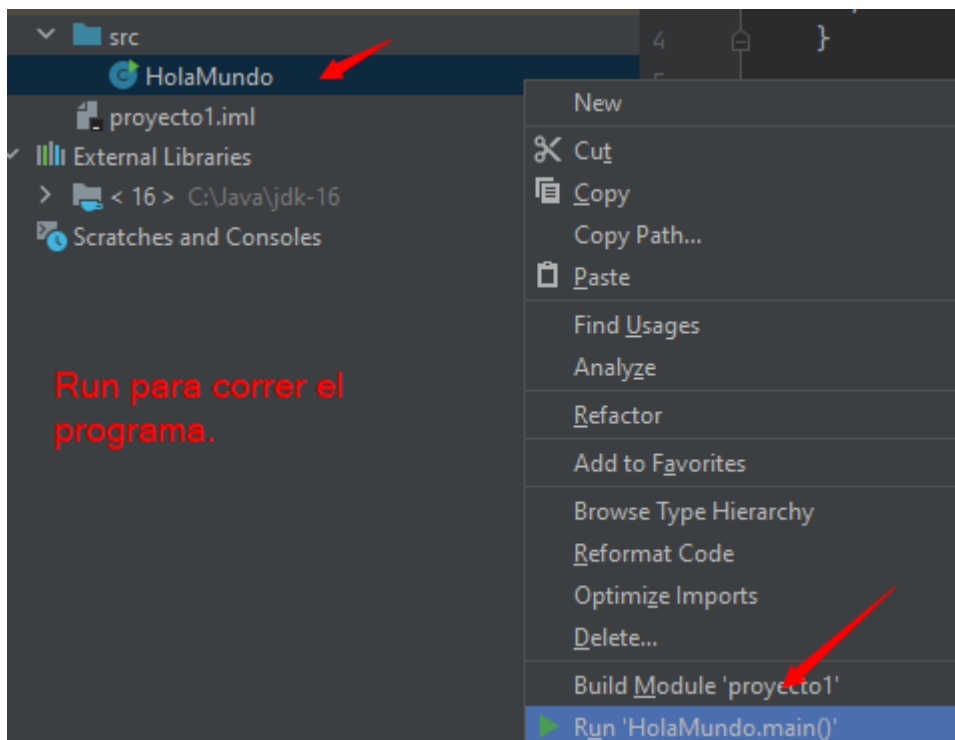
`main(String[] args)`: Nombre del método principal para ejecutar en consola con el JVM. Entre los paréntesis está el parámetro que recibe, en este caso un array (arreglo) que contiene strings; este array dentro del método se le

1

`static`: Indica que no es necesario crear una instancia de la clase para acceder al método. Corresponde a la clase en general, mas no a una instancia.

2

`void`: Indica el tipo de objeto que regresa la función. En este caso la función no regresa ningún valor, por eso es void. Si una función regresara un texto sería `String`, si regresara un número entero sería `int`, etc.



