

# Entrada/Salida en JAVA

Versiones anteriores a  
J2SE 5.0

## E/S Ficheros secuenciales de texto

### ◆ Paquete:

- java.io

### ◆ Excepciones:

- IOException
- La operación que haga E/S con ficheros debe indicarlo en la cabecera

```
public void nombre () throws IOException
```

## Escritura en la salida estándar

### ◆ Escribir:

```
String nombre;  
...  
System.out.println("Hola " + nombre);  
System.out.print("Hola " + nombre);
```

## Salida ficheros secuenciales de texto

### ◆ PrintWriter: clase de Stream

### ◆ Preparar para escritura:

```
PrintWriter salida;  
salida = new PrintWriter (  
    new FileWriter(nombre_fichero));
```

### ◆ Escribir:

```
salida.println("Hola " + nombre);  
salida.print("Hola " + nombre);
```

### ◆ Terminar:

```
salida.close();
```

## Lectura desde la entrada estándar

### ◆ InputStreamReader: clase de Stream

### ◆ BufferedReader: clase para leer de streams

### ◆ Preparar para lectura:

```
BufferedReader entrada;  
entrada = new BufferedReader (  
    new InputStreamReader(System.in));
```

### ◆ Leer:

```
String linea = entrada.readLine();  
String palabra = entrada.read();
```

### ◆ La lectura de partes de un string se realiza con los métodos de la clase StringTokenizer previamente descritos

## Entrada ficheros secuenciales de texto

### ◆ FileReader: clase de Stream

### ◆ BufferedReader: clase para leer de streams

### ◆ Preparar para lectura:

```
BufferedReader entrada;  
entrada = new BufferedReader (  
    new FileReader(nombre_fichero));
```

### ◆ Leer:

```
String linea = entrada.readLine();  
String palabra = entrada.read();
```

### ◆ Final:

```
linea == null
```

### ◆ Terminar:

```
entrada.close();
```

## Entrada ficheros secuenciales de texto

- ◆ StringTokenizer;
  - clase para leer partes de un String
  - se le indican los separadores
- ◆ Leer de un Tokenizer

```
StringTokenizer tokenizer;
tokenizer = new StringTokenizer(linea,"-",false);
palabra = (String)tokenizer.nextElement();
numString = (String)tokenizer.nextElement();
int num = Integer.parseInt(numString.trim());
```
- ◆ Final:

```
tokenizer.hasMoreElements()
```

## J2SE 5.0

J2SE 5.0 incluye, entre otras novedades, paquetes con nuevas clases para realizar la Entrada/Salida a partir de ficheros

## E/S Ficheros secuenciales de texto

- ◆ Clase:
  - java.util.Scanner;
- ◆ Excepciones:
  - InputMismatchException
  - IllegalStateException
  - NoSuchElementException
- ◆ Contiene métodos para leer y escribir datos primitivos

## Entrada ficheros secuenciales de texto

- ◆ Preparar para lectura:
  - ✓ Desde la entrada estándar:

```
Scanner entrada = new Scanner(System.in);
```
  - ✓ Desde fichero:

```
Scanner entrada = new Scanner
(new File("nombre_fichero"));
```
- ◆ Leer un elemento:

```
String palabra = entrada.next();
int entero = entrada.nextInt();
... Otras para el resto de tipos primitivos.
```
- ◆ Leer una línea:

```
String linea = entrada.nextLine();
```
- ◆ Final:  
Cuando `entrada.hasNext()` devuelve `false`

## Ejemplo

Sumar los números enteros contenidos en el fichero «FicheroEnteros.txt»

```
import java.util.Scanner;
...
Scanner entrada=new Scanner("FicheroEnteros.txt");
int suma = 0;
while (entrada.hasNextInt()){
    suma = suma + entrada.nextInt();
}
```

## Entrada ficheros secuenciales de texto

- ◆ Por defecto, `next()` lee hasta encontrar un espacio en blanco
- ◆ Una vez leído un String, es posible dividirlo en varios tokens indicando un separador:

```
String cadena = entrada.next();
String datos[] = cadena.split("-");
```
- ◆ El método `split` guarda la cadena leída en una tabla de Strings.

```
String unToken = datos[posicion];
```
- ◆ Número total de tokens: `datos.length`

## Ejemplo

Leer y escribir la matrícula de un coche con el formato: ABC-1234, utilizando la entrada y la salida estándar

```
import java.util.Scanner;
...
Scanner entrada = new Scanner(System.in);
System.out.println("Teclea una matricula");
String cadena = entrada.next();
String datos[] = cadena.split("-");
for (int i = 0; i < datos.length; i++) {
    System.out.println(datos[i]);
}
```