

# **Boletín de Ejercicios Básicos en Java**

Programación

Universidad Carlos III de Madrid

Curso 2015-2016

Dr. Jose María Alvarez Rodríguez

Octubre 2015



# Índice general

1. Introducción	1
2. Ejemplos básicos en Java	3
3. Ejemplos de sentencias condicionales y bucles en Java	13
4. Ejemplos de bucles anidados en Java	31



# Listings

2.1. Detectar si un año es bisiesto. . . . .	3
2.2. Convertir coordenadas cartesianas a polares. . . . .	4
2.3. Calcular el número de día de la semana dada una fecha. . . . .	4
2.4. Calcular la distancia euclídea entre dos puntos. . . . .	5
2.5. Ejemplos de uso de Math API. . . . .	6
2.6. Convertir grados Celsius a Fahrenheit. . . . .	6
2.7. Intercambiar el valor de dos variables. . . . .	7
2.8. Calcular el valor mayor de dos números reales. . . . .	8
2.9. Calcular el valor mayor de tres números enteros. . . . .	9
2.10. Calcular el valor mayor de tres números reales. . . . .	9
2.11. Calcular la media de dos números. . . . .	10
2.12. Calcular la sensación térmica. . . . .	11
2.13. Calcular la suma de dos números. . . . .	12
3.1. Contar los divisores de un número. . . . .	13
3.2. Contar números impares hasta un número determinado. . . . .	14
3.3. Calcular el cubo de los primeros veinte números naturales uti- lizando Math API de Java. . . . .	16
3.4. Calcular el factorial de un número. . . . .	17
3.5. Mostrar la secuencia de Fibonacci. . . . .	17
3.6. Calcular la potencia de $a^b$ . . . . .	18
3.7. Mostrar los números pares hasta un número determinado uti- lizando diferentes tipos de bucles. . . . .	19
3.8. Mostrar los números naturales hasta un número determinado utilizando diferentes tipos de bucles. . . . .	20
3.9. Mostrar los números naturales de forma descendente utilizan- do diferentes tipos de bucles. . . . .	21
3.10. Detectar si un número natural es un número de Armstrong. . . . .	22
3.11. Calcular el número combinatorio. . . . .	23
3.12. Detectar si un número natural es un palíndromo. . . . .	25
3.13. Detectar si un número natural es número perfecto. . . . .	26
3.14. Detectar si un número natural es número primo. . . . .	26

3.15. Sumar $n$ números enteros. . . . .	27
3.16. Sumar los primeros 20 $n$ números naturales. . . . .	28
4.1. Dibujar un árbol de navidad con *. . . . .	31
4.2. Dibujar una cruz con *. . . . .	32
4.3. Dibujar la diagonal inferior de una matriz ficticia con *. . . . .	33
4.4. Dibujar la diagonal superior de una matriz ficticia con *. . . . .	34
4.5. Dibujar un diamante de radio $N$ con *. . . . .	35
4.6. Mostrar la tabla de multiplicar del 1 al 10. . . . .	36
4.7. Mostrar los primeros 100 números primos. . . . .	37
4.8. Mostrar la tabla de divisores para un determinado número. . . . .	38
4.9. Mostrar la siguiente figura (Wedge of Stars). . . . .	39

# Capítulo 1

## Introducción

En este documento se recopilan ejemplos de Programación en Java para su posible impresión.

Es importante resaltar que se deberían seguir los siguientes pasos para abordar los problemas.

1. Analizar y entender el problema. ¿Existe algún modelo formal?
2. Diseñar una solución al problema.
3. Implementar el diseño propuesto.
4. Test: verificar y validar los resultados. Traza y depuración.
5. Documentar la solución.
6. Refinar, refactor” del código fuente (“mejora”).

También es importante destacar:

1. Todos los ejemplos están disponibles en el repositorio: <https://github.com/chemaar/java-programming-course/>.
2. Se presenta una solución al problema planteado **pero no es única**.
3. Este documento se actualizará a medida que se impartan más conceptos.





## Capítulo 2

# Ejemplos básicos en Java

A continuación se dispone de una serie de ejemplos básicos en Java (ordenados alfabéticamente) para trabajar con:

- Declaración de variables.
- Operaciones simples.