1.me voy a la página principal de NetBeans

<https://netbeans.apache.org/download/index.html>



2. Damos clic dependiendo de la arquitectura de nuestro equipo.

- **Installers:**

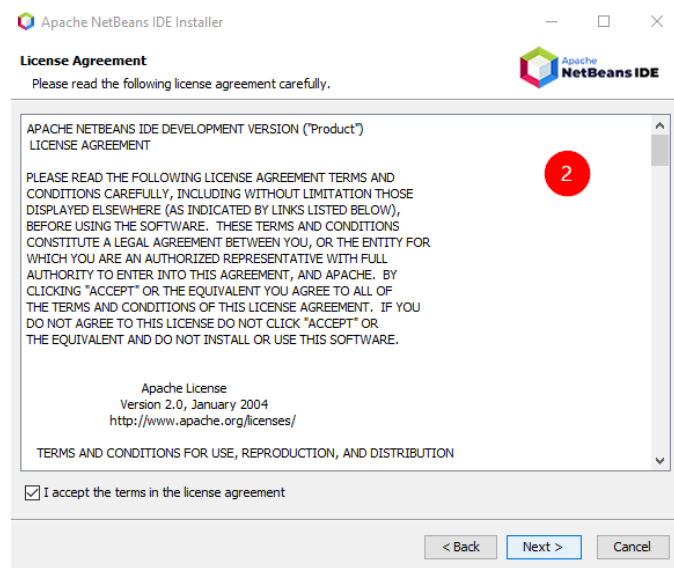
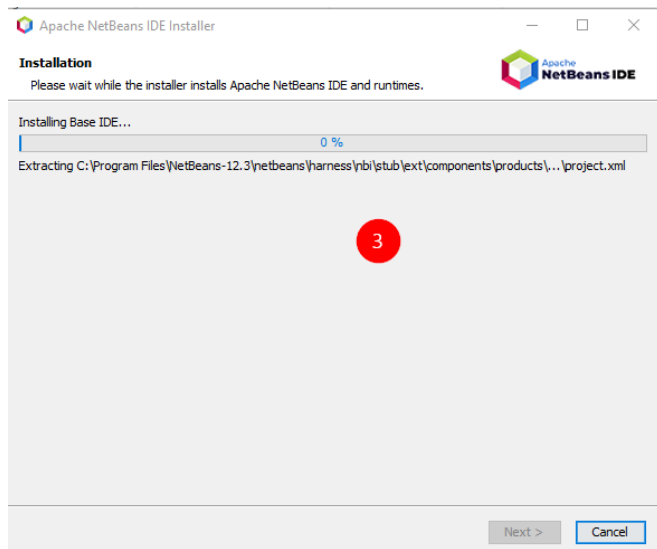
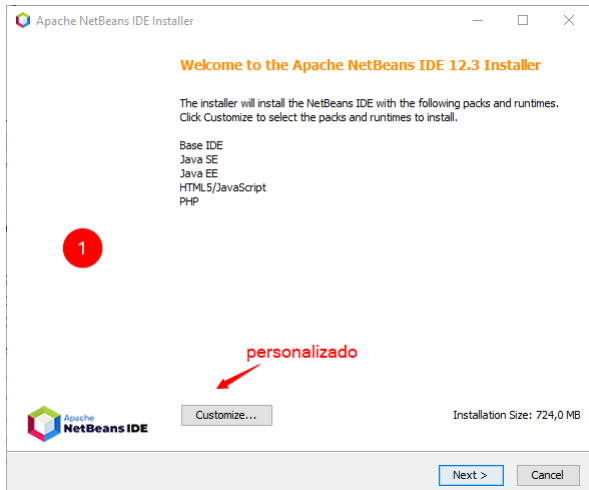
- [Apache-NetBeans-12.3-bin-windows-x64.exe](#) (SHA-512, PGP ASC)
- [Apache-NetBeans-12.3-bin-linux-x64.sh](#) (SHA-512, PGP ASC)
- [Apache-NetBeans-12.3-bin-macosx.dmg](#) (SHA-512, PGP ASC)

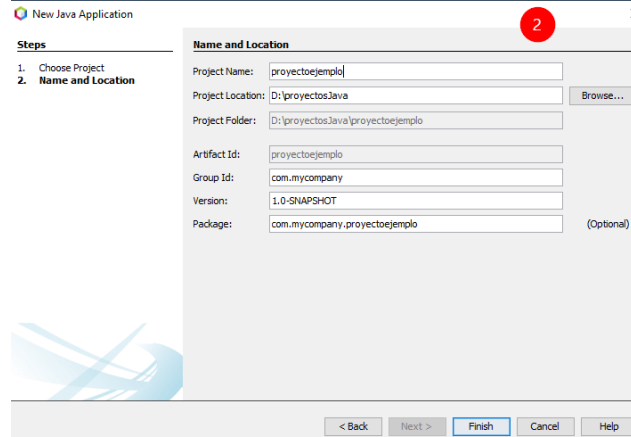
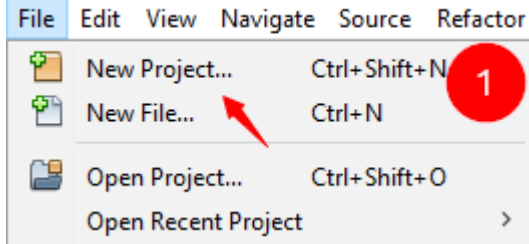
3. Dar clic sobre este enlace.

