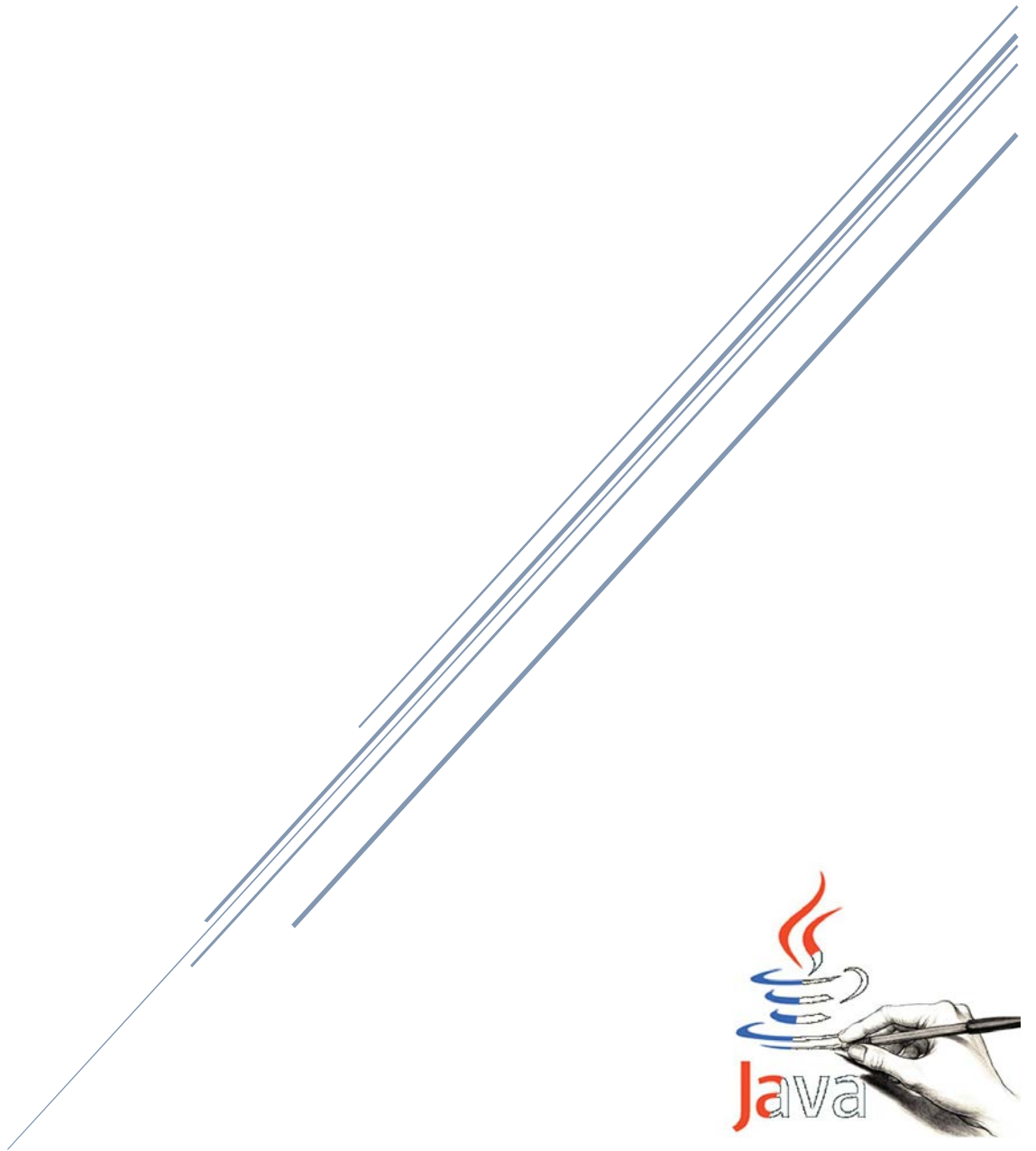


SCANNER

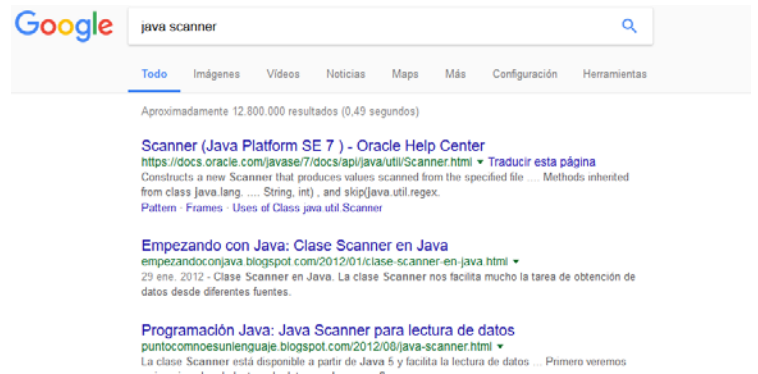
PROGRAMACIÓN



API / Librerías

Java nos proporciona una serie de librerías que nos facilitan el uso del lenguaje en diferentes sectores.

