

Ejercicios Resueltos. Secuencial, Selectivas y Repetitivas

Ejemplo 1: Diseñar un programa en el a partir de dos enteros no negativos, calculemos su suma (+) y producto (*).

Entrada: *¿Qué necesito?* Dos enteros no negativos.

Salida: *¿Qué me proporciona?* La suma y el producto.

```
package org.ip.tema01;

public class Ejercicio01 {

    /**
     * Suma y multiplica dos enteros no negativos
     *
     * @param args
     */
    public static void main(String[] args) {

        int dato1, dato2, suma, producto;
        // PRE: dato1 >= 0 y dato2 >= 0
        dato1 = 7;
        dato2 = 10;
        suma = dato1 + dato2;
        producto = dato1 * dato2;
        System.out.println("La suma es ... " + suma);
        System.out.println("El producto es ... " + producto);
    }
}
```

Ejemplo 2. Diseñar un programa que permita calcular la suma de dos enteros si son positivos y el producto en caso contrario.

Entrada: Dos números enteros.

Salida: Suma o producto.

```
package org.ip.tema01;

public class Ejercicio02 {

    /**
     * Permite calcular la suma de dos enteros si son positivos y el
     * producto en caso contrario
     *
     * @param args
     */
}
```

```
public static void main(String[] args) {  
  
    int dato1, dato2, suma, producto;  
    dato1 = 10;  
    dato2 = -5;  
    if ((dato1 > 0) && (dato2 > 0)) {  
        suma = dato1 + dato2;  
        System.out.println("la suma es ... " + suma);  
    } else {  
        producto = dato1 * dato2;  
        System.out.println("El producto es ... " + producto);  
    }  
}  
}
```

Ejemplo 3. En un centro comercial se han detectado errores en aquellos artículos cuyos códigos están comprendidos entre:

14681 hasta 15681 ambos inclusive

70001 hasta 79999 ambos inclusive

99999 hasta 110110 ambos inclusive

Diseñar un programa que con el código de un artículo muestre por pantalla un mensaje que indique si el código es o no defectuoso.

Entrada: un número de código

Salida: mensaje “Defectuoso” o “No defectuoso”

```
package org.ip.tema01;  
  
public class Ejercicio03 {  
  
    /**  
     * A partir del codigo de un articulo, muestre por pantalla un  
     * mensaje que indica si el codigo es o no defectuoso.  
     * Codigo defectuosos:  
     * 14681 hasta 15681 ambos inclusive  
     * 70001 hasta 79999 ambos inclusive  
     * 99999 hasta 110110 ambos inclusive  
     * @param args  
     */  
    public static void main(String[] args) {  
        int codigo;  
        codigo = 50000; // Ejecutar cambiando el valor a 300000  
  
        if ((codigo >= 14681) && (codigo <= 15681)  
            || (codigo >= 70001) && (codigo <= 79999)  
            || (codigo >= 99999) && (codigo <= 110110)) {  
            System.out.println("CODIGO DEFECTUOSO");  
        } else {  
            System.out.println("CODIGO CORRECTO");  
        }  
    }  
}
```

Ejemplo 4. Una empresa desea contratar a aquellos trabajadores cuya edad sea de al menos 18 años y tengan más de 85 puntos en alguna de las dos pruebas que han realizado. Se debe de realizar un programa que muestre por pantalla un mensaje de “Contratado” o “No contratado” si la edad y las dos calificaciones cumplen las condiciones que se han indicado.

Entrada: Edad y dos calificaciones

Salida: Mensaje “Contratado” o “No contratado”

```
package org.ip.tema01;

public class Ejercicio04 {

    /**
     * Contratacion en una empresa si los trabajadores cumplen:
     * Que la edad sea de al menos 18 anios y
     * que tengan mas de 85 puntos en alguna de las dos pruebas
     * que han realizado
     * @param args
     */
    public static void main(String[] args) {
        int edad, notal, nota2;

        // PRE: edad >= 0 y notal >= 0 y nota2 >= 0
        edad = 18;
        notal = 15;
        nota2 = 90;
        if ((edad >= 18) && ((notal > 85) || (nota2 > 85))) {
            System.out.println("CONTRATADO");
        } else {
            System.out.println("NO CONTRATADO");
        }
    }
}
```

Ejemplo 5. Diseñar un programa que determine si un año es bisiesto. Deberá mostrar un mensaje “Año Bisiesto” o “Año No Bisiesto”.

