



<p><b><i>Listado Básico 1</i></b></p>
---------------------------------------

Fecha entrega: <04/10/2017>

Autores: <Angelo Barbara>

1. Los tipos de datos “primitivos” tienen un tratamiento especial, ya que no son objetos (no se crean con new) y se usan frecuentemente en los programas. La razón para el tratamiento especial es que crear un objeto con new -especialmente variables pequeñas y simples- no es eficiente porque new coloca el objeto en el montículo (trozo de memoria RAM). Para estos tipos, Java crea una variable “automática” que no es una referencia. La variable guarda el valor, y se coloca en la pila (otro trozo de memoria RAM más rápida y eficiente) para que sea más eficiente. Java determina el tamaño de cada tipo primitivo. Estos tamaños no varían de una plataforma a otra. Indica los bits (no bytes) que ocupan y rango de valores posibles

VARIABLES	TAMAÑO	VALORES	TIPO
byte	8 bits	-128 a +128	Entero corto
short	16 bits	-32768 a +32767	Entero
int	32 bits	$-2^{31}$ a $2^{31}-1$	Entero largo
long	64 bits	$-2^{63}$ a $2^{63}-1$	Entero largo
float	32 bits	-3,402823e38 a 3,402823e38	Decimal simple
double	64 bits	-1,79769313486232e308 a 1,79769313486232e308	Decimal doble
boolean		True, false	0,1
char	16 bits	\u0000 a \uffff	Carácter Unicode

2. Crea la clase EjemploListaAnidada que muestre la siguiente salida. Para ello utiliza el método System.out.print() con los caracteres \n y \t

```

C:\Users\d17barba\Desktop\Tareas\PGN>cd cd tema1
El sistema no puede encontrar la ruta especificada.
C:\Users\d17barba\Desktop\Tareas\PGN>cd Tema1
C:\Users\d17barba\Desktop\Tareas\PGN\Tema1>cd Listado
C:\Users\d17barba\Desktop\Tareas\PGN\Tema1\Listado>javac EjemploListaAnidada.java
C:\Users\d17barba\Desktop\Tareas\PGN\Tema1\Listado>java EjemploListaAnidada
Ejemplo de una lista anidada en HTML:
    <ul>
        <li>
        <li>
        <li>
            <ul>
                <li>
                <li>
                <li>
            </ul>
        </ul>
C:\Users\d17barba\Desktop\Tareas\PGN\Tema1\Listado>

```

3. Crea la clase `TablaAbecedarioMinusculas` que muestre la siguiente salida. Para ello utiliza el método `System.out.print()` con los caracteres `\n` y `\t`. Utiliza una variable `char` con el código Unicode para mostrar la letra ñ. (no uses `\udddd`)

```

C:\Users\d17barba\Desktop\Tareas\PGN\Tema1\Listado>java EjemploListaAnidada
Ejemplo de una lista anidada en HTML:
<ul>
    <li>
    <li>
    <li>
        <ul>
            <li>
            <li>
            <li>
        </ul>
    </li>
</ul>

C:\Users\d17barba\Desktop\Tareas\PGN\Tema1\Listado>java TablaAbecedarioMinusculas
Esto es una tabla con el abecedario en minúsculas:
a      b      c      d
e      f      g      h
i      j      k      l
m      n      ñ      o
p      q      r      s
t      u      v      w
x      y      z

C:\Users\d17barba\Desktop\Tareas\PGN\Tema1\Listado>

