



# Programación

Act01\_01Introducción

Francisco José García Cutillas | 1FPGS\_DAM








## Índice

Ejercicio A1 .....	3
Ejercicio A2 .....	4
Ejercicio A3 .....	6
Ejercicio A4 .....	6
Ejercicio A5 .....	7
Ejercicio A6 .....	7
Ejercicio A7 .....	8
Ejercicio A8 .....	9
Ejercicio A9 .....	10
Ejercicio A10 .....	11
Ejercicio A11 .....	12
Ejercicio A12 .....	13
Ejercicio A13 .....	14
Ejercicio A14 .....	15
A11 con operador ternario .....	15
A13 con operador ternario .....	16
Ejercicio A15 .....	17
Ejercicio A16 .....	18
Ejercicio A17 .....	19
Ejercicio A18 .....	20
Ejercicio A19 .....	21
Ejercicio A20 .....	22

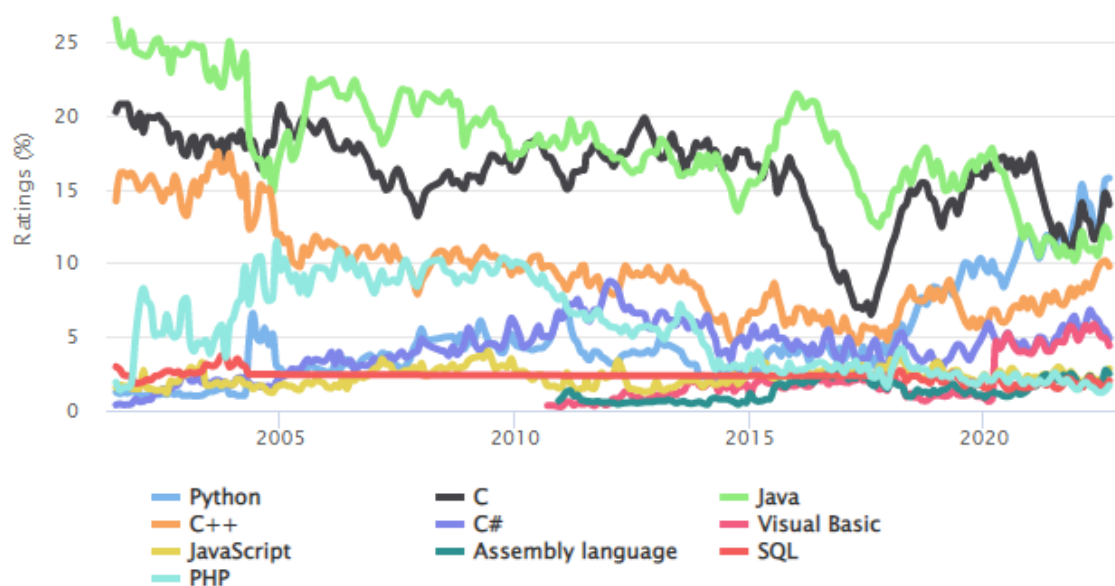
## Ejercicio A1

¿Qué lenguajes son los más utilizados en la actualidad según el índice TIOBE? ¿Han sido siempre dichos lenguajes o ha variado mucho en los últimos años? Además de indicar la información y tus conclusiones, muestra la información en forma de estadística o tabla.

Según el índice TIOBE de septiembre de 2022, el top 10 de los lenguajes de programación más utilizados son Python, C, Java, C++, C#, Visual Basic, JavaScript, Assembly Language, SQL y por último PHP.

