

## Programación

### UT2 – TAREA6

(10 puntos) Indica cuáles de los siguientes literales son válidos en Java. Si el literal es válido indica además de qué tipo es (int, double, long, etc) y el sistema de numeración en el que está escrito (decimal, binario, octal, hexadecimal). Si el literal no es válido explica porqué no lo es.

1) 0.5	2) .5	3) 9.3e12	4) 9.3e-12
5) 12345678	6) 12345678_L	7) 0.8E+0.8	8) 0.8E 8
9) 05_15	10) 018CDF	11) 0XBC5DA	12) 0x87e3a
13) 234567L	14) 0_B11	15) 010101	16) 0_557
17) .00.8E2	18) .3e3f	19) 0b111	20) 12_234L
21) 0Xabcd	22) 0xabcEL	23) _234	24) 1010B
25) 0x1010B	26) 1_234.2E-2	27) 1234.2EF	28) 1234.2E3F
29) 1_1.2e_2	30) 0bABCDL	31) 0X1A	32) 0X12AL
33) abcd	34) 0125	35) .01011	36) 3e12
37) 3_e12	38) -3E-1_2	39) 0.8E	40) 0B1212
41) 1_2_3	42) 0xedad	43) 0XBE2	44) 101e2
45) B1101	46) 1.34.5	47) 12.3E4F	48) 0X12AG

Cada apartado vale 0,2083 puntos.

Antes de ver la solución te comento que en Java se pueden usar subrayados (guiones bajos) en literales numéricos para mejorar la legibilidad del código. Esta característica fue introducida en Java 7 y es especialmente útil para separar dígitos en números largos. Los subrayados pueden ser colocados entre dígitos para hacer el número más fácil de leer. Sin embargo, hay algunas reglas que debes seguir:

1. No puedes colocar un subrayado al inicio o al final de un número.
2. No puedes colocar un subrayado junto a un punto decimal en números de punto flotante.
3. No puedes colocar un subrayado antes de un sufijo de tipo (**L** para **long**, **F** para **float**, etc.).

### SOLUCIÓN:

LITERAL	VÁLIDO	TIPO	SISTEMA NUMERACIÓN
1) 0.5	SI	<code>double</code>	<code>decimal</code>
2) .5	SI. Si la parte entera es 0 se puede omitir	<code>double</code>	<code>decimal</code>
3) 9.3e12	SI. Representa el valor $9.3 \times 10^{12}$	<code>double</code>	<code>decimal</code>
4) 9.3e-12	SI. Representa el valor $9.3 \times 10^{-12}$	<code>double</code>	<code>decimal</code>
5) 12345678	SI	<code>int</code>	<code>decimal</code>
6) 12345678_L	NO. Posición no válida para el carácter _		
7) 0.8E+0.8	NO. Valor no válido para el exponente en un literal Java. No puede contener decimales. Intenta representar el valor $0.8 \times 100.8$		
8) 0.8E 8	NO. Hay un espacio en blanco entre E y 8		
9) 05_15	SI	<code>int</code>	<code>octal</code>

10) 018CDF	NO. Si el número comienza por 0 está indicando que es un entero escrito en octal pero contiene los caracteres CDF no válidos para el sistema octal. Si fuese un entero hexadecimal debería comenzar por 0X.		
11) 0XBC5DA	SI	int	hexadecimal
12) 0x87e3a	SI	int	hexadecimal
13) 234567L	SI	long	decimal
14) 0_B11	NO. El carácter _ no puede separar los caracteres 0B que indican que el número está escrito en binario.		
15) 010101	SI	int	octal
16) 0_557	SI	int	octal
17) .00.8E2	NO. El literal solo puede contener un punto (.) que separe la parte entera de la parte decimal.		
18) .3e3f	SI	float	decimal
19) 0b111	SI	int	binario
20) 12_234L	SI	long	decimal
21) 0Xabcd	SI	int	hexadecimal
22) 0xabcEL	SI	long	hexadecimal
23) _234	NO. El carácter _ no puede aparecer al principio del número.		
24) 1010B	NO. Si fuese binario debería empezar por 0B: 0B1010		
25) 0x1010B	SI	int	hexadecimal

26) 1_234.2E-2	SI	double	decimal
Representa el valor $1234.2 \times 10^{-2}$			
27) 1234.2EF	NO. Falta el valor del exponente entre los caracteres E y F		
28) 1234.2E3F	SI	float	decimal
Representa el valor $1234.2 \times 10^3$			
29) 1_1.2e_2	NO. En un literal Java de tipo float o double el carácter _ no puede aparecer antes o después del carácter e. En este caso se quiere representar el valor $11.2 \times 10^2$		
30) 0bABCDL	NO. Comienza por 0b. Si fuese hexadecimal debería comenzar por 0X.		
31) 0X1A	SI	int	hexadecimal
32) 0X12AL	SI	long	hexadecimal
33) abcd	NO. Si fuese hexadecimal debería comenzar por 0X.		
34) 0125	SI	int	octal
35) .01011	SI	double	decimal
36) 3e12	SI	double	decimal
Representa el valor $3 \times 10^{12}$			
37) 3_e12	NO. En un literal Java de tipo float o double el carácter _ no puede aparecer antes o después del carácter e. En este caso se quiere representar el valor $3 \times 10^{12}$		
38) -3E-1_2	SI	double	decimal
Representa el valor $-3 \times 10^{-12}$			
39) 0.8E	NO. Falta el valor del exponente		

40) 0B1212	NO. Un número <b>binario</b> (0B) solo puede contener ceros y unos		
41) 1_2_3	SI	<b>int</b>	<b>decimal</b>
42) 0xedad	SI	<b>int</b>	hexadecimal
43) 0XBE2	SI	<b>int</b>	hexadecimal
44) 101e2	SI	<b>double</b>	<b>decimal</b>
	Representa el valor $101 \cdot 10^2$		
45) B1101	NO. Si fuese binario debería empezar por 0B		
46) 1.34.5	NO. El literal solo puede contener un <b>punto</b> (.) que separe la parte entera de la parte <b>decimal</b> .		
47) 12.3E4F	SI	<b>float</b>	<b>decimal</b>
	Representa el valor $12.3 \cdot 10^4$		
48) 0X12AG	NO. G no es un carácter válido para un número <b>hexadecimal</b> (0X)		