```

4. Crea la clase `CaracteresEspagnoles` que muestre la siguiente salida. Para ello utiliza el método `System.out.print()` con los caracteres `\n` y `\t`. Utiliza el formato de carácter `\udddd`

```

C:\Users\d17barba\Desktop\Tareas\PGN\Tema1\Listado>java EjemploListaAnidada
Ejemplo de una lista anidada en HTML:
<ul>
    <li>
    <li>
    <li>
        <ul>
            <li>
            <li>
            <li>
        </ul>
    </li>
</ul>

C:\Users\d17barba\Desktop\Tareas\PGN\Tema1\Listado>java TablaAbecedarioMinusculas
Esto es una tabla con el abecedario en minúsculas:
a      b      c      d
e      f      g      h
i      j      k      l
m      n      ñ      o
p      q      r      s
t      u      v      w
x      y      z

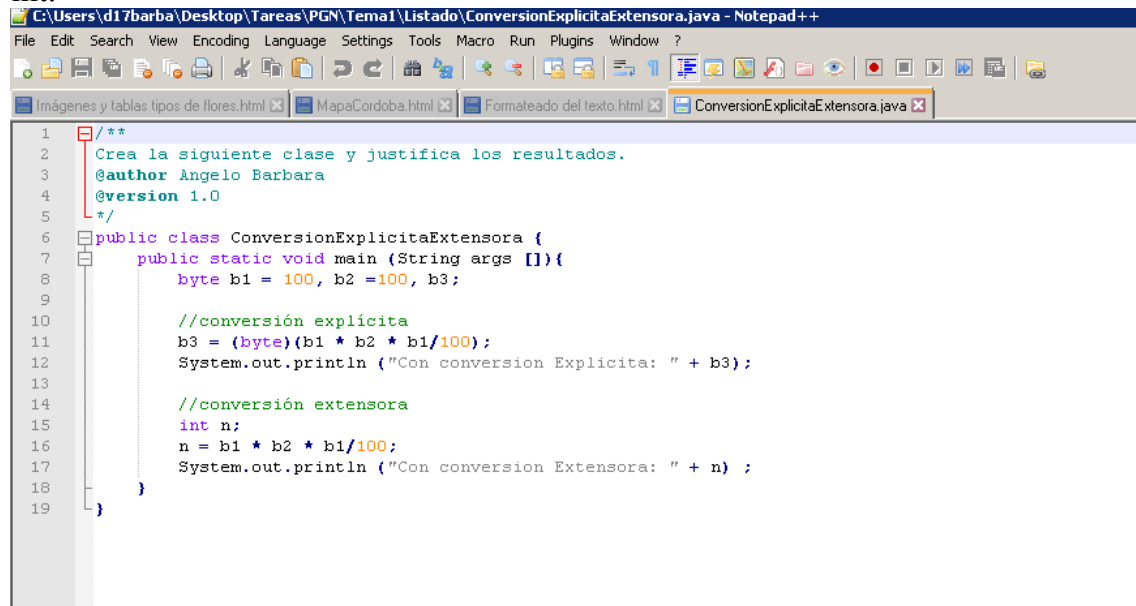
C:\Users\d17barba\Desktop\Tareas\PGN\Tema1\Listado>java CaracteresEspagnoles
Estos caracteres son muy nuestros:
á      í      ó      ú      ñ      Ñ      à      ó

C:\Users\d17barba\Desktop\Tareas\PGN\Tema1\Listado>

```

## 5. Crea la siguiente clase y justifica los resultados

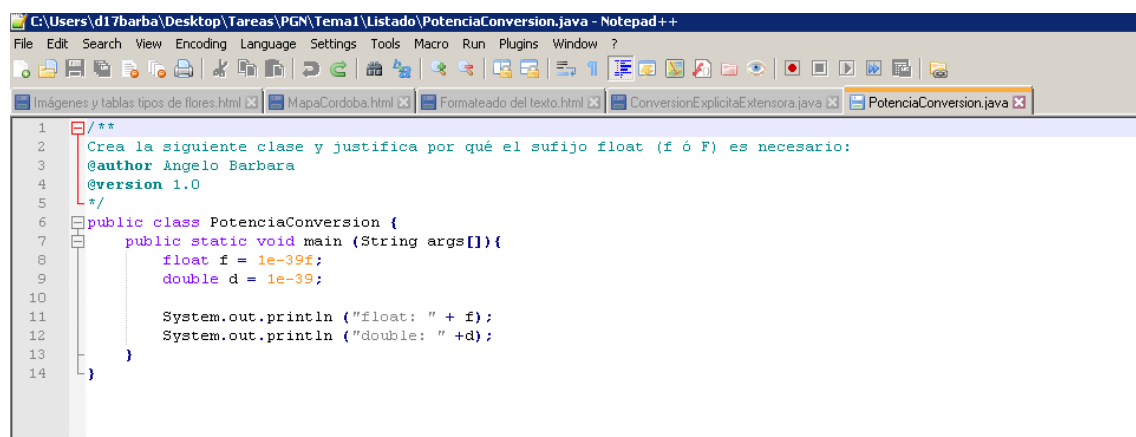
El resultado de la operación no está incluido entre los valores expresables con la variable byte, por tanto forzando java con el casting nos devuelve un resultado equivocado. Sin embargo podemos expresarlo sin problema utilizando la variable int.



```
1  /**
2   * Crea la siguiente clase y justifica los resultados.
3   * @author Angelo Barbara
4   * @version 1.0
5   */
6  public class ConversionExplicitaExtensora {
7      public static void main (String args []){
8          byte b1 = 100, b2 =100, b3;
9
10         //conversión explícita
11         b3 = (byte)(b1 * b2 * b1/100);
12         System.out.println ("Con conversion Explícita: " + b3);
13
14         //conversión extensora
15         int n;
16         n = b1 * b2 * b1/100;
17         System.out.println ("Con conversion Extensora: " + n) ;
18     }
19 }
```