Análisis del problema. Veamos algunos ejemplos:

1600 \Rightarrow 1600 / 4 = 400 \Rightarrow Bisiesto

1800 \Rightarrow 1800 / 4 = 450 \Rightarrow No bisiesto

1992 \Rightarrow 1992 / 4 = 498 \Rightarrow Bisiesto

1993 \Rightarrow 1993 / 4 = 498.25 \Rightarrow No bisiesto

Entrada: año

Salida: mensaje “Bisiesto” o “No bisiesto”

```
package org.ip.tema01;

public class Ejercicio05 {
```

```
/**
 * Muestra un mensaje en pantalla indicando si un año es
 * bisiesto.
 * @param args
 */
public static void main(String[] args) {

    int año = 1800;
    // PRE: año > 0
    if ((año % 4 == 0) && (año % 100 != 0) || (año % 400 == 0)) {
        System.out.println("AÑO BISIESTO");
    } else {
        System.out.println("AÑO NO BISIESTO");
    }
}
```

Ejemplo 6. Diseña un programa que a partir de un número comprendido entre 1 y 7 muestre por pantalla el día de la semana con el que se corresponde.

Entrada: entero comprendido entre 1 y 7

Salida: mensaje “LUNES” o “MARTES” o ... “DOMINGO”

```
package org.ip.tema01;

public class Ejercicio06 {

    /**
     * A partir de un numero comprendido entre 1 y 7 muestra por
     * pantalla el dia de la semana con el que se corresponde
     * @param args
     */
    public static void main(String[] args) {

        int dia = 9;

        switch (dia) {
            case 1:
                System.out.println("LUNES");
                break;
            case 2:
                System.out.println("MARTES");
                break;
            case 3:
                System.out.println("MIERCOLES");
                break;
            case 4:
                System.out.println("JUEVES");
                break;
            case 5:
                System.out.println("VIERNES");
                break;
            case 6:
                System.out.println("SABADO");
                break;
        }
    }
}
```

```

        case 7:
            System.out.println("DOMINGO");
            break;
        default:
            System.out.println("NO CORRESPONDE A NINGUN DIA DE LA
SEMANA");
            break;
    }
}

```

Ejemplo 7. Dados dos enteros positivos, diseñar un programa que calcule el cociente y el resto haciendo uso exclusivamente de los operadores + y −.

$$\begin{array}{r}
 39 \overline{)11} \\
 \underline{6 3} \\
 17 3
 \end{array}
 \Rightarrow
 \begin{array}{l}
 39 - 11 = 28 \text{ Cociente} = 1 \\
 28 - 11 = 17 \text{ Cociente} = 2 \\
 17 - 11 = 6 \text{ Cociente} = 3
 \end{array}
 \left. \vphantom{\begin{array}{l} 39 \\ 28 \\ 17 \end{array}} \right\} \Rightarrow \begin{cases} \text{Resto} - \text{Divisor} = \text{Resto} \\ y \\ \text{Cociente} = \text{Cociente} + 1 \end{cases}$$

Entrada: Dividendo y divisor, enteros positivos

Salida: Cociente y resto.

Nos debemos de plantear las siguientes preguntas: ¿hay proceso que se repite?, etc.

```

package org.ip.tema01;

public class Ejercicio07 {
    /**
     * @param args
     */
    public static void main(String[] args) {

        int dividendo = 39;
        int divisor = 11;
        int cociente, resto;
        cociente = 0;
        resto = dividendo;

        while (resto >= divisor) {
            resto = resto - divisor;
            cociente = cociente + 1;
        }
        System.out.println("El cociente es " + cociente);
        System.out.println("El resto es " + resto);
    }
}

```

Seguimiento manual de la ejecución del programa

Iteración	Dividendo	Divisor	Cociente	Resto
0	39	11	0	39
1	39	11	1	28
2	39	11	2	17
3	39	11	3	6

No habría sido necesario poner las columnas del **dividendo** y del **divisor** puesto que no cambian en las distintas iteraciones.