Sep 2022	Sep 2021	Change	Programming Language	Ratings	Change
1	2	▲	 Python	15.74%	+4.07%
2	1	▼	 C	13.96%	+2.13%
3	3		 Java	11.72%	+0.60%
4	4		 C++	9.76%	+2.63%
5	5		 C#	4.88%	-0.89%
6	6		 Visual Basic	4.39%	-0.22%
7	7		 JavaScript	2.82%	+0.27%
8	8		 Assembly language	2.49%	+0.07%
9	10	▲	 SQL	2.01%	+0.21%
10	9	▼	 PHP	1.68%	-0.17%

Como podemos observar en la siguiente gráfica, en los últimos años el lenguaje más utilizado mayoritariamente ha sido Java, salvo en alguna ocasión que ha sido desbancado por C. Pero desde hace 5 años Python ha ido creciendo de forma exponencial, hasta colocarse en la primera posición actualmente.



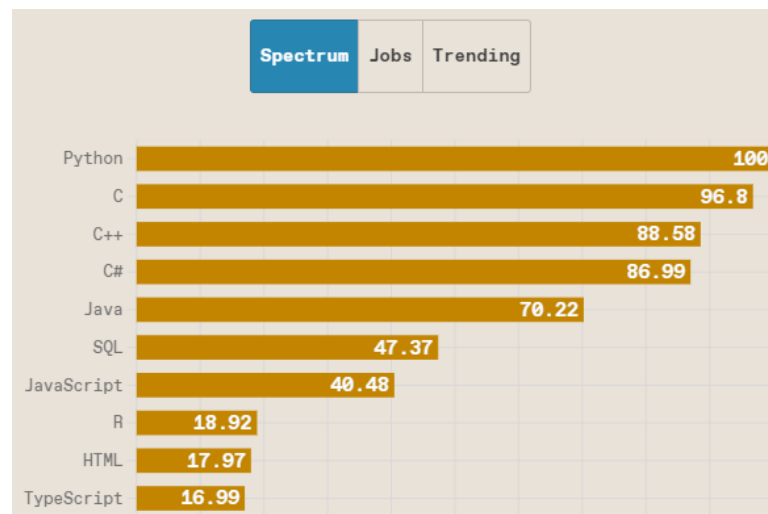
Fuente: <https://www.tiobe.com/tiobe-index/>

## Ejercicio A2

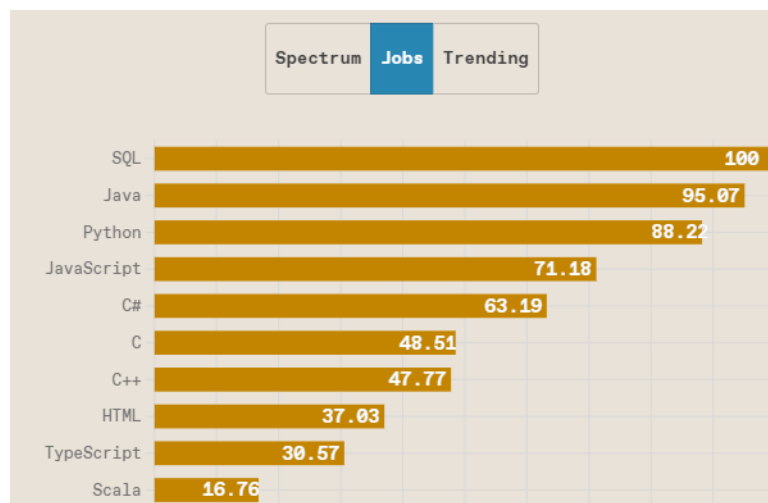
**Investiga sobre algún índice similar al índice TIOBE y comprueba si los resultados que muestra son similares o existen grandes desviaciones respecto a TIOBE.**

Un índice similar a TIOBE es el “*IEEE Spectrum’s Top programming Languages*”, en el cual se realizan tres tipos de sondeos diferentes:

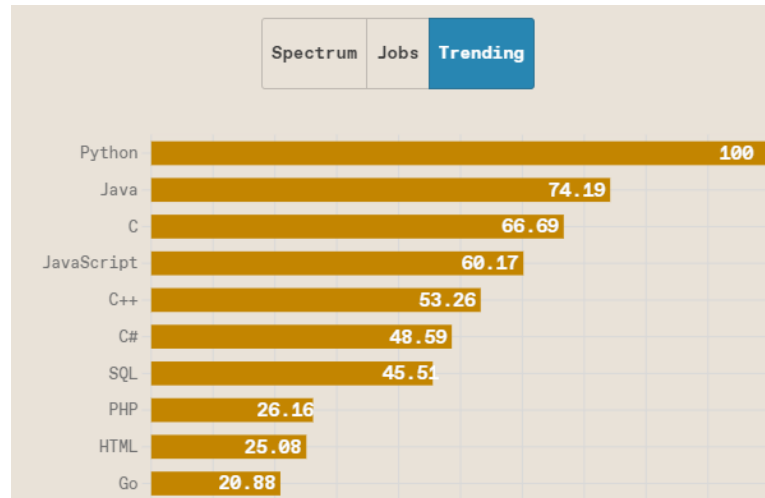
- Spectrum. Este sondeo está basado en la información obtenida de fuentes como GitHub, Google, Stack Overflow, Twitter e IEEE Xplore.



- Jobs. Para este sondeo se basan en la cantidad de perfiles buscados que programen un lenguaje en concreto en su página de demanda de empleo “*IEEE Job Site*”.



- Trending. Basado mayoritariamente en la tendencia en foros y redes sociales.



En este sondeo la máxima variación con respecto a TIOBE se encuentra en el sondeo “Jobs”, en el cual SQL es el lenguaje más buscado a la hora de la demanda de empleo, al igual que también existe una pequeña variación en el sondeo “Spectrum” en el cual Java se encuentra en quinta posición.

El que más se puede asemejar a TIOBE es el sondeo “Trending”, ya que usa una recopilación de datos más parecida.

Fuente: <https://spectrum.ieee.org/top-programming-languages-2022>

## Ejercicio A3

Define variables de al menos 5 tipos diferentes y dale un valor adecuado.

```

4- class Main {
5-     public static void main(String[] args) {
6
7         //inicialización de variables
8         byte edad;
9         char grupo;
10        double nota;
11        boolean apto;
12        String estado;
13
14        /* asignación de valores a las variables*/
15        edad = 25;
16        grupo = 'c';
17        nota = 8.35;
18        apto = true;
19        estado = "aprobado";
20
21        //comprobación de salida por consola
22        System.out.println("edad del alumno "+edad);
23        System.out.println("grupo "+grupo);
24        System.out.println("calificación "+nota);
25        System.out.println("apto "+apto);
26        System.out.println("resumen "+estado);
27    }
28 }

```

Java (OpenJDK 13.0.1)

edad del alumno 25  
grupo c  
calificación 8.35  
apto true  
resumen aprobado

## Ejercicio A4

Define tres variables, dotando a cada una de ellas los siguientes valores: Nombre, primer apellido y segundo apellido. Muestra por pantalla el valor de cada una de estas variables de manera separada en una línea (serían tres líneas por terminal) y luego en una cuarta línea del terminal muestra todas estas variables de manera conjunto mostrándose tal que así “Nombre de alumno: Guillermo Palazón Cano

```

1 // Online Java Compiler
2 // Run Java code without any setup
3
4- class Main {
5-     public static void main(String[] args) {
6
7         //inicialización y asignación de valores a las variables
8         String nombre = "Guillermo";
9         String primerApellido = "Palazón";
10        String segundoApellido = "Cano";
11
12        //muestra por pantalla en renglones separados
13        System.out.println(nombre);
14        System.out.println(primerApellido);
15        System.out.println(segundoApellido);
16
17        //muestra por pantalla en el mismo renglón
18        System.out.println("Nombre de alumno: "+nombre+" "+primerApellido+" "+segundoApellido);
19    }
20 }