Listing 2.1: Detectar si un año es bisiesto.

```
package es.uc3m.programacion.basico;

import java.util.Scanner;

/**
 * Escribir un programa para detectar si un año es bisiesto
 *
 *
 * @author chema
 *
 */
public class AnyoBisiesto {

    public static void main(String [] args){
        Scanner mc = new Scanner(System.in);
        int year = 0;
        year = mc.nextInt();
        // divisible by 4
        boolean isLeapYear = (year % 4 == 0);
        // divisible by 4 and not 100
        isLeapYear = isLeapYear && (year % 100 != 0);
```

```
        // divisible by 4 and not 100 unless divisible by
        // 400
        isLeapYear = isLeapYear || (year % 400 == 0);
        System.out.println(isLeapYear);
    }
}
```

Listing 2.2: Convertir coordenadas cartesianas a polares.

```
package es.uc3m.programacion.basico;
/**
 * Escribir un programa que calcule los valores de: r y
 * theta, necesarios para transformar unas coordenadas
 * cartesianas (x,y) a polares.
 *
 * @author chema
 */
public class CartesianoAPolar {
    public static void main(String[] args) {
        double x = 2.0;
        double y = 3.0;

        double r      = Math.sqrt(x*x + y*y);
        double theta = Math.atan2(y, x);

        System.out.println("r      = " + r);
        System.out.println("theta = " + theta);
    }
}
```

Listing 2.3: Calcular el número de día de la semana dada una fecha.

```
package es.uc3m.programacion.basico;
/**
 * Escribir un programa que dado un mes (m), día (d) y año
 * (y) muestre por pantalla
```

```

    * el número de día de la semana.
    *
    * Los meses comienzan en 1 hasta 12.
    * Los días comienzan en 0 (domingo).
    *
    * Se debería aplicar la siguiente fórmula:
    *
    *       $y0 = y - (14 - m) / 12$ 
    *       $x = y0 + y0/4 - y0/100 + y0/400$ 
    *       $m0 = m + 12 * ((14 - m) / 12) - 2$ 
    *       $d = (d + x + (31*m0)/12) \bmod 7$ 
    *
    * @author chema
    */
    public class DiaDeLaSemana {
        public static void main(String[] args) {
            int m = 10;
            int d = 5;
            int y = 2015;

            int y0 = y - (14 - m) / 12;
            int x = y0 + y0/4 - y0/100 + y0/400;
            int m0 = m + 12 * ((14 - m) / 12) - 2;
            int d0 = (d + x + (31*m0)/12) % 7;

            System.out.println(d0);
        }
    }

```

Listing 2.4: Calcular la distancia euclídea entre dos puntos.

```

package es.uc3m.programacion.basico;
/**
 * Escribir un programa que calcule la distancia entre dos
 * puntos.
 *
 * @author chema
 */
public class DistanciaEuclidea {

    public static void main(String [] args){

```

```
        double x1 = 2.0;
        double y1 = 3.0;
        double x2 = 4.0;
        double y2 = 5.0;

        double distancia = Math.sqrt(Math.pow(x1-
            x2,2) + Math.pow(y1-y2,2));

        System.out.println("Distancia = " +
            distancia);
    }
}
```

Listing 2.5: Ejemplos de uso de Math API.

```
package es.uc3m.programacion.basico;

import java.util.Scanner;

public class EjemplosMathAPI {

    public static void main(String [] args){
        Scanner mc = new Scanner(System.in);
        //Max
        System.out.println(Math.max(2, 3));
        //Min
        System.out.println(Math.min(2, 3));
        //Abs
        System.out.println(Math.abs(-2));
        //Pow
        System.out.println(Math.pow(2, 3));
        //Sqrt
        System.out.println(Math.sqrt(144));
    }
}
```

Listing 2.6: Converstir grados Celsius a Fahrenheit.

```

package es.uc3m.programacion.basico;
/**
 * Escribir un programa que transforme una temperatura
 * expresada en grados Fahrenheit a Celsius.
 *
 * Se debería aplicar la siguiente fórmula:
 *
 *  $T(C) = (T(F) - 32) * 5/9$ 
 *
 * @author chema
 */
public class FahrenheitACelsius {

    public static void main(String[] args) {
        double temperatura = 3.0;
        temperatura = ((temperatura - 32)*5)/9;
        System.out.println("Temperatura en Celsius
            = " + temperatura);

    }

}

```

Listing 2.7: Intercambiar el valor de dos variables.

```

package es.uc3m.programacion.basico;

import java.util.Scanner;

public class IntercambiarVariables {

    public static void main(String [] args){
        Scanner mc = new Scanner(System.in);
        int a,b;
        int temp;
        a = mc.nextInt();
        b = mc.nextInt();
        System.out.println("Los valores (antes de
            intercambiar) son: "+a+", "+b);
        temp = a;
        a = b;
        b = temp;
    }
}

```

```

    }
    System.out.println("Los valores (después de
        intercambiar) son: "+a+", "+b);
}

