

## 2.B. Tipos de datos.

### 2. Literales de los tipos primitivos.

Un **literal, valor literal o constante literal** es un valor concreto para los tipos de datos primitivos del lenguaje, el tipo String o el tipo null.

Los **literales booleanos** tienen dos únicos valores que puede aceptar el tipo: true y false. Por ejemplo, con la instrucción boolean encontrado = true; estamos declarando una variable de tipo booleana a la cual le asignamos el valor literal true.

Los **literales enteros** se pueden representar en tres notaciones:

- **Decimal**: por ejemplo 20. Es la forma más común.
- **Octal**: por ejemplo 024. Un número en octal siempre empieza por cero, seguido de dígitos octales (del 0 al 7).
- **Hexadecimal**: por ejemplo 0x14. Un número en hexadecimal siempre empieza por 0x seguido de dígitos hexadecimales (del 0 al 9, de la 'a' a la 'f' o de la 'A' a la 'F').

Las constantes literales de tipo long se le debe añadir detrás una l ó L, por ejemplo 873L, si no se considera por defecto de tipo int. Se suele utilizar L para evitar la confusión de la ele minúscula con 1.

Los **literales reales** o en coma flotante se expresan con coma decimal o en notación científica, o sea, seguidos de un exponente e ó E. El valor puede finalizarse con una f o una F para indica el formato float o con una d o una D para indicar el formato double (por defecto es double). Por ejemplo, podemos representar un mismo literal real de las siguientes formas: 13.2, 13.2D, 1.32e1, 0.132E2. Otras constantes literales reales son por ejemplo: .54, 31.21E-5, 2.f, 6.022137e+23f, 3.141e-9d.

Un **literal carácter** puede escribirse como un carácter entre comillas simples como 'a', 'ñ', 'Z', 'p', etc. o por su código de la tabla Unicode, anteponiendo la secuencia de escape '\ ' si el valor lo ponemos en octal o '\u' si ponemos el valor en hexadecimal. Por ejemplo, si sabemos que tanto en ASCII como en Unicode, la letra A (mayúscula) es el símbolo número 65, y que 65 en octal es 101 y 41 en hexadecimal, podemos representar esta letra como '\101' en octal y '\u0041' en hexadecimal. Existen unos caracteres especiales que se representan utilizando secuencias de escape:

Secuencias de escape en Java			
Secuencia de escape	Significado	Secuencia de escape	Significado
\b	Retroceso	\r	Retorno de carro
\t	Tabulador	\"	Carácter comillas dobles
\n	Salto de línea	\'	Carácter comillas simples
\f	Salto de página	\\	Barra diagonal

**Normalmente, los objetos en Java deben ser creados con la orden new. Sin embargo, los literales String no lo necesitan ya que son objetos que se crean implícitamente por Java.**

Los **literales de cadenas de caracteres** se indican entre comillas dobles. En el ejemplo anterior "El primer programa" es un literal de tipo cadena de caracteres. Al construir una cadena de caracteres se puede incluir cualquier carácter Unicode excepto un carácter de retorno de carro, por ejemplo en la siguiente instrucción utilizamos la secuencia de escape '\ ' para escribir dobles comillas dentro del mensaje:

```
String texto = "Juan dijo: \"Hoy hace un día fantástico...\"";
```

En el ejemplo anterior de tipos enumerados ya estábamos utilizando secuencias de escape, para introducir un salto de línea en una cadena de caracteres, utilizando el carácter especial \n.