```

Java (OpenJDK 13.0.1) Run

Guillermo  
Palazón  
Cano  
Nombre de alumno: Guillermo Palazón Cano

## Ejercicio A5

Realiza lo mismo que el ejercicio anterior, pero en este caso las dos variables de los apellidos están fijas (como en el ejercicio anterior), pero el nombre se deberá meter por consola.

```

1 import java.util.Scanner;
2
3 class Main {
4     public static void main(String[] args) {
5         //inicialización variable de tipo Scanner
6         Scanner nombreScanner = new Scanner(System.in);
7         //inicialización de la variable introducida por teclado
8         String nombre;
9         //inicialización y asignación de valores a las variables fijas
10        String primerApellido = "Palazón";
11        String segundoApellido = "Cano";
12        // introducción por teclado de la variable Scanner
13        nombre = nombreScanner.nextLine();
14        //muestra por pantalla en renglones separados
15        System.out.println(nombre);
16        System.out.println(primerApellido);
17        System.out.println(segundoApellido);
18        //muestra por pantalla en el mismo renglón
19        System.out.println("Nombre de alumno: "+nombre+" "+primerApellido+" "+
20                           +segundoApellido);
21    }
22 }

```

Java (OpenJDK 13.0.1) Run

Guillermo  
Palazón  
Cano  
Nombre de alumno: Guillermo Palazón Cano

Guillermo

## Ejercicio A6

Realiza un programa que pida al usuario un número de tipo real y muestre el resultado de multiplicar el mismo por un número que será el correspondiente al mes de tu fecha de nacimiento (por ejemplo, si has nacido en agosto, deberás multiplicar dicho número real por el 8).

```

1 import java.util.Scanner;
2
3
4 public class MyClass {
5     public static void main(String args[]) {
6
7         //inicialización variable tipo Scanner y variable para guardarla
8         Scanner numTeclado = new Scanner(System.in);
9         double teclado;
10
11        //inicialización demás variables
12        double mesNacimiento = 4;
13        double resultado;
14
15        //introducción por teclado
16        teclado = numTeclado.nextDouble();
17
18        //multiplicación
19        resultado = mesNacimiento * teclado;
20
21        System.out.println("El resultado es: "+resultado);
22    }
23 }
24

```

Execute Mode, Version, Inputs & Arguments

JDK 17.0.1 Interactive Stdin Inputs

CommandLine Arguments

2.6

Execute

Result

CPU Time: 0.30 sec(s), Memory: 37724 kilobyte(s)

El resultado es: 10.4

## Ejercicio A7

Realiza un programa que pida un número y muestre por pantalla el número siguiente haciendo uso del operador de incremento.

```
1  import java.util.Scanner;
2
3  public class MyClass {
4      public static void main(String args[]) {
5
6          //inicialización de la variables
7          int resultado;
8          int numTeclado;
9
10         //creación de la variable scanner e introducción
11         Scanner numScan = new Scanner(System.in);
12         numTeclado = numScan.nextInt();
13
14         //incremento
15         resultado = ++numTeclado;
16
17         System.out.println("El resultado es: "+resultado);
18
19     }
20 }
21 }
```

Execute Mode, Version, Inputs & Arguments

JDK 17.0.1

CommandLine Arguments

Stdin Inputs

10

Execute

Result

CPU Time: 0.25 sec(s), Memory: 37020 kilobyte(s)

El resultado es: 11



## Ejercicio A8

Suponiendo que ya has cumplido años en el año actual. Realiza un programa que pida tu año de nacimiento y calcule el número de años que tienes en la actualidad.

```
1 import java.util.Scanner;
2
3 public class MyClass {
4     public static void main(String args[]) {
5
6         //declaración de las variables
7         int añoNacimiento;
8         int añoActual = 2022;
9         int edad;
10
11         //creación de la clase Scanner
12         Scanner añoTeclado = new Scanner(System.in);
13
14         //petición del año de nacimiento por pantalla
15         añoNacimiento = añoTeclado.nextInt();
16
17         //cálculo de la edad
18         edad = añoActual - añoNacimiento;
19
20         //salida por consola
21         System.out.println("Tu edad es: "+edad);
22
23     }
24 }
25 }
```