Es buena costumbre poner las variables en la tabla en el orden en el que aparecen en el cuerpo del bucle, esto permite hacer un seguimiento más fácil de las iteraciones.

Ejemplo 8. Diseñar un programa que permita realizar el producto de dos enteros positivos utilizando la “*Multiplicación Rusa*”. Para ello se parte de los dos enteros positivos y se multiplica por 2 el multiplicando y se divide entre 2 el multiplicador. Continuamos realizando esas operaciones hasta que el multiplicador vale 1. Se suman todos los multiplicandos correspondientes a multiplicadores impares y este resultado es el producto.

Ejemplo: 37 12 = 444

Multiplicador	Multiplicando	
i 37	12	
18	24	
i 9	48	
4	96	
2	192	
i 1	384	12 + 48 + 384 = 444

Entrada: Multiplicador y multiplicando, enteros positivos

Salida: Producto

```
package org.ip.tema01;

public class Ejercicio08 {

    /**
     * @param args
     */
    public static void main(String[] args) {

        int multiplicador = 37;
        int multiplicando = 12;
        int producto = 0;
        System.out.println("MULTIPLICADOR      MULTIPLICANDO");

        while (multiplicador >= 1) {
            System.out.printf("%7d%18d\n", multiplicador, multiplicando);
            if (multiplicador % 2 != 0) // multiplicador es impar
                producto = producto + multiplicando;
            multiplicador = multiplicador / 2;
            multiplicando = multiplicando * 2;
        }
        System.out.println("El producto es " + producto);
    }
}
```

Ejemplo 9. Diseñar un programa que permita el cálculo del factorial de un valor entero positivo.

n entero positivo $\Rightarrow n! = n * (n-1) * (n-2) * \dots * 2 * 1$

Entrada: entero positivo

Salida: factorial

Veamos varias soluciones para diseñar el bucle:

1.- Primer solución.

Es la solución más intuitiva, en la que utilizamos un único índice (i) en el bucle para hacer el producto.

```
package org.ip.tema01;

public class Ejercicio09V1 {

    /**
     * Calcula el factorial de un entero positivo
     *  $n! = n * (n-1) * (n-2) * \dots * 2 * 1$ 
     * @param args
     */

    public static void main(String[] args) {
        int dato = 9;
        long fact = 1;
        int i = 1;

        while (i <= dato) {
            fact = fact * i;
            i++;
        }
        System.out.println("El factorial de " + dato + " es " + fact);
    }
}
```

2.- Segunda solución.

Usando dos índices (i y j) en bucle para hacer el producto, uno por la derecha y otro por la izquierda.

Ejemplo:

$6! = 6 * 5 * 4 * 3 * 2 * 1$

$j \rightarrow$ | | \times | $\leftarrow i$
 i j

$7! = 7 * 6 * 5 * 4 * 3 * 2 * 1$

$j \rightarrow$ | | | | | $\leftarrow i$
 i=j

```
package org.ip.tema01;

public class Ejercicio09V2 {

    /**
     * Calcula el factorial de un entero positivo haciendo productos por
     * la derecha y por la izquierda  $n! = n \cdot (n-1) \cdot (n-2) \dots 1$ 
     *
     * @param args
     */
    public static void main(String[] args) {
        int dato = 9;
        long fact = 1;
        int i = 1;
        int j = dato;

        while (i < j) {
            fact = fact * i * j;
            i++;
            j--;
        }

        if (i == j) {
            fact = fact * i;
        }
        System.out.println("El factorial de " + dato + " es " + fact);
    }
}
```

Ejemplo 10. Diseñar un programa que muestre las potencias de 2 comprendidas entre 1 y 100000.

```
package org.ip.tema01;

public class Ejercicio10 {

    /**
     * Muestra las potencias de 2 comprendidas entre 1 y 100000
     * @param args
     */
    public static void main(String[] args) {
        int potencia = 1;
        int i = 0;
        int limite = 100000;

        while (potencia <= limite) {
            System.out.printf("%d\t%d\n", i, potencia);
            potencia = potencia * 2;
            i++;
        }
    }
}
```


Ejemplo 11. Diseñar un programa que muestre las tablas de multiplicar del 1 al 10.

```
package org.ip.tema01;

public class Ejercicio11 {

    /**
     * Muestra las tablas de multiplicar del 1 al 10
     * @param args
     */
    public static void main(String[] args) {
        for (int i = 1; i <= 10; i++) {
            System.out.println(" TABLA DEL " + i);
            System.out.println();
            for (int j = 1; j <= 10; j++) {
                System.out.println("      " + i + " X " + j + " = " + i*j);
            }
            System.out.println();
        }
    }
}
```

Ejemplo 12. Diseñar un programa que muestre si un entero positivo (>1) es primo

Definición de nº primo: aquel que es divisible entre el mismo y la unidad.

