



Universidad Autónoma de Tamaulipas

Facultad de Ingeniería Tampico

Ingeniería en Sistemas Computacionales

Asignatura: Fundamentos De programación

Grupo: N Grado: 1

Nombre del Docente: Álvarez Navarro Eduardo

Alumno: Guevara Martinez Angel Jeremy

Matricula: 2243330342



1. El Problema de las Sandías Descripción: Dos amigos compran una sandía de W kilos. Quieren dividirla de tal forma que cada uno reciba una cantidad de kilos que sea un número par (no necesariamente iguales). • Entrada: Un número entero W . • Lógica: Verificar si el peso es par y si es mayor a 2 (ya que 2 solo se divide en 1 y 1, que son impares). • Salida: "SI" o "NO".

Variables y formulas : W

```
1 Algoritmo tarea_8_01
2     Definir w Como Entero
3
4     Escribir "Ingrese el peso de la sandia:"
5     Leer w
6
7     Si (w MOD 2 = 0) Y (w > 2) Entonces
8         Escribir "SI"
9     SiNo
10        Escribir "NO"
11    FinSi
12 FinAlgoritmo
13
```

*** Ejecución Iniciada. ***

Ingrese el peso de la sandia:

> 4

SI

*** Ejecución Finalizada. ***



```
1 import java.io.*;
2
3 public class tarea_8_01{
4     public static void main(String args[]) throws IOException{
5
6         BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
7         int w;
8
9         w = Integer.parseInt(br.readLine());
10
11        if(w>2 && w%2==0)
12            System.out.println("SI");
13        else
14            System.out.println("NO");
15    }
16}
17
```

Problems Javadoc Declaration Console Chat Eclipse IDE for Java Developers 2026-03 M2
terminated> tarea_8_01 [Java Application] C:\Program Files\Java\jdk-24\bin\javaw.exe (15 feb 2026 14:53:36 – 14:53:51 elapsed 0:00:15) [pid: 18348]

I

2. Clasificación de Triángulos por Ángulos Descripción: Dados tres ángulos internos A, B y C. • Lógica: 1. Validar si la suma es 180 y todos son mayores a 0. 2. Si alguno es 90: "Rectángulo". 3. Si alguno es mayor a 90: "Obtusángulo". 4. Si todos son menores a 90: "Acutángulo". • Salida: El tipo de triángulo o "Ángulos Inválidos".



```
1 Algoritmo tarea_8_02
2     definir a,b,c,may,x,n Como Real
3
4     escribir "Ingrese los lados del triangulo"
5     leer a,b,c
6
7     Si (a+b>c) Y (a+c>b) Y (b+c>a) Entonces
8
9         Si a>b Y a>c Entonces
10            may<=a
11            x<=b
12            n<=c
13        SiNo
14            Si b>a Y b>c Entonces
15                may<=b
16                x<=a
17                n<=c
18            SiNo
19                may<=c
20                x<=a
21                n<=b
22        FinSi
23    FinSi
24
25    Si may2 = x2 + n2 Entonces
26        Escribir "Rectangulo"
27    SiNo
28        Si may2 > x2 + n2 Entonces
29            Escribir "Obtusangulo"
30        SiNo
31            Escribir "Acutangulo"
32        FinSi
33    FinSi
34
35 SiNo
36     Escribir "No forman un triangulo"
37 FinSi
38 FinAlgoritmo
```

*** Ejecución Iniciada. ***

Ingrese los lados del triangulo

> 89

> 56

> 99

Acutangulo

*** Ejecución Finalizada. ***



The screenshot shows the Eclipse IDE interface with the following details:

- Java Editor:** Displays the code for "tarea_8_02.java". The code reads three double values from standard input, calculates their sum, and prints the type of triangle based on the sum and individual angles (a, b, c). The code handles invalid inputs where the sum is not 180 or individual angles are not between 0 and 90 degrees.
- Console Output:** Shows the terminal window with the following text:

```
34
90
12
Angulos invalidos
```
- Bottom Status Bar:** Shows the status bar with various icons and the text: "Problems Javadoc Declaration Console Chat Eclipse IDE for Java Developers 2026-03 M <terminated> tarea_8_02 [Java Application] C:\Program Files\Java\jdk-24\bin\javaw.exe (15 feb 2026 15:22:56 – 15:23:07 elapsed 0:00:11) [pid: 17600]

3. El Salto del Caballo (Ajedrez) Descripción: Dadas las coordenadas iniciales (x_1, y_1) y finales (x_2, y_2) en un tablero de 8x8. • Lógica: Un caballo mueve en "L". Calcular la diferencia absoluta en X (dx) y en Y (dy). El movimiento es válido si ($dx==2$ y $dy==1$) o ($dx==1$ y $dy==2$). • Salida: "Movimiento Válido" o "Inválido".



```
1 Algoritmo tarea_8_03
2     Definir x1,y1,x2,y2,dx,dy Como Entero
3
4     Escribir "Ingrese x1:"
5     Leer x1
6     Escribir "Ingrese y1:"
7     Leer y1
8     Escribir "Ingrese x2:"
9