Execute Mode, Version, Inputs & Arguments

JDK 17.0.1 ☐ Interactive Stdin Inputs

CommandLine Arguments

1993

**Execute**

Result

CPU Time: 0.23 sec(s), Memory: 37272 kilobyte(s)

Tu edad es: 29

## Ejercicio A9

Realiza un programa que muestre la media de tres números que se hayan introducido. El resultado que se muestre debe obtenerse con decimales.

```
1 import java.util.Scanner;
2
3 public class MyClass {
4     public static void main(String args[]) {
5
6         //creación de la clase Scanner
7         Scanner numTeclado = new Scanner(System.in);
8
9         //inicialización de las variables
10        float num1;
11        float num2;
12        float num3;
13        float media;
14
15        //lectura por pantalla
16        num1 = numTeclado.nextFloat();
17        num2 = numTeclado.nextFloat();
18        num3 = numTeclado.nextFloat();
19
20        //cálculo de la media
21        media = (num1 + num2 + num3) / 3;
22
23        //salida por pantalla
24        System.out.println("La media es: "+media);
25    }
26 }
```

Execute Mode, Version, Inputs & Arguments

JDK 17.0.1  ☐ Interactive Stdin Inputs

CommandLine Arguments

2.2  
3  
4.5

result

PU Time: 0.27 sec(s), Memory: 38944 kilobyte(s)

```
La media es: 3.2333333
```

## Ejercicio A10

Realiza un programa que pida un número y muestre por consola TRUE o FALSE, en función de si ese número es IMPAR (con un 3 indicará TRUE y un 4 será FALSE).

```
1 import java.util.Scanner;
2
3 public class MyClass {
4     public static void main(String args[]) {
5
6         //creación de variables
7         int num;
8         int modulo;
9         boolean impar;
10
11         //introducción del número por pantalla
12         Scanner numPantalla = new Scanner(System.in);
13         num = numPantalla.nextInt();
14
15         //cálculo de par o impar
16         modulo = num % 2;
17         impar = modulo != 0 ? true : false;
18
19         //salida por pantalla
20         System.out.println("El número introducido es impar: "+impar);
21
22     }
23 }
```

Execute Mode, Version, Inputs & Arguments

JDK 17.0.1  ☐ Interactive Stdin Inputs

CommandLine Arguments

13

Result

CPU Time: 0.27 sec(s), Memory: 37220 kilobyte(s)

```
El número introducido es impar: true
```

## Ejercicio A11

Realiza un programa que a partir de dos números que ha introducido el usuario, indique TRUE si estos son iguales y FALSE en caso contrario. NO debes utilizar el operador ternario.

```
1 import java.util.Scanner;
2
3 public class MyClass {
4     public static void main(String args[]) {
5
6         //creación de variables
7         int num1;
8         int num2;
9         boolean comparacion;
10
11         //introducción por pantalla de los números
12         Scanner numPantalla = new Scanner(System.in);
13         num1 = numPantalla.nextInt();
14         num2 = numPantalla.nextInt();
15
16         //cálculo de la igualdad
17         comparacion = num1 == num2;
18
19         //salida por pantalla
20         System.out.println("Los números introducidos son iguales: "+comparacion);
21     }
22 }
```

Execute Mode, Version, Inputs & Arguments

JDK 17.0.1

Interactive

Stdin Inputs

8

8

CommandLine Arguments

Execute

Result

CPU Time: 0.28 sec(s), Memory: 37068 kilobyte(s)