Entrada: Un nº entero positivo > 1.

Salida: Mensaje nº primo/nº no primo.

Daremos una primera solución aplicando la definición y posteriormente haremos dos mejoras.

```
package org.ip.tema01;

public class Ejercicio12V1 {

    /**
     * Muestra si un entero positivo es primo
     * @param args
     */
    public static void main(String[] args) {
        int dato = 7;
        int divisor = 2;
        boolean primo = true;

        for (divisor = 2; divisor < dato; divisor++) {
            if (dato % divisor == 0)
                primo = false;
        }
        if (primo)
            System.out.println("El numero es primo");
        else
            System.out.println("El numero no es primo");
    }
}
```

Primera mejora: si encontramos un divisor no es necesario seguir probando ya que podemos asegurar que el número ya no es primo. Esto hará que transformemos la estructura **for** en un **while** que permita escapar del bucle en la circunstancia expuesta.

```
package org.ip.tema01;

public class Ejercicio12V2 {

    /**
     * Muestra si un entero positivo es primo
     * @param args
     */
    public static void main(String[] args) {
        int dato = 7;
        int divisor = 2;
        boolean primo = true;

        while (primo && (divisor < dato)) {
            if (dato % divisor == 0)
                primo = false;
            divisor++;
        }
        if (primo)
            System.out.println("El numero es primo");
        else
            System.out.println("El numero no es primo");
    }
}
```

Segunda mejora: Se deja como ejercicio.

- Comprobamos si es divisible entre 2. De no serlo no comprobaremos divisores pares.
- Llegaremos probando divisores hasta \sqrt{dato} . Si hasta ese valor no hemos encontrado ningún divisor, no es necesario seguir comprobando.

Ejemplo 13. Ejecuta y razona la salida del siguiente programa:

```
package org.pi.tema01;
public class OperadoresUnarios {
    public static void main(String[] args) {
        int i = 8;
        int a, b, c;
        System.out.println("\tantes\t durante\t despues");
        i = 8;
        a = i;
        b = i++;
        c = i;
        System.out.println("i++\t" + a + '\t' + b + '\t' + c);
        i = 8;
        a = i;
        b = i--;
        c = i;
        System.out.println("i--\t" + a + '\t' + b + '\t' + c);
        i = 8;
        a = i;
        b = ++i;
        c = i;
        System.out.println("++i\t" + a + '\t' + b + '\t' + c);
        i = 8;
        a = i;
        b = --i;
        c = i;
        System.out.println("--i\t" + a + '\t' + b + '\t' + c);
    }
}
```

Ejemplo 14. Dada la temperatura t (en grados Fahrenheit) y la velocidad del viento v (en millas por hora), desarrolle un programa (**EnfriamientoViento**) que permita calcular el enfriamiento del viento, w (en grados Fahrenheit), usando la fórmula del National Weather Service:

$$w = 35.74 + 0.6216 * t + (0.4275 * t - 35.75) * v^{0.16}$$

Referencia: <https://www.weather.gov/media/epz/wxcalc/windChill.pdf>

```
package org.ip.tema01;

public class EnfriamientoViento {
    public static void main(String[] args) {
        double t = 40;
        double v = 30;
        double w = 35.74 + 0.6215 * t + (0.4275 * t - 35.75)
            * Math.pow(v, 0.16);
        System.out.println();
        System.out.println("Temperatura = " + t + " (grados Fahrenheit)");
        System.out.println("Velocidad del viento = " + v + " (millas por hora)");
        System.out.println();
        System.out.printf("Enfriamiento del viento = %6.2f (grados Fahrenheit)", w);
    }
}
```

Ejemplo 15. Desarrolle un programa que, a partir de las coordenadas de dos puntos, calcule y muestre la ecuación de la recta que pasa por ellos. $y = ax + b$.