```

```
Listing 2.8: Calcular el valor mayor de dos números reales.
```

```
package es.uc3m.programacion.basico;

import java.util.Scanner;

/**
 * Escribir un programa para mostrar por pantalla para
 *      mostrar el mayor valor de 2 números reales.
 *
 * @author chema
 *
 */
public class MayorDeDosReales {

    public static void main(String [] args){
        Scanner mc = new Scanner(System.in);
        float op1, op2;
        op1 = mc.nextFloat();
        op2 = mc.nextFloat();

        // Returns:
        //           the value 0 if f1 is numerically
        //           equal to f2; a value less than 0 if f1 is numerically
        //           less than f2;
        //           and a value greater than 0 if f1 is
        //           numerically greater than f2.
        int compare = Float.compare(op1, op2);
        if (compare == 0){
            System.out.println("Los valores son
                               iguales.");
        }else if(compare < 0){
            System.out.println("El valor mayor
                               es: "+op2);
        }else {
            System.out.println("El valor mayor
                               es: "+op1);
        }
    }
}
```

```

package es.uc3m.programacion.basico;

import java.util.Scanner;

/**
 * Escribir un programa para mostrar por pantalla para
 * mostrar el mayor valor de 2 números reales.
 *
 * @author chema
 *
 */
public class MayorDeDosReales {

    public static void main(String [] args){
        Scanner mc = new Scanner(System.in);
        float op1, op2;
        op1 = mc.nextFloat();
        op2 = mc.nextFloat();
        Returns:
        // the value 0 if f1 is numerically
        // equal to f2; a value less than 0 if f1 is numerically
        // less than f2;
        // and a value greater than 0 if f1 is
        // numerically greater than f2.
        int compare = Float.compare(op1, op2);
        if (compare == 0){
            System.out.println("Los valores son
            iguales.");
        }else if(compare < 0){
            System.out.println("El valor mayor
            es: "+op2);
        }else {
            System.out.println("El valor mayor
            es: "+op1);
        }
    }
}

```

```
}
```

Listing 2.9: Calcular el valor mayor de tres números enteros.

```
package es.uc3m.programacion.basico;

import java.util.Scanner;

/**
 * Escribir un programa para mostrar por pantalla para
 * mostrar el mayor valor de 3 números enteros.
 *
 * @author chema
 */
public class MayorDeTresEnteros {

    public static void main(String [] args){
        Scanner mc = new Scanner(System.in);
        int op1, op2, op3;
        op1 = mc.nextInt();
        op2 = mc.nextInt();
        op3 = mc.nextInt();
        if (op1 > op2 && op1 > op3){
            System.out.println("El valor mayor
                               es: "+op1);
        } else if (op2 > op1 && op2 > op3){
            System.out.println("El valor mayor
                               es: "+op2);
        } else {
            System.out.println("El valor mayor
                               es: "+op3);
        }
    }
}
```

Listing 2.10: Calcular el valor mayor de tres números reales.

```
package es.uc3m.programacion.basico;

import java.util.Scanner;

/**
 * Escribir un programa para mostrar por pantalla para
 * mostrar el mayor valor de 3 números reales.
 *
 * @author chema
 *
 */
public class MayorDeTresReales {

    public static void main(String [] args){
        Scanner mc = new Scanner(System.in);
        float op1, op2, op3;
        op1 = mc.nextFloat();
        op2 = mc.nextFloat();
        op3 = mc.nextFloat();
        if (Float.compare(op1, op2)>0 && Float.
            compare(op1, op3)>0){
            System.out.println("El valor mayor
                               es: "+op1);
        }else if (Float.compare(op2, op3)>0 &&
            Float.compare(op2, op3)>0){
            System.out.println("El valor mayor
                               es: "+op2);
        }else{
            System.out.println("El valor mayor
                               es: "+op3);
        }
    }
}
```

Listing 2.11: Calcular la media de dos números.

```
package es.uc3m.programacion.basico;

import java.util.Scanner;
```



```

/**
 * Escribir un programa que realice la media de dos números
 * reales solicitados al usuario.
 *
 * @author chema
 *
 */
public class MediaDosNumeros {

    public static void main(String [] args){
        Scanner mc = new Scanner(System.in);
        float op1, op2;
        float average;
        op1 = mc.nextFloat();
        op2 = mc.nextFloat();
        average = (op1 + op2) / 2;
        System.out.println("La media es: "+average)
        ;
    }
}