Los números introducidos son iguales: true

## Ejercicio A12

Realiza un programa que a partir de dos números que ha introducido el usuario, indique TRUE si el primero es mayor que el segundo y la suma de ambos sea mayor o igual a 23. En cualquier otro caso debe devolver FALSE.

```
1 import java.util.Scanner;
2
3 public class MyClass {
4     public static void main(String args[]) {
5
6         //creación de las variables
7         int num1;
8         int num2;
9         boolean comparacion;
10
11         //introducción de los números por pantalla
12         Scanner numPantalla = new Scanner(System.in);
13         num1 = numPantalla.nextInt();
14         num2 = numPantalla.nextInt();
15
16         //comparación de los dos números
17         comparacion = (num1 > num2) && ((num1 + num2) >= 23);
18
19         //salida por pantalla
20         System.out.println("Se cumple la condición: "+comparacion);
21     }
22 }
```

Execute Mode, Version, Inputs & Arguments

JDK 17.0.1



Interactive

Stdin Inputs

CommandLine Arguments

13

10



Execute



Result

CPU Time: 0.25 sec(s), Memory: 37176 kilobyte(s)

Se cumple la condición: true

## Ejercicio A13

Realiza un programa que a partir de dos números que ha introducido el usuario, devuelva TRUE si ambos números son mayores a 25.

```
1 import java.util.Scanner;
2
3 public class MyClass {
4     public static void main(String args[]) {
5
6         //creación de las variables
7         int num1;
8         int num2;
9         boolean comparacion;
10
11         //introducción de los números por pantalla
12         Scanner numPantalla = new Scanner(System.in);
13         num1 = numPantalla.nextInt();
14         num2 = numPantalla.nextInt();
15
16         //comparación de los dos números
17         comparacion = (num1 > 25) && (num2 > 25);
18
19         //salida por pantalla
20         System.out.println("Se cumple la condición: "+comparacion);
21     }
22 }
```

Execute Mode, Version, Inputs & Arguments

JDK 17.0.1  ☐ Interactive Stdin Inputs

CommandLine Arguments

27  
26

Result

CPU Time: 0.23 sec(s), Memory: 37320 kilobyte(s)

```
Se cumple la condición: true
```

## Ejercicio A14

Utilizando el operador ternario resuelve la actividad A11. A13.

A11 con operador ternario

```
1 import java.util.Scanner;
2
3 public class MyClass {
4     public static void main(String args[]) {
5
6         //creación de las variables
7         int num1;
8         int num2;
9         boolean iguales;
10
11         //introducción de los números por pantalla
12         Scanner numPantalla = new Scanner(System.in);
13         num1 = numPantalla.nextInt();
14         num2 = numPantalla.nextInt();
15
16         //comparación de los dos números
17         iguales = num1 == num2 ? true : false;
18
19         //salida por pantalla
20         System.out.println("Los números introducidos son iguales: "+iguales);
21     }
22 }
```

Execute Mode, Version, Inputs & Arguments

JDK 17.0.1  ☐ Interactive Stdin Inputs

CommandLine Arguments

2  
2

Result

CPU Time: 0.26 sec(s), Memory: 37196 kilobyte(s)

Los números introducidos son iguales: true

## A13 con operador ternario

```
1 import java.util.Scanner;
2
3 public class MyClass {
4     public static void main(String args[]) {
5
6         //creación de las variables
7         int num1;
8         int num2;
9         boolean mayoresDe25;
10
11         //introducción de los números por pantalla
12         Scanner numPantalla = new Scanner(System.in);
13         num1 = numPantalla.nextInt();
14         num2 = numPantalla.nextInt();
15
16         //comparación de los dos números
17         mayoresDe25 = (num1 > 25) && (num2 > 25) ? true : false;
18
19         //salida por pantalla
20         System.out.println("Los números introducidos son mayores de 25: "+mayoresDe25);
21     }
22 }
```

## Execute Mode, Version, Inputs &amp; Arguments

JDK 17.0.1



Interactive

Stdin Inputs

CommandLine Arguments

28  
26

Execute



## Result

CPU Time: 0.25 sec(s), Memory: 37464 kilobyte(s)