We suggest the following mirror site for your download:

<https://downloads.apache.org/netbeans/netbeans/12.3/Apache-NetBeans-12.3-bin-windows-x64.exe>







```
6 package com.mycompany.proyectoejemplo;
7
8
9 public class NewMain {
10     public static void main(String[] args) {
11         System.out.println("hola");
12     }
13
14 }
```

3



7 TIPOS PRIMITIVOS EN JAVA



7 TIPO OBJECT



En JAVA el manejo de cadenas es un tipo Object, pero con características particulares y no necesita el uso del operador New.



7 CARACTERES DE ESCAPE AL UTILIZAR CADENAS



En la tabla mostrada, podemos observar los caracteres de escape que son utilizados al momento de desplegar en la salida estándar.

Secuencia de Escape	Descripción
\t	Inserta un tabulador
\b	Inserta un retroceso (backspace)
\n	Inserta una nueva línea
\r	Inserta un retorno de carro
\f	Se mueve a la siguiente pagina (Form feed). Se utiliza para impresoras, no en consolas.
\'	Inserta una comilla simple
\"	Inserta una comilla doble
\\	Inserta una barra invertida

7.1 EJEMPLO APLICATIVO (manejo de cadenas)



```
1 package paquete1;
2
3 ▶ public class Cadenas {
4 ▶     public static void main(String[] args) {
5         String nombre = "Juan Camilo";
6         String apellido = "Perez";
7         System.out.println("concatenación " + nombre + " " + apellido);
8         System.out.println("nueva línea \n " + nombre);
9         System.out.println("Tabulador\t" + nombre );
10        System.out.println("retroceso\b");
11        System.out.println("retorno de carro\r " + nombre + " " + apellido);
12        System.out.println("comilla simple\' " + nombre + " " + apellido);
13        System.out.println("comilla doble\" \" " + nombre + " " + apellido);
14    }
15 }
```

8 EJEMPLO APLICATIVO



```
//Variables boolean, declaracion
boolean bool1;
//inicializacion
bool1 = true;
//Declaracion e inicializacion
boolean bool2 = false;
System.out.println("Valor bool1:" + bool1);
System.out.println("Valor bool2:" + bool2);
System.out.println("");
```

```
//Variables byte
byte b1 = 10;
System.out.println("Valor byte1:" + b1);
System.out.println("");
```

```
//Variables short
short s1 = 2;
System.out.println("Valor char1:" + s1);
System.out.println("");
```

```
//Variables long
long long1 = 10, long2 = 20L;
System.out.println("Valor long1:" + long1);
System.out.println("Valor long2:" + long2);
System.out.println();
```

```
char ch1 = 65, ch2 = 'A';
System.out.println("Valor char1:" + ch1);
System.out.println("Valor char2:" + ch2);
System.out.println("");
```

```
//Variable enteras
int decimal = 100;
int octal = 0144;//Valor octal inicia con 0
int hexa = 0x64;//Valor hexadecimal onicia con 0x
System.out.println("Valor int decimal:" + decimal);
System.out.println("Valor int octal:" + octal);
System.out.println("Valor int hexadecimal:" + hexa);
System.out.println();
```

```
//Variables float
float f1 = 15, f2 = 22.3F;
System.out.println("Valor float1:" + f1);
System.out.println("Valor float2:" + f2);
System.out.println();
```

```
//Variables double
double d1 = 11.0, d2 = 30.15D;
System.out.println("Valor long1:" + d1);
System.out.println("Valor long2:" + d2);
System.out.println();
```