```

Listing 2.12: Calcular la sensación térmica.

```

package es.uc3m.programacion.basico;

/**
 * Escribir un programa que calcule la sensación térmica
 * siguiendo la siguiente fórmula.
 *
 * Se deberá aplicar la siguiente fórmula en función de la
 * temperatura y la velocidad:
 *
 *  $T_{s} = 13,12 + 0,6215 * T - 11,37 * V^{0,16} + 0,3965 * T * V^{0,16}$ 
 *
 * @author chema
 *
 */
public class SensacionTermica {
    public static void main(String[] args) {
        double t = 10.0; //Temperatura del aire
        double v = 5.0; //Velocidad del aire
        double ts = 13.12 + 0.6215*t - (11.37*Math.
            pow(v, 0.16)) + (0.3965*t* Math.pow(v,
            0.16));
    }
}

```

```
        System.out.println("Temperatura = " + t);
        System.out.println("Velocidad del tiempo = " + v);
        System.out.println("Sensación térmica = " + ts);
    }
}
```

Listing 2.13: Calcular la suma de dos números.

```
package es.uc3m.programacion.basico;

import java.util.Scanner;

/**
 * Escribir un programa que sume dos números enteros y
 * muestre el resultado por pantalla.
 *
 * @author chema
 */
public class SumaDosNumeros {

    public static void main(String [] args){
        Scanner mc = new Scanner(System.in);
        int op1, op2, result;
        op1 = mc.nextInt();
        op2 = mc.nextInt();
        result = op1 + op2;
        System.out.println("El resultado es: "+
            result);
    }
}
```

## Ejemplos de sentencias condicionales y bucles en Java

A continuación se dispone de una serie de ejemplos básicos en Java (ordenados alfabéticamente) para trabajar con:

- Sentencias condicionales simples.
- Bucles simples.

Listing 3.1: Contar los divisores de un número.

```
package es.uc3m.programacion.bucles;

import java.util.Scanner;

/**
 * Escribir un programa para calcular un número máximo de
 * divisores de un número introducido por el usuario.
 *
 * @author chema
 */
public class ContarDivisores {
    public static void main(String [] args){
        Scanner mc = new Scanner(System.in);
        int MAX_TESTED = 1000000;
        int n;
        int testDivisor;
        int divisorCount;
        int numberTested;
```

```

        n = mc.nextInt();

        divisorCount = 0;
        numberTested = 0;
        for (testDivisor = 1; testDivisor <= n;
            testDivisor++) {
            if (n % testDivisor == 0 ) {
                divisorCount++;
            }
            numberTested++;
            if (numberTested == MAX_TESTED) {
                numberTested = 0;
            }
        }
        System.out.println("El número de divisores
            de " + n + " es " + divisorCount);
    }
}

```

Listing 3.2: Contar números impares hasta un número determinado.

```

package es.uc3m.programacion.bucles;

import java.util.Scanner;

/**
 * Escribir un programa para contar los números impares en
 * los primeros 20 naturales utilizando diferentes bucles.
 *
 * @author chema
 */
public class ContarNumerosImpares {

    public static void main(String [] args){
        int top = 20;
        int i = 0;
        int counter = 0;
        Scanner mc = new Scanner(System.in);

        do{

```

```

        if (i%2 != 0){
            counter++;
        }
        i = i + 1; //i++
    }while(i<=top);
    System.out.println("Encontrados "+counter+"
        números impares.");

    counter = 0;
    for(int j = 0; j<=top; j++){
        if (j%2 != 0){
            counter++;
        }
    }
    System.out.println("Encontrados "+counter+"
        números impares.");

    int k = 0;
    counter = 0;
    while (k<=top){
        if (k%2 == 0){
            counter++;
        }
        k = k+1; //k++
    }
    System.out.println("Encontrados "+counter+"
        números impares.");

    k = 0;
    counter = 0;
    while (k++<=top){
        if (k%2 != 0){
            counter++;
        }
    }
    System.out.println("Encontrados "+counter+"
        números impares.");
}

```

```
}
```

Listing 3.3: Calcular el cubo de los primeros veinte números naturales utilizando Math API de Java.

```
package es.uc3m.programacion.bucles;

import java.util.Scanner;

/**
 * Escribir un programa calcular el cubo de los primeros 20
 * números naturales utilizando diferentes bucles
 * y el API de Java Math.
 *
 * @author chema
 */
public class CuboNumerosMathAPI {
    public static void main(String [] args){
        int top = 20;
        int i = 0;
        Scanner mc = new Scanner(System.in);
        do{
            System.out.println(Math.pow(i, 3));
            i = i + 1; //i++
        }while(i<=top);

        for(int j = 0; j<=top; j++){
            System.out.println(Math.pow(j, 3));
        }

        int k = 0;
        while (k<=top){
            System.out.println(Math.pow(k, 3));
            k = k+1; //k++
        }
    }
}
```

