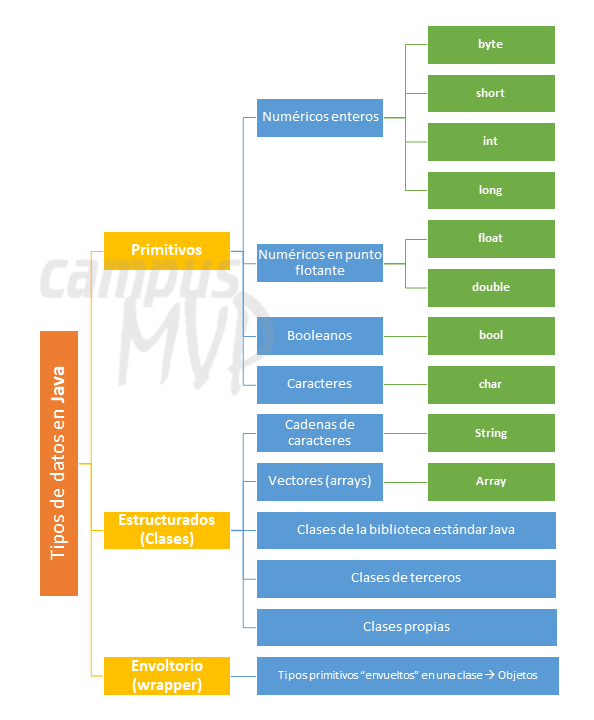
**Evidencia Dia 3 Semana 2 - Clase jueves 06 de Mayo - MODULO I**

**Investigación**

**Interfaz de usuario gráfica, Texto, Aplicación

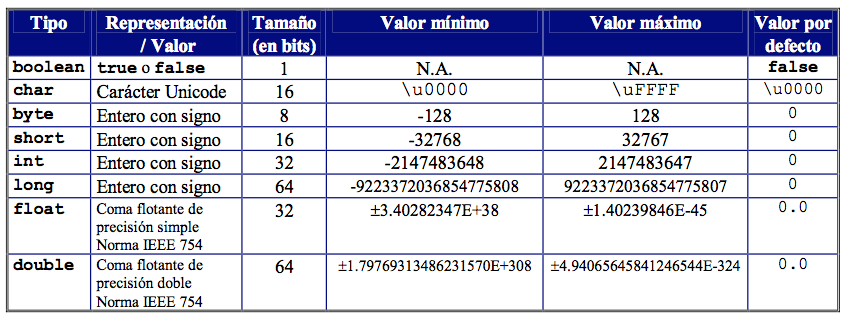
Descripción generada automáticamente**

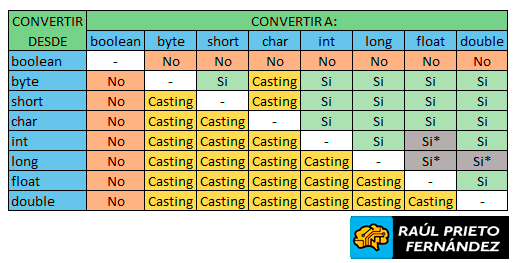
**Respuesta**



Tabla

Descripción generada automáticamente





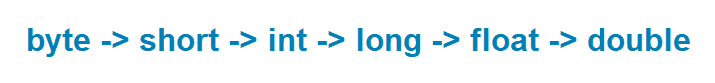
Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

1. Conversión automática de tipos o ampliación

La conversión de ampliación tiene lugar cuando dos tipos de datos se convierten automáticamente. Esto sucede cuando:

* Los dos tipos de datos son compatibles.
* Cuando asignamos el valor de un tipo de datos más pequeño a un tipo de datos más grande.

[](https://www.javadesdecero.es/wp-content/uploads/2017/12/Conversi%C3%B3n-autom%C3%A1tica-de-tipos-Java.png)

Conversión automática de tipos Java

Por ejemplo, en Java, los tipos de datos numéricos son compatibles entre sí, pero no se admite la conversión automática de tipo numérico a char o boolean. Además, **char y boolean no son compatibles entre sí**.

Ejemplo:

class Test

{

public static void main(String[] args)

{

int i = 100;

//conversion automatica de tipo

long l = i;

//conversion automatica de tipo

float f = l;

System.out.println("Valor Int "+i);

System.out.println("Valor Long "+l);

System.out.println("Valor Float "+f);

}

}

Salida:

Valor Int 100

Valor Long 100

Valor Float 100.0

## 2. Conversión explícita de tipo o casting

Si queremos asignar un valor de tipo de dato más grande a un tipo de dato más pequeño, realizamos un **casteo/casting**o lo que se conoce como**conversión de tipo explícito**.

* Esto es útil para tipos de datos incompatibles donde la conversión automática no se puede realizar.
* **char** y **int** no son compatibles entre sí. Veamos luego cuando tratamos de convertir uno en otro.