Los números introducidos son mayores de 25: true



## Ejercicio A15

Utilizando el operador “opera y asigna” realiza un programa que pida el valor de un producto sin IVA y lo guarde en una variable. Sobre dicha variable debe calcular el total del producto (en la misma variable) aplicando un IVA que será el valor del mes en el que has nacido.

```
1 import java.util.Scanner;
2
3 public class MyClass {
4     public static void main(String args[]) {
5
6         //creación de las variables
7         double precioSinIVA;
8
9         //introducción del precio por pantalla
10        Scanner precioPantalla = new Scanner(System.in);
11        precioSinIVA = precioPantalla.nextDouble();
12
13        //cálculo del precio con IVA
14        precioSinIVA *= 1.04;
15
16
17        //salida por pantalla
18        System.out.println("El precio con IVA es: "+precioSinIVA);
19    }
20 }
```

Execute Mode, Version, Inputs & Arguments

JDK 17.0.1  ☐ Interactive Stdin Inputs

CommandLine Arguments

20.50

Result

PU Time: 0.27 sec(s), Memory: 38092 kilobyte(s)

El precio con IVA es: 21.32

## Ejercicio A16

Realiza un programa que tome como entrada un número entero e indique qué cantidad hay que sumarle para que sea múltiplo de 9 (por ejemplo, si se introduce un 2 habrá que sumarle 7 y si se introduce un 17 habrá que sumarle 1 para que el resultado sea múltiplo de 9).

```
1 import java.util.Scanner;
2
3 public class MyClass {
4     public static void main(String args[]) {
5
6         //creación de las variables
7         int num;
8         int cantidadSumar;
9
10        //introducción del número por pantalla
11        Scanner numPantalla = new Scanner(System.in);
12        num = numPantalla.nextInt();
13
14        //cálculo de la cantidad a sumar
15        cantidadSumar = (num % 9) == 0 ? 0 : 9 - (num % 9);
16
17        //salida por pantalla
18        System.out.println("Tenemos que sumar "+cantidadSumar+" para que sea múltiplo de 9");
19    }
20 }
```

Execute Mode, Version, Inputs & Arguments

JDK 17.0.1  ☐ Interactive Stdin

CommandLine Arguments

17

Result

CPU Time: 0.25 sec(s), Memory: 37588 kilobyte(s)

Tenemos que sumar 1 para que sea múltiplo de 9

## Ejercicio A17

Realiza un programa que a partir de dos números que serán la base y la altura de un triángulo, nos calcule su área:  $\text{Área} = (\text{base} * \text{altura}) / 2$ .

```
1 import java.util.Scanner;
2
3 public class MyClass {
4     public static void main(String args[]) {
5
6         //creación de las variables
7         int base;
8         int altura;
9         int area;
10
11         //introducción del número por pantalla
12         Scanner numPantalla = new Scanner(System.in);
13         base = numPantalla.nextInt();
14         altura = numPantalla.nextInt();
15
16         //cálculo del área
17         area = (base * altura)/2;
18
19         //salida por pantalla
20         System.out.println("El área del triángulo es: "+area);
21     }
22 }
```

Execute Mode, Version, Inputs & Arguments

JDK 17.0.1  ☐ Interactive Stdin Inputs

CommandLine Arguments

4  
2

Result

CPU Time: 0.24 sec(s), Memory: 37308 kilobyte(s)

