Lectura de teclado en java

**Contenido**

 [[ocultar](http://chuwiki.chuidiang.org/index.php?title=Lectura_de_teclado_en_java)]

* [1 *System.in*](http://chuwiki.chuidiang.org/index.php?title=Lectura_de_teclado_en_java#System.in)
* [2 *InputStreamReader* y *BufferedReader*](http://chuwiki.chuidiang.org/index.php?title=Lectura_de_teclado_en_java#InputStreamReader_y_BufferedReader)
* [3 *Scanner*](http://chuwiki.chuidiang.org/index.php?title=Lectura_de_teclado_en_java#Scanner)
* [4 Ejemplo de lectura de un fichero con *Scanner*](http://chuwiki.chuidiang.org/index.php?title=Lectura_de_teclado_en_java#Ejemplo_de_lectura_de_un_fichero_con_Scanner)

*System.in*

En *java* tenemos accesible el teclado desde *System.in*, que es un *InputStream* del que podemos leer *bytes*. Por ejemplo, podemos leer *bytes* del teclado de esta forma

// Lectura de un byte

int aByte = System.in.read();

// Lectura de hasta 10 bytes

byte [] buffer = new byte[10];

System.in.read(buffer);

El problema de leer *bytes*, es que luego debemos convertirlos a lo que necesitemos. Por ejemplo, si tecleamos una letra A mayúscula, el *byte* leído es el 65, correspodiente a la A mayúscula en código *ASCII*. Si tecleamos un 3 y un 2, es decir, un 32, leeremos dos bytes 51 y 52, correspondientes a los caracteres *ASCII* del 3 y del 2, NO leeremos un 32.

*InputStreamReader* y *BufferedReader*

Antes de *java 1.5*, la forma de leer más fácilmente cosas que no sean *bytes* (leer cadenas de texto o números), era usar las clases *InputStreamReader* y *BufferedReader* conjuntamente. La primera es capaz de convertir esos *bytes* a caracteres. La segunda es capaz de leer hasta un fin de línea. La forma de instanciar estas clases para usarlas con *System.in* es la siguiente:

InputStreamReader isr = new InputStreamReader(System.in);

BufferedReader br = new BufferedReader (isr);

Con esto, podemos leer líneas completas usando el *BufferedReader br*. Así, por ejemplo, si escribimos una A, con

String cadena = br.readLine();

obtendremos en cadena una "A".

Sin embargo, seguimos necesitando hacer la conversión si queremos leer números. Por ejemplo, si escribimos un entero 32, en cadena obtendremos "32". La conversión se haría con

int numero = Integer.parseInt (br.readLine());

y debemos, por supuesto, capturar la excpeción en caso de que la cadena no se pueda convertir a *String*. En cualquier caso, todo esto sigue siendo algo engorroso.

*Scanner*

Desde *java 1.5* tenemos disponible para facilitarnos toda esta tarea, la clase *Scanner*. Basta instanciar esta clase pasándole el *System.in* y a partir de ahí tememos muchos métodos para obtener exactamente lo que queremos. Por ejemplo, podemos leer la A así

Scanner sc = new Scanner(System.in);

String cadena = sc.nextLine();

o bien, para obtener el 32, con

int entero = sc.nextInt();

Esta clase es bastante potente, por las siguientes características:

* Tiene varios constructores que admiten, además de System.in, cosas como secuencias de bytes o ficheros. Esto nos permite leer, por ejemplo, ficheros de forma más cómoda.
* Los métodos nextInt() admiten un entero radix, con lo que podríamos leer números en hexadecimal. Por ejemplo, si tecleamos FF y hacemos la lectura con radix 16, obtendríamos un 255.

// Lectura de un número en hexadecimal.

int entero = sc.nextInt(16);

* Admite [Expresiones Regulares en Java](http://chuwiki.chuidiang.org/index.php?title=Expresiones_Regulares_en_Java) como patrones de búsqueda, por lo que podemos leer trozos de línea directamente usando los separadores que queramos o buscando expresiones concretas. Por ejemplo, si introducimos 11:33:44, usando el siguiente código obtendremos los número 11, 33 y 44

Scanner sc = new Scanner(System.in);

// Usamos como delimitador el dos puntos, o bien cualquier

// espacio/fin de línea (el \\s)

sc.useDelimiter("[:\\s]");

// Leemos los tres enteros

int a = sc.nextInt();

int b = sc.nextInt();

int c = sc.nextInt();

// Obtendremos 11-33-44 de salida.

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

Ejemplo de lectura de un fichero con *Scanner*

Supongamos que tenemos un fichero en el que en cada línea hay los datos de una persona. Pueden ser un id, un nombre y una edad, separados por comas y quizás espacios. Algo como lo siguiente

1 , Pedro , 33

2, Juan, 44

4, Antonio, 55

Vamos a hacer y explicar un pequeño programa en java usando *Scanner* que nos permita leer estos datos.

En primer lugar, creamos un [File](http://chuwiki.chuidiang.org/index.php?title=La_clase_File) con el contenido del fichero y después una instancia de *Scanner* pasándole ese File. Por supuesto, al terminar de leer nuestro fichero, debemos cerrarlo

File f = new File("fichero.txt");

Scanner s;

try {

s = new Scanner(f);

//

// Aquí la lectura del fichero

//

s.close();

} catch (FileNotFoundException e) {

e.printStackTrace();

}

Para la lectura del fichero, bastará de momento con un bucle para ir leyendo línea a línea. Para ello, podemos usar el método hasNextLine() que nos indica si hay o no una línea más que leer, y el método nextLine() que nos la devuelve

while (s.hasNextLine()) {

String linea = s.nextLine();

//

// Aquí el tratamiento de la línea

//

}

Para tratar la línea y sacar los tres campos que hay en ella, podemos usar nuevamente otra instancia de la clase *Scanner*. El delimitador para los campos será una coma, precedida o no de uno o más espacios y seguida o no de uno o más espacios. Eso, usando [Expresiones Regulares en Java](http://chuwiki.chuidiang.org/index.php?title=Expresiones_Regulares_en_Java) se expresa así "\\s\*,\\s\*", donde \\s indica un espacio blanco y con asterisco detrás \\s\* indica cero o más espacios en blanco. Por tanto, el código para recoger los tres campos, puede ser como este

Scanner sl = new Scanner(linea);

sl.useDelimiter("\\s\*,\\s\*");

System.out.println(sl.next());

System.out.println(sl.next());

System.out.println(sl.next());

Y si ahora ponemos el programa completo

package com.chuidiang.ejemplos.file\_scanner;

import java.io.File;

import java.io.FileNotFoundException;

import java.util.Scanner;

public class FileScanner {

public static void main(String[] args) {

File f = new File("fichero.txt");

Scanner s;

try {

s = new Scanner(f);

while (s.hasNextLine()) {

String linea = s.nextLine();

Scanner sl = new Scanner(linea);

sl.useDelimiter("\\s\*,\\s\*");

System.out.println(sl.next());

System.out.println(sl.next());

System.out.println(sl.next());

}

s.close();

} catch (FileNotFoundException e) {

e.printStackTrace();

}

}

}