[](https://www.javadesdecero.es/wp-content/uploads/2017/12/Conversi%C3%B3n-expl%C3%ADcita-o-casting-Java.png)Conversión explícita o casting en Java

Ejemplo:

// Programa Java para ilustrar tipos de datos incompatibles

// para la conversión de tipo explicita

public class Test

{

public static void main(String[] argv)

{

char ch = 'c';

int num = 88;

ch = num;

}

}

Salida:

9: error: incompatible types: possible lossy conversion from int to char

ch = num;

^

1 error

**¿Cómo hacer una conversión explícita?**

Ejemplo:

// Programa Java para ilustrar el tipo de conversion explicita

class Test

{

public static void main(String[] args)

{

double d = 100.04;

//casting de tipo

long l = (long)d;

//casting de tipo

int i = (int)l;

System.out.println("Valor Double "+d);

//parte fraccionaria perdida

System.out.println("Valor Long "+l);

//parte fraccionaria perdida

System.out.println("Valor Int "+i);

}

}

Salida:

Valor Double 100.04

Valor Long 100

Valor Int 100

[×](https://javadesdecero.es/basico/conversion-tipo-ejemplos-casting/)**Nota**: Al asignarle valor al tipo de **byte**, la parte fraccionaria se pierde y se reduce al módulo 256 (rango de byte).

Ejemplo:

// Programa Java para ilustrar Conversion de int y double a byte

class Test

{

public static void main(String args[])

{

byte b;

int i = 257;

double d = 323.142;

System.out.println("Conversion de int a byte.");

//i%256

b = (byte) i;

System.out.println("i = " + i + " b = " + b);

System.out.println("\nConversion de double a byte.");

//d%256

b = (byte) d;

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

}

}

Salida:

Conversion de int a byte.

i = 257 b = 1

Conversion de double a byte.

d = 323.142 b= 67

3. Tipo de promoción en Expresiones

public class PromocionAutomatica

{

public static void main(String args[])

{

byte a = 40;

byte b = 50;

byte c = 100;

int d = a \* b / c;

}

}

Del código anterior, notamos que el resultado de *a\* b* excede el rango de **byte**. Para manejar este tipo de problema, Java automáticamente promueve cada operando *byte* o *short* a ***int***. En este ejemplo, a\* b se realiza usando enteros.

Al evaluar expresiones, el valor intermedio puede exceder el rango de operandos y, por lo tanto, se promoverá el valor de la expresión. Algunas condiciones para la promoción de tipo son:

* Java automáticamente promueve cada operando byte, short, o char al evaluar una expresión.
* Si un operando es long, float o double, la expresión se promueve a long, float o double, respectivamente.

Ejemplo:

// Programa Java para ilustrar la promoción de Tipo en Expresiones

class Test

{

public static void main(String args[])

{

byte b = 42;

char c = 'a';

short s = 1024;

int i = 50000;

float f = 5.67f;

double d = .1234;

// La expresión

double result = (f \* b) + (i / c) - (d \* s);

//Resultado después de todas las 'promociones'

System.out.println("resultado = " + result);

}

}

Salida:

resultado = 626.7784146484375

## 4. Casting de Tipo explícito en Expresiones

Al evaluar expresiones, el resultado se actualiza automáticamente a un tipo de datos más grande del operando. Pero si almacenamos ese resultado en un tipo de datos más pequeño, genera un error de tiempo de compilación, por lo que debemos “castear” el resultado.

Ejemplo:

// Programa Java para ilustrar el tipo de conversion int a byte

class Test

{

public static void main(String args[])

{

byte b = 50;

//casting de tipo int a byte

//mostraría error si no detallamos (byte)

b = (byte)(b \* 2);

System.out.println(b);

}

}

Salida:

100

[×](https://javadesdecero.es/basico/conversion-tipo-ejemplos-casting/)**NOTA**: En el caso de operandos individuales, el resultado se convierte en int y luego se realiza una conversión de tipos en consecuencia.

**Datos Primitivos**

Interfaz de usuario gráfica, Aplicación, Tabla, Excel

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

Imagen que contiene Dibujo de ingeniería

Descripción generada automáticamente

**Datos Numéricos**

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**el atajo para el system.out.println es sout + ctrl + espacio**

**Ejemplo de Casting o Casteo**

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Ejemplo de Parseo**

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

**Ejercicio**

**Hacer una aplicación que pida 3 numeros al usuario y luego los muestre de menor a mayor y de mayor a menor**

**Respuesta**

**import** java.util.Scanner;

**public** **class** Ejercicio050521 {

**public** **static** **void** main(String[] args) {

**int** a,b,c, Mayor, Medio, Menor;

Scanner lector = **new** Scanner(System.***in***);

System.***out***.println("Dados tres numeros, devolver los tres en orden ascendente y descendente");

System.***out***.println("Ingrese el primer numero");

a = lector.nextInt();

System.***out***.println("Ingrese el segundo numero");

b = lector.nextInt();

System.***out***.println("Ingrese el tercer numero");

c = lector.nextInt();

**if** (a > b && a > c) {

Mayor = a;

} **else** **if** (b > a && b > c) {

Mayor = b;

} **else** {

Mayor = c;

}

**if** (a < b && a < c) {

Menor = a;

} **else** **if** (b < a && b < c) {

Menor = b;

} **else** {

Menor = c;

}

Medio = (a + b + c) - (Mayor + Menor);

System.***out***.println("");

System.***out***.println("el orden ascendente de los numeros es:");

System.***out***.println(Menor + "" + Medio + "" + Mayor);

Medio = (a + b + c) - (Mayor + Menor);

System.***out***.println("el orden descendente de los numeros es:");

System.***out***.println(Mayor + "" + Medio + "" + Menor);

}

}