## 6. Crea la siguiente clase y justifica por qué el sufijo float (f ó F) es necesario:

El casting float es necesario para que no lo interprete por defecto con un double.



```
1  /**
2   * Crea la siguiente clase y justifica por qué el sufijo float (f ó F) es necesario:
3   * @author Angelo Barbara
4   * @version 1.0
5   */
6  public class PotenciaConversion {
7      public static void main (String args[]){
8          float f = 1e-39f;
9          double d = 1e-39;
10
11         System.out.println ("float: " + f);
12         System.out.println ("double: " +d);
13     }
14 }
```

## 7. Crea la clase EsconditeConfuso con las siguientes sentencias. Interpretalas y coméntalas en el código

Nos da error porque en la misma clase hemos declarado dos veces la misma variable con valores diferentes.

## 8. Indica si es verdadero o falso

- a) El tamaño del tipo boolean no está explícitamente definido; sólo se especifica que debe ser capaz de tomar los valores true o false.

Verdadero.

- b) Todos los tipos numéricos tienen signo, de forma que es inútil tratar de utilizar tipos sin signo.

Verdadero

- c) El ámbito determina tanto la visibilidad como la vida de los nombres definidos dentro. Se delimita mediante corchetes []

Falso

- d) La asignación de una variable dentro de un ámbito se mantiene fuera del ámbito.

Falso

- e) Una variable definida dentro de un ámbito está disponible también fuera de su ámbito.

Falso

- f) Java permite convertir (casting) cualquier tipo primitivo en cualquier otro tipo, excepto boolean, que no permite ninguna conversión

Verdadero

- g) Java convierte automáticamente un tipo de datos en otro cuando sea adecuado (de int a float, p.ej.)

Verdadero

- h) Con las conversiones reductoras (de float a int, p.ej.) nunca se corre el riesgo de perder información

Falso.

- i) Hay representación literal de números binarios en Java.

Falso

- j) Los sufijos de double, float y long pueden aparecer tanto en mayúsculas como en minúsculas: 0d, 0D, 0f, 0F, 0l, 0L.

Verdadero

- k) El prefijo hexadecimal 0x siempre ha de ir en minúsculas

Falso

**9. Si se asigna un valor a una variable fuera de rango (mayor de lo establecido), ¿qué ocurre? ¿Existe alguna manera de resolverlo? Demuéstralo mediante un ejemplo. (VariableFueraDeRango)**

Asignando un valor fuera de rango dará error. Sin embargo podemos resolverlo forzando java con un casting o simplemente cambiando la variable con una más oportuna.

**10. Crea una clase Asignacion Variables con el siguiente código. Corrígela si es necesario y coméntala en el mismo código fuente, indicando errores y valores de variables.**

Si no se declara la variable i nos dará error.

```
1  /**
2   * Crea una clase AsignacionVariables con el siguiente código. Corrígela si es necesario y coméntala en el mismo código fuente, indicando errores y valores de variab
3   * @author Angelo Barbara
4   * @version 1.0
5   */
6  public static class AsignacionVariable{
7      public static void main (String args[]){
8          int i = 2400; //Es necesario declarar la variable i o dará error
9          byte b = (byte)i;
10         System.out.println(i);
11         System.out.println(b);
12     }
13 }
```

Declarando la variable i es posible ejecutar sin problema, ya que el casting byte nos permite expresar un valor fuera de rango.

**11. Como veremos más adelante, para garantizar que un objeto ha sido inicializado existen los constructores. ¿Existe algún problema si se intenta acceder al valor de una variable y ésta no ha sido inicializada?**

Se puede declarar una variable sin un valor definido, sin embargo no se puede predecir el valor que se le asignará.

**12. Qué tipo de problema puede existir con la asignación de variables**

Declaración de un valor fuera de rango.

Declaración de la misma variable mas veces

Declaración de los mismos valores para dos variables distintas en la misma clase