Si hacemos una búsqueda en google de cualquier de las librerías de java, esta nos mostrara donde podemos encontrar estas referencias.



[Scanner \(Java Platform SE 7\) - Oracle Help Center](https://docs.oracle.com/javase/7/docs/api/java/util/Scanner.html)

Si entramos en <https://docs.oracle.com/javase/7/docs/api/java/util/Scanner.html>

Overview Package **Class** Use Tree Deprecated Index Help

Prev Class Next Class Frames No Frames All Classes

Summary: Nested | Field | Constr | Method Detail: Field | Constr | Method

java.util

Class Scanner

java.lang.Object
java.util.Scanner

All Implemented Interfaces:
Closeable, AutoCloseable, Iterator<String>

```
public final class Scanner
extends Object
implements Iterator<String>, Closeable
```

A simple text scanner which can parse primitive types and strings using regular expressions.

A Scanner breaks its input into tokens using a delimiter pattern, which by default matches whitespace. The resulting tokens may then be converted into values of different types using the various next methods.

For example, this code allows a user to read a number from System.in:

```
Scanner sc = new Scanner(System.in);
int i = sc.nextInt();
```

As another example, this code allows long types to be assigned from entries in a file myNumbers:

```
Scanner sc = new Scanner(new File("myNumbers"));
while (sc.hasNextLong()) {
    long aLong = sc.nextLong();
}
```

The scanner can also use delimiters other than whitespace. This example reads several items in from a string:

Veremos que disponemos de una serie de métodos que podemos utilizar como se muestra en el ejemplo.

No pretendemos que se entienda de momento como funciona, esto se explicara más adelante.

Esta librería nos va a permitir leer caracteres y números por teclado. Podemos utilizar los métodos que se muestra en la lista, aunque para ello debemos hacer un “import” y declarar una variable “teclado” como se muestra en el ejemplo.

BigInteger	nextBigDecimal()	Scans the next token of the input as a BigDecimal .
BigInteger	nextBigInteger()	Scans the next token of the input as a BigInteger .
BigInteger	nextBigInteger(int radix)	Scans the next token of the input as a BigInteger .
boolean	nextBoolean()	Scans the next token of the input into a boolean value and returns that value.
byte	nextByte()	Scans the next token of the input as a byte .
byte	nextByte(int radix)	Scans the next token of the input as a byte .
double	nextDouble()	Scans the next token of the input as a double .
float	nextFloat()	Scans the next token of the input as a float .
int	nextInt()	Scans the next token of the input as an int .
int	nextInt(int radix)	Scans the next token of the input as an int .
String	nextLine()	Advances this scanner past the current line and returns the input that was scanned.
long	nextLong()	Scans the next token of the input as a long .
long	nextLong(int radix)	Scans the next token of the input as a long .
short	nextShort()	Scans the next token of the input as a short .

Línea 1: Obligatorio.

Línea 7: Obligatorio.

Línea 16: Solicito por pantalla un dato

Línea 17: El sistema espera a que entre un dato por pantalla y lo asigna a x.

nextFloat – Lee un float

nextInt – Lee un int

nextDouble – Lee un doublé

nextLine – Lee una línea.

.....

```
OJ_Ejercicio3.java
1 import java.util.Scanner;
2
3 class OJ_Ejercicio3 {
4
5     public static void main(String[] args) {
6
7         Scanner teclado = new Scanner(System.in);
8
9         //Declaraciones
10
11         float x, y, z;
12         float fMedia;
13
14         //Imprimimos en pantalla
15
16         System.out.println("Introduce un numero x: ");
17         x = teclado.nextFloat();
18         System.out.println("\nIntroduce un numero y: ");
19         y = teclado.nextFloat();
20         System.out.println("\nIntroduce un numero z: ");
21         z = teclado.nextFloat();
22
23         fMedia = (x+y+z) / 3;
24         System.out.println("La media es:"+fMedia);
25
26     } //main
27 } //class
```