Listing 3.4: Calcular el factorial de un número.

```

package es.uc3m.programacion.bucles;

import java.util.Scanner;

/**
 * Escribir un programa para implementar la función
 * factorial de un número introducido por el usuario.
 *
 * @author chema
 */
public class Factorial {
    public static void main(String [] args){
        Scanner mc = new Scanner(System.in);
        int op1;
        int fact = 1;
        op1 = mc.nextInt();
        //fact(n) = n * fact(n-1);
        if (op1 < 0){
            System.out.println("No existe el
                               factorial de: "+op1);
        }else{
            for (int i = 1; i<=op1; i++){
                fact = fact * i;
            }
            System.out.println("El factorial de
                               : "+op1+" es "+fact);
        }
    }
}

```

Listing 3.5: Mostrar la secuencia de Fibonacci.

```

package es.uc3m.programacion.bucles;

import java.util.Scanner;

/**

```

```

* Escribir un programa para generar la secuencia de
* Fibonacci para un número n.
*
* @author chema
*
*/
public class Fibonacci {
    public static void main(String [] args){
        Scanner mc = new Scanner(System.in);
        int numbersToGenerate;
        int fn1, fn2, fcurrent, temp;
        //fibonacci(n) =
        //
        //                                fibonacci
        //    (0) = 0
        //
        //                                fibonacci
        //    (1) = 1
        //
        //                                fibonacci (
        //    n) = fibonacci(n-1) + fibonacci (n-2)
        numbersToGenerate = mc.nextInt();
        fn2 = 0;
        fn1 = 1;
        System.out.println(fn2);
        System.out.println(fn1);
        for (int i = 2; i < numbersToGenerate; i++)
        {
            fcurrent = fn1 + fn2;
            temp = fn1;
            fn1 = fcurrent;
            fn2 = temp;
            System.out.println(fcurrent);
        }
    }
}

```

Listing 3.6: Calcular la potencia de  $a^b$ .

```

package es.uc3m.programacion.bucles;

import java.util.Scanner;

/**
* Escribir un programa para implementar la función
* potencia.

```



```

*
* @author chema
*
*/
public class MiPotencia {

    public static void main(String [] args){
        Scanner mc = new Scanner(System.in);
        double base = 0;
        int exponent = 0;
        double value = 1;
        base = mc.nextDouble();
        exponent = mc.nextInt();
        for (int i = 0; i<exponent;i++){
            value = value * base;
        }
        System.out.println("La potencia de "+base+"
            elevado a "+exponent+" es: "+value);
        System.out.println("Using the Java API: "+
            Math.pow(base, exponent));
    }
}

```

Listing 3.7: Mostrar los números pares hasta un número determinado utilizando diferentes tipos de bucles.

```

package es.uc3m.programacion.bucles;

import java.util.Scanner;

/**
 * Escribir un programa para mostrar los números pares en
 * los primeros 20 naturales utilizando diferentes bucles.
 *
 * @author chema
 *
 */
public class MostrarNumerosPares {

    public static void main(String [] args){
        int top = 20;
        int i = 0;
    }
}

```

```

Scanner mc = new Scanner(System.in);

do{
    if (i%2 == 0){
        System.out.println(i);
    }
    i = i + 1; //i++
}while(i<=top);

for(int j = 0; j<=top; j++){
    if (j%2 == 0){
        System.out.println(j);
    }
}

int k = 0;
while (k<=top){
    if (k%2 == 0){
        System.out.println(k);
    }
    k = k+1; //k++
}

while (k++<=top){
    if (k%2 == 0){
        System.out.println(k);
    }
}

}

}

```

Listing 3.8: Mostrar los números naturales hasta un número determinado utilizando diferentes tipos de bucles.

```

package es.uc3m.programacion.bucles;

import java.util.Scanner;

```

```

/**
 * Escribir un programa para mostrar los primeros 20 números naturales con distintos bucles.
 *
 * @author chema
 */
public class MostrarVeinteNaturales {
    public static void main(String [] args){
        int top = 20;
        int i = 0;
        Scanner mc = new Scanner(System.in);
        do{
            System.out.println(i);
            i = i + 1; //i++
        }while(i<=top);

        for(int j = 0; j<=top; j++){
            System.out.println(j);
        }

        int k = 0;
        while (k<=top){
            System.out.println(k);
            k = k+1; //k++
        }
    }
}

```

Listing 3.9: Mostrar los números naturales de forma descendente utilizando diferentes tipos de bucles.

```

package es.uc3m.programacion.bucles;

import java.util.Scanner;

/**
 * Escribir un programa para mostrar los primeros 20 números naturales con distintos bucles en forma descendente.
 *
 * @author chema
 */