Ejemplo de ejecución:



CÁLCULO DE LA ECUACIÓN DE LA RECTA QUE PASA POR DOS PUNTOS

Ecuación de la recta que pasa por (0,1) y (1,4)

$$y = 3.0x + 1.0$$

Se debe comprobar con los valores que se indican en la tabla (puntos por los que pasa la recta (x1, y1) y (x2, y2) y que genere la misma salida que se muestra al lado de cada par de puntos.

(x1, y1)	(x2, y2)	Salida por consola
(1, 2)	(1, 2)	CÁLCULO DE LA ECUACIÓN DE LA RECTA QUE PASA POR DOS PUNTOS Los dos puntos (1,2) y (1,2) COINCIDEN, no se puede obtener la ecuación de la recta
(1, 2)	(1, 1)	CÁLCULO DE LA ECUACIÓN DE LA RECTA QUE PASA POR DOS PUNTOS Ecuación de la recta que pasa por (1,2) y (1,1) $x = 1$
(1, 2)	(2, 2)	CÁLCULO DE LA ECUACIÓN DE LA RECTA QUE PASA POR DOS PUNTOS Ecuación de la recta que pasa por (1,2) y (2,2) $y = 2$
(1, 2)	(2, 4)	CÁLCULO DE LA ECUACIÓN DE LA RECTA QUE PASA POR DOS PUNTOS Ecuación de la recta que pasa por (1,2) y (2,4) $y = 2.0x$
(1, 1)	(2, 4)	CÁLCULO DE LA ECUACIÓN DE LA RECTA QUE PASA POR DOS PUNTOS Ecuación de la recta que pasa por (1,1) y (2,4) $y = 3.0x - 2.0$
(0, 1)	(1, 4)	CÁLCULO DE LA ECUACIÓN DE LA RECTA QUE PASA POR DOS PUNTOS Ecuación de la recta que pasa por (0,1) y (1,4) $y = 3.0x + 1.0$

```

package org.ip.tema01;

public class EcuacionRecta {

    public static void main(String[] args) {

        int x1 = 0;
        int x2 = 1;
        int y1 = 1;
        int y2 = 4;
        double a;
        double b;
        System.out.println();
        System.out.println("CALCULO DE LA ECUACION DE LA RECTA QUE PASA
POR DOS PUNTOS");
        System.out.println();
        if ((x1 == x2) && (y1 == y2)) {
            System.out.println("Los dos puntos (" + x1 + "," +
                + y1 + ") y (" + x2 + "," + y2
                + ") COINCIDEN, no se puede obtener la ecuacion de la
recta");
        } else {
            System.out.println("Ecuacion de la recta que pasa por ("
                + x1 + "," + y1 + ") y (" + x2 + "," + y2 + ") \n");
            if (x1 == x2) {
                System.out.println("x = " + x1);
            } else if (y1 == y2) {
                System.out.println("y = " + y1);
            } else {
                a = (double) (y1 - y2) / (x1 - x2);
                b = y1 - ((double) (y1 - y2) / (x1 - x2)) * x1;
                if (b == 0) {
                    System.out.println("y = " + a + "x ");
                } else if (b > 0) {
                    System.out.println("y = " + a + "x + " + b);
                } else {
                    System.out.println("y = " + a + "x - " + -b);
                }
            }
        }
    }
}

```

Ejemplo 16. Un entero positivo es un *Número Perfecto* si es igual a la suma de todos sus divisores positivos excluyendo el mismo. Por ejemplo, 6 es el primer número perfecto porque sus divisores (excluyendo el mismo) son 1, 2 y 3 cuya suma $1 + 2 + 3 = 6$ coincide con el propio número. Desarrolle un programa que pida un número y muestre por consola si es perfecto o si no lo es.