El área del triángulo es: 4

## Ejercicio A18

Realiza un programa que a partir de un número introducido que será el número de segundos, se obtenga de resultado el número de horas, minutos y segundos que equivalen a ese número de segundos.

```
1 import java.util.Scanner;
2
3 public class MyClass {
4     public static void main(String args[]) {
5
6         //creación de las variables
7         int segundosPantalla;
8         int segundos;
9         int horas;
10        int minutos;
11        int moduloHoras;
12
13        //introducción del número por pantalla
14        Scanner numPantalla = new Scanner(System.in);
15        segundosPantalla = numPantalla.nextInt();
16
17        //conversión de los segundos a horas, minutos y segundos
18        horas = segundosPantalla / 3600;
19        moduloHoras = segundosPantalla % 3600;
20        minutos = moduloHoras / 60;
21        segundos = segundosPantalla % 60;
22
23        //salida por pantalla
24        System.out.println(segundosPantalla+" segundos "+
25                           "equivalen a "+horas+" horas, "+minutos+" minutos y "+segundos+" segundos");
26    }
```

Execute Mode, Version, Inputs & Arguments

JDK 17.0.1



Interactive

Stdin Inputs

CommandLine Arguments

3670

Execute



Result

CPU Time: 0.25 sec(s). Memory: 38868 kilobyte(s)

3670 segundos equivalen a 1 horas, 1 minutos y 10 segundos

## Ejercicio A19

Un parque de actividades nos contrata una aplicación para calcular el importe que se le tiene que cobrar a las familias que acudan a sacar sus entradas. El precio de las entradas infantiles es de 12,25 € y el de las entradas de adultos es 17,50€. Si el total de la compra es superior a 80 € se le aplica automáticamente un descuento del 7%. A partir del número de entradas infantiles y de adultos, calcula el precio que debe pagar dicha familia.

```

1 import java.util.Scanner;
2
3 public class MyClass {
4     public static void main(String args[]) {
5
6         //creación de las variables
7         double entradaInfantil = 12.25;
8         double entradaAdulto = 17.50;
9         int numEntradasInfantil;
10        int numEntradasAdulto;
11        double precioFinalTotal;
12        double precioFinalDescuento;
13
14        //introducción de la cantidad de entradas por tipo por pantalla
15        Scanner tipoEntrada = new Scanner(System.in);
16        numEntradasInfantil = tipoEntrada.nextInt();
17        numEntradasAdulto = tipoEntrada.nextInt();
18
19        //cálculo del pago a realizar
20        precioFinalTotal = (numEntradasInfantil * entradaInfantil) + (numEntradasAdulto * entradaAdulto);
21        precioFinalDescuento = (precioFinalTotal > 80) ? (precioFinalTotal - (precioFinalTotal * 0.07)) : precioFinalTotal;
22
23        //salida por pantalla
24        System.out.println("El precio de las entradas es: "+precioFinalDescuento+" €");
25    }
26 }




```

Execute Mode, Version, Inputs & Arguments

JDK 17.0.1 ☐ Interactive Stdin Inputs

CommandLine Arguments

4  
4

**Execute**   

Result

CPU Time: 0.24 sec(s), Memory: 38760 kilobyte(s)

**El precio de las entradas es: 110.67 €**

## Ejercicio A20

Solicita al usuario un número decimal y devuelve un mensaje con dicho número redondeado a su entero más próximo (muestra un número entero).

```
1 import java.util.Scanner;
2
3 public class MyClass {
4     public static void main(String args[]) {
5
6         //creación de las variables
7         double numEntrada;
8         int entero;
9         int redondeo;
10
11         //introducción del número por pantalla
12         Scanner numero = new Scanner(System.in);
13         numEntrada = numero.nextDouble();
14
15         //cálculo del numero redondeado
16         entero = (int)numEntrada;
17         redondeo = (numEntrada - entero) > 0.5 ? ++entero : entero;
18
19         //salida por pantalla
20         System.out.println("El número redondeado es: "+redondeo);
21     }
22 }
```

Execute Mode, Version, Inputs & Arguments

JDK 17.0.1  ☐ Interactive Stdin Inputs

CommandLine Arguments

2.6

Result

CPU Time: 0.27 sec(s), Memory: 38004 kilobyte(s)

```
El número redondeado es: 3
```