```

```

    */
    public class MostrarVeinteNaturalesDescendente {

        public static void main(String [] args){
            Scanner mc = new Scanner(System.in);
            int i = 20;
            do{
                System.out.println(i);
                i = i-1 ; //i—
            }while(i>=0);

            for(int j = 20; j>=0; j--){
                System.out.println(j);
            }

            int k = 20;
            while (k>=0){
                System.out.println(k);
                k = k-1; //k—
            }

        }

    }

```

Listing 3.10: Detectar si un número natural es un número de Armstrong.

```

package es.uc3m.programacion.bucles;

import java.util.Scanner;

/**
 *
 * Escribir un programa que detecte si un número entero es
 * un número de Armstrong.
 *
 * "An Armstrong number of three digit is a number whose
 * sum of cubes of its digit is equal
 * to its number."

```

```

*
* 153 = 1^3+5^3+3^3 o 1+125+27=153
*
* http://en.wikipedia.org/wiki/Narcissistic\_number
*
* @author Chema
*
*
*/
public class NumeroArmstrong {

    public static void main(String [] args){
        Scanner mc = new Scanner(System.in);
        int numero = 0;
        int resultado, numero_inicial, resto;
        numero = mc.nextInt();
        numero_inicial = numero;
        //FIXME: Chequear que el número este entre
        [100-999]
        resultado = 0;
        while(numero != 0){
            resto = numero%10;
            resultado = resultado + resto*resto*resto;
            numero = numero/10;
        }

        System.out.println("El número "+numero+" es
            un número de Armstrong: "+(
                numero_inicial==resultado));
    }
}

```

Listing 3.11: Calcular el número combinatorio.

```

package es.uc3m.programacion.bucles;

import java.util.Scanner;

import java.util.Scanner;

/**
 * Escribir un programa calcule un número combinatorio.
 *

```

```

* Ver definición: http://en.wikipedia.org/wiki/Combinatorial\_number\_system
* @author chema
*
*/
public class NumeroCombinatorio {

    //
    public static void main(String[] args) {
        Scanner mc = new Scanner(System.in);
        int m = mc.nextInt();
        int n = mc.nextInt();
        int factorialmn = 1, factorialn = 1,
            factorialm = 1;
        if (m < n) {
            System.out.println("M must be >= n");
            System.exit(0);
        }
        //Fact(m)
        for (int i = m; i > 0; i--) {
            if (m == 0 || m == 1) {
                factorialm = 1;
                break;
            }
            factorialm = i * factorialm;
        }

        //Fact(n)
        for (int i = n; i > 0; i--) {
            if (n == 0 || n == 1) {
                factorialn = 1;
                break;
            }
            factorialn = i * factorialn;
        }

        //Fact(m-n)
        for (int i = m - n; i > 0; i--) {
            if (m - n == 0 || m - n == 1) {
                factorialmn = 1;
                break;
            }
            factorialmn = i * factorialmn;
        }
    }
}

```

```

        System.out.println("El resultado es: "+ (
            factorialm / (factorialn * factorialmn))
        );
    }
}

```

Listing 3.12: Detectar si un número natural es un palíndromo.

```

package es.uc3m.programacion.bucles;

import java.util.Scanner;

/**
 * Escribir un programa para detectar si un número entero
 * es un palíndromo (capícua).
 *
 * @author chema
 */
public class NumeroPalindromo {

    public static void main(String [] args){
        Scanner mc = new Scanner(System.in);
        int number;
        int palindrome, reverse;
        number = mc.nextInt();
        palindrome = number;
        reverse = 0;
        while (palindrome != 0) {
            int remainder = palindrome % 10;
            reverse = reverse * 10 + remainder;
            palindrome = palindrome / 10;
        }
        System.out.println("El número "+number+" es
            palíndromo: "+(number==reverse));
    }
}

```

Listing 3.13: Detectar si un número natural es número perfecto.

```
package es.uc3m.programacion.bucles;

/**
 * Escribir un programa que dado un número entero detecte
 * si es un número perfecto.
 *
 * Un número perfecto es aquel que es igual a la suma de
 * todos sus divisores propios positivos.
 *
 * 6 es un número perfecto: 6=3+2+1
 *
 * @author chema
 */
public class NumeroPerfecto {

    public static void main(String [] args){
        int N = 6;
        int i = 1;
        int divisor;
        int calculado = 0;
        while(i<N){
            if(N%i == 0){
                calculado = calculado + i;
            }
            i++;
        }
        System.out.println("El número "+N+" es
            perfecto: "+(calculado==N));
    }
}
```