Ejemplo de ejecución:



```

Introduzca un número entero positivo para saber si es perfecto
28
El número es perfecto

```

```

package org.ip.tema01;

import java.util.Scanner;

public class NumeroPerfecto {
    /**
     * @param args
     */
    public static void main(String[] args) {
        @SuppressWarnings("resource")
        Scanner entrada = new Scanner(System.in);
        System.out.println("Introduzca un numero entero positivo para
saber si es perfecto");
        int numero = entrada.nextInt();
        int suma = 1;
        for (int i = 2; i < numero; i++) {
            if (numero % i == 0) {
                suma += i;
            }
        }
        if (suma == numero)
            System.out.println("El numero es perfecto");
        else
            System.out.println("El numero no es perfecto");
    }
}

```

Ejemplo 17. El número **PI** (π) puede ser calculado mediante la siguiente serie:

$$\pi(i) = 4 \left(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11} + \dots + \frac{(-1)^{i+1}}{2i-1} \right)$$

Implementar un método que genere dicha serie para un i dado (ese i corresponde con la precisión del cálculo de la estimación de π , es decir, cuanto mayor sea el valor de i más precisa será la estimación de π) y devuelva el último valor generado según la serie mostrada en consola. La generación de la serie debe ser la misma que se muestra como salida a continuación y se ha de comprobar con el valor de π que tiene en Math (**Math.PI**). El nombre de la clase y del archivo java debe ser (**EstimatePI**), y la cabecera del método tiene que ser la siguiente:

```
public static double estimationOfPi(long precision)
```

Por último, la función **main** que hay en la clase **EstimatePI** es la que se muestra a continuación, por lo que debe ser muy cuidadoso en la implementación del método **estimationOfPi**

Ejemplo de ejecución:



i	PI(i)
1	4,00000000
101	3,15149340
201	3,14656775
301	3,14491490
401	3,14408642
501	3,14358866
601	3,14325655
701	3,14301919
801	3,14284109
901	3,14270253
Precision =	1000
miPI =	3,142702531161429
Math.PI =	3,141592653589793

```
package org.ip.tema01;
```

```
public class EstimarPI {
```

```
    public static double estimationOfPi(long precision) {
        double valorEstimado = 0.0;
        double sum = 0.0;
        boolean mostrar = true;
        long cantidadMostrar = precision / 10;

        System.out.printf("%-6s\t\t%8s\n", "i" , "PI(i)");
        System.out.println("-----");
        for (long i = 1; i <= precision; i++){
            //if (i % 2 == 0)
            //    sum += -1.0 / ((2.0 * (double)i) - 1.0);
            //else
            //    sum += 1.0 / ((2.0 * (double)i) - 1.0);

            sum += Math.pow(-1.0, (double)i + 1.0) / ((2.0 * (double)i) -
1.0);

            if (mostrar) {
                //System.out.println("i = " + i + ", PI(" + i + ") = " +
4.0 * sum);

                System.out.printf("%-8d\t%10.8f\n", i , 4.0 * sum);
                valorEstimado = 4.0 * sum;
                mostrar = false;
            }
            if (i % cantidadMostrar == 0)
                mostrar = true;
        }

        return valorEstimado;
    }
```

```

public static void main(String[] args) {
    double piEstimatedValue = estimationOfPi(1000);
    System.out.printf("\nmyPI = \t\t%17.15f", piEstimatedValue);
    System.out.printf("\nMath.PI = \t%17.15f", Math.PI);
}
}

```

Ejemplo 18. Haz un programa, que contenga un método recursivo (mostrarInverso), y que permita mostrar un valor entero invertido, es decir si el entero es 12345 mostrará 54321. La cabecera del método debe ser: `public static void mostrarInverso(int valor)`

Ejemplo de ejecución:



```

Introduzca un valor entero
123456
El entero 123456 invertido es 654321

```

```

package org.ip.tema01;

import java.util.Scanner;

public class InvertirEntero {

    @SuppressWarnings("resource")
    public static void main(String[] args) {
        Scanner entrada = new Scanner(System.in);
        System.out.println("Introduzca un valor entero");
        int valor = entrada.nextInt();
        if (valor == 0) {
            System.out.print("El entero " + valor + " invertido es "
                + valor);
        }
        else if (valor > 0) {
            System.out.print("El entero " + valor + " invertido es ");
            mostrarInverso(valor);
        }
        else{
            System.out.print("El entero " + valor + " invertido es -");
            mostrarInverso(-valor);
        }
    }

    public static void mostrarInverso(int valor) {
        if (valor != 0) {
            System.out.print(valor % 10);
            valor = valor / 10;
            mostrarInverso(valor);
        }
    }
}

```