Operadores	Operador
Operadores aritméticos	+ , - , * , / , %
Operadores de relación	< , > , <= , >= , != , ==
Operadores lógicos	&& o & , o , ! , ^
Operadores unitarios	~ , -
Operadores a nivel de bits	& , , ^ , << , >> , >>>
Operadores de asignación	++ , -- , = , *= , /= , %= , += , -= , <<= , >>= , >>>= , &= , = , ^=
Operador condicional	?:
Prioridad y orden de evaluación	() , [] , . , - ~ ! ++ -- , new (tipo) expresión , * / % , + - , << >> >>> , < <= > >= , == != , & , ^ , , && , , ? :

10 PALABRAS RESERVADAS



abstract	do	import	public	throws
boolean	double	instanceof	return	transient
break	else	int	short	try
byte	extends	interface	static	void
case	final	long	strictfp	volatile
catch	finally	native	super	while
char	float	new	switch	
class	for	package	synchronized	
continue	if	private	this	
default	implements	protected	throw	
const	goto	Palabras reservadas (sin uso actualmente)		
null	true	false	Literales Reservadas	

11 EJERCICIO OPERADORES



```
System.out.println("Operadores Aritmeticos:");
int a = 1 + 2;
int b = a * 4;
int c = b / 2;
int d = c - a;
int e = -d;

System.out.println("a = " + a);
System.out.println("b = " + b);
System.out.println("c = " + c);
System.out.println("d = " + d);
System.out.println("e = " + e);

System.out.println("\nOperador Módulo (residuo):");
System.out.println("x mod 10 = " + a % 2);
```

```
System.out.println("\nOperador Compuesto:");
//a = a + 4
a += 2;
b -= 4;
c *= a;

System.out.println("a = " + a);
System.out.println("b = " + b);
System.out.println("c = " + c);

System.out.println("\nOperador Incremento:");
//int a = a + 1;
//Puede reescribirse como
a++;
System.out.println("a = " + a);
```

```
//Preincremento/decremento (se incrementa antes de asignar el valor)
c = ++a;
//Postincremento/decremento (se incrementa despues de asignar el valor)
//La siguiente vez que se evalua b, es cuando se incrementa
d = b++;
System.out.println("b = " + b);
System.out.println("c = " + c);
System.out.println("d = " + d);
```



```
System.out.println("\nOperador relacional:");
//a es menor que b
boolean res = a < b;
System.out.println("res = " + res);

System.out.println("\nOperador Ternario:");
int min = (a < b) ? a : b;
System.out.println("min = " + min);

System.out.println("\nOperador de Asignación:");
int i, j, k;
//cadena de asignación
i = j = k = 100; // valor de x, y, and z igual a 100
System.out.println("i = " + i);
System.out.println("j = " + j);
System.out.println("k = " + k);
```

11 EJERCICIO PRECEDENCIA OPERADORES



```
System.out.println("Primer Ejemplo Precedencia Operadores");
int x = 5;
int y = 10;
int z = ++x * y--;

System.out.println("x = " + x);
System.out.println("y = " + y);
System.out.println("z = " + z);

System.out.println("Ejemplo Evaluacion");
System.out.println(4 + 5 * 6 / 3);
System.out.println((4 + 5) * (6 / 3));

System.out.println("Otro Ejemplo Evaluacion");
System.out.println(1 + 2 - 3 * 4 / 5);
System.out.println(1 + 2 - (3 * (4 / 5)));

System.out.println("\nOtro ejemplo");
//Si detecta una cadena, lo demas lo convierte en cadena
System.out.println("1 + 2 = " + 1 + 2);
//Los parentesis rompen esta regla, ya que tiene la mayor prioridad
System.out.println("(1 + 2) = " + (1 + 2));

System.out.println("\nOtro ejemplo");
//El operador + es asociativo a la izquierda
System.out.println(1 + 2 + "abc");//Detecta una operacion primero
System.out.println("abc" + 1 + 2);//Detecta una cadena primero
```





G R A C I A S

Línea de atención al ciudadano: 018000 910270
Línea de atención al empresario: 018000 910682



@SENAcomunica

www.sena.edu.co