Listing 3.14: Detectar si un número natural es número primo.

```
package es.uc3m.programacion.bucles;

import java.util.Scanner;

/**
```



```

    * Escribir un programa para detectar si un número entero
    * es primo.
    *
    * @author chema
    *
    */
    public class NumeroPrimo {

        public static void main (String [] args){
            Scanner mc = new Scanner(System.in);
            int number = 0;
            boolean isPrime = true;
            number = mc.nextInt();
            //FIXME: Chequear > 0
            int i = 2;
            while(isPrime && i!=number){
                if(number%i == 0){
                    isPrime = false;
                }
                i++;
            }
            System.out.println("El número "+number+" es
                                primo: "+isPrime);
        }
    }

```

Listing 3.15: Sumar  $n$  números enteros.

```

package es.uc3m.programacion.bucles;

import java.util.Scanner;

/**
 * Escribir un programa que sume hasta n números enteros
 * solicitados al usuario.
 *
 * @author chema
 *
 */
public class SumarNEnteros {

```

```

    public static void main(String [] args){
        Scanner mc = new Scanner(System.in);
        int numberToRead = 0;
        numberToRead = mc.nextInt();
        if(numberToRead>0){
            int value = 0;
            int sum = 0;
            for(int i = 0; i<numberToRead; i++){
                System.out.println("
                    Introduza un número...")
                ;
                value = mc.nextInt();
                sum = sum + value;
            }
            System.out.println("La suma es: "+
                sum);
        }
    }
}

```

Listing 3.16: Sumar los primeros 20  $n$  números naturales.

```

package es.uc3m.programacion.bucles;

import java.util.Scanner;

/**
 * Escribir un programa para sumar los primeros 20 números
 * naturales
 *
 * @author chema
 */
public class SumarVeinteNaturales {

    public static void main(String [] args){
        int top = 20;
        int i = 0;
        int sum = 0;
        Scanner mc = new Scanner(System.in);

        sum = 0;
    }
}

```

```
do{
    sum = sum + i;
    i = i + 1; //i++
}while(i<=top);
System.out.println("Suma: "+sum);

sum=0;
for(int j = 0; j<=top; j++){
    sum = sum + j;
}
System.out.println("Suma: "+sum);

int k = 0;
sum = 0;
while (k<=top){
    sum = sum + k;
    k = k+1; //k++
}
System.out.println("Suma: "+sum);
}

}
```



# Ejemplos de bucles anidados en Java

A continuación se dispone de una serie de ejemplos básicos en Java (ordenados alfabéticamente) para trabajar con:

- Bucles anidados.

```
Listing 4.1: Dibujar un árbol de navidad con *.
package es.uc3m.programacion.bucles.anidados;

import java.io.IOException;

import java.util.Scanner;

/**
 * Escribir un programa para mostrar por pantalla la
 * siguiente figura.
 *
 *
 *      *
 *
 *      ***
 *
 *      *****
 *
 *      *********
 *
 *      *********
 *
 *      *****
 *
 *      ***
 *
 *      ***

```

```
    /**
    ***
    */
    public class ArbolNavidad {
        public static void main(String [] args) throws
        IOException{
            Scanner mc = new Scanner(System.in);
            int baseStars = 15;
            int halfBlankSpaces = 0;
            for (int i = 1; i<baseStars; i=i+2){
                halfBlankSpaces = (baseStars-i)/2;
                for(int j=0; j<halfBlankSpaces;j++)
                {
                    System.out.print(" ");
                }
                for(int k = 0; k<i;k++){
                    System.out.print("*");
                }
                for(int j=0; j<halfBlankSpaces;j++)
                {
                    System.out.print(" ");
                }
                System.out.println("");
            }

            halfBlankSpaces = (baseStars/2)-1;
            for(int i = 0; i<3;i++){
                for(int j=0; j<halfBlankSpaces;j++)
                {
                    System.out.print(" ");
                }
                for(int k=0; k<3;k++){
                    System.out.print("*");
                }
                for(int j=0; j<halfBlankSpaces;j++)
                {
                    System.out.print(" ");
                }
                System.out.println(" ");
            }
        }
    }
```

Listing 4.2: Dibujar una cruz con \*.

```

package es.uc3m.programacion.bucles.anidados;

/**
 * Diseñar y codificar un programa en Java que muestre por
 * pantalla una X de radio N.
 * @author chema
 */
public class Cruz {
    public static void main(String [] args){
        int N = 9;
        for(int i = 0; i<N; i++){
            for(int j=0; j<N; j++){
                if(i==j || (N-i-1==j)){
                    System.out.print("
                        *");
                }else{
                    System.out.print("
                        .");
                }
            }
            System.out.println();
        }
    }
}

```

Listing 4.3: Dibujar la diagonal inferior de una matriz ficticia con \*.

```

package es.uc3m.programacion.bucles.anidados;

/**
 * Diseñar y codificar un programa Java que muestra por
 * pantalla el siguiente patrón de un
 * triángulo en una tabla implícita NxN. Por ejemplo para N
 * =3 la salida sería.
 *
 * * * *
 * . * *
 * . . *
 *
 * @author chema
 */

```

```

*/
public class DiagonalInferior {
    public static void main(String [] args){
        int N = 3;
        for(int i =1; i<=N; i++){
            for(int j =1; j<=N; j++){
                if(i>=j){
                    System.out.print("*
");
                }else{
                    System.out.print(".
");
                }
            }
            System.out.println("");
        }
    }
}

```

Listing 4.4: Dibujar la diagonal superior de una matriz ficticia con \*.

```

package es.uc3m.programacion.bucles.anidados;

/**
 * Diseñar y codificar un programa Java que muestra por
 * pantalla el siguiente patrón de un
 * triángulo en una tabla implícita NxN. Por ejemplo para N
 * =3 la salida sería.
 *
 * * . .
 * * * .
 * * * *
 *
 * @author chema
 */
public class DiagonalSuperior {
    public static void main(String [] args){
        int N = 3;
        for(int i =1; i<=N; i++){
            for(int j =1; j<=N; j++){
                if(i<=j){

```





```

        System.out.
            print("
                *");
    }else{
        System.out.
            print("
                .");
    }
}
System.out.println();
}
} else{
    System.err.println("Introduzca un n
        úmero impar.");
}
}
}
}

```

```
package es.uc3m.programacion.bucles.anidados;
```

```

Listing 4.6: Mostrar la tabla de multiplicar del 1 al 10.
package es.uc3m.programacion.bucles.anidados;

/**
 * Escribir un programa para mostrar la tabla de
 * multiplicar del 1 al 10.
 *
 * @author chema
 *
 */
public class MultiplicationTable {
    public static void main(String[] args) {
        for (int i = 1; i <= 10; i++) {
            for (int j = 1; j <= 10; j++) {
                System.out.print( i * j );
                System.out.print("\t");
            }
            System.out.println();
        }
    }
}

```

Listing 4.7: Mostrar los primeros 100 números primos.

```

package es.uc3m.programacion.bucles.anidados;

import java.util.Scanner;

/**
 * Escribir un programa para mostrar por pantalla los 100
 * primeros números primos.
 *
 * @author chema
 */
public class Primeros100Primos {

    public static void main (String []args){
        Scanner mc = new Scanner(System.in);
        int nprimes = 100;
        int matches = 1; //1 es primo
        int number = 2; //...empezamos en 2...
        boolean isPrime = true;
        int i = 2;
        while (matches<nprimes){
            isPrime = true;
            i = 2;
            while(isPrime && i!=number){
                if(number%i == 0){
                    isPrime = false;
                }
                i++;
            }
            if (isPrime){
                System.out.println(number);
                matches++;
            }
            number++;
        }
    }
}

```

```
package es.uc3m.programacion.bucles.anidados;

/**
 * 1–Diseñar y codificar un programa en Java que muestre
 * por pantalla una tabla de
 * dimensión (1..N) en la cual la celda (i, j) contendrá '*'
 * si i es divisor j o j es divisor
 * de i, de otra forma mostrará '.' . Por ejemplo para N
 * =10 la salida sería:
 *
 *
 *
 *
 *
 *
 *
 *
 *
 *
 *
 *
 *
 *
 */
public class TablaDivisores {

    public static void main(String [] args){
        int N = 10;
        for(int i =1; i<=N; i++){
            for(int j =1; j<=N; j++){
                if(i%j == 0 || j%i == 0){
                    System.out.print("*");
                }else{
                    System.out.print(".");
                }
            }
            System.out.print("\n");
        }
    }
}
```

```

        System.out.println("");
    }
}

```

Listing 4.9: Mostrar la siguiente figura (Wedge of Stars).

```

package es.uc3m.programacion.bucles.anidados;

import java.util.Scanner;

/**
 *
 * Escribir un programa que muestre por pantalla la
 * siguiente figura:
 *
 *      *****
 *      *****
 *      *****
 *      *****
 *      *****
 *      *****
 *
 *
 */
public class WedgeOfStars {

    public static void main(String args[]) {
        Scanner mc = new Scanner(System.in);
        int numberOfStars = 0;
        numberOfStars = mc.nextInt();
        for (int i = numberOfStars; i>0; i--){
            for(int j = 0; j<i; j++){
                System.out.print("*");
            }
            System.out.println("");
        }
    }
}

```

