***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();

}

}

}

Sin usar la clase scanner

mport java.io.\*;

public class ENTRADA\_DATOS\_02

{

public static void main(String[] ARGUMENTOS) throws IOException

{

BufferedReader TECLADO = new BufferedReader(new InputStreamReader(System.in));

String VAR\_STRING;

int VAR\_INT;

float VAR\_FLOAT;

double VAR\_DOUBLE;

char VAR\_CHAR;

//-- ENTRADA DE DATOS EN DIFERENTES TIPOS

//-- IMPORTANTE..¡¡¡

//-- SE TIENE QUE INGRESAR SEGUN AL FORMATO DE LA VARIABLE

//-- O NOS GENERARA UN ERROR

System.out.print("INGRESE UN STRING : ");

VAR\_STRING = TECLADO.readLine();

System.out.print("INGRESE UN NUMERO INT : ");

VAR\_INT = Integer.parseInt(TECLADO.readLine());

System.out.print("INGRESE UN DECIMAL FLOAT : ");

VAR\_FLOAT = Float.parseFloat(TECLADO.readLine());

System.out.print("INGRESE UN DECIMAL BYTE : ");

VAR\_DOUBLE = Double.parseDouble(TECLADO.readLine());

System.out.print("INGRESA UN CHAR (UNA SOLA LETRA): ");

VAR\_CHAR = TECLADO.readLine().charAt(0);

//-- SALIDA DE DATOS EN DIFERENTES TIPOS

System.out.println("EL STRING INGRESADO ES : " + VAR\_STRING);

System.out.println("EL INT INGRESADO ES : " + VAR\_INT);

System.out.println("EL FLOAT INGRESADO ES : " + VAR\_FLOAT);

System.out.println("EL DOUBLE INGRESADO ES : " + VAR\_DOUBLE);

System.out.println("EL CHAR INGRESADO ES : " + VAR\_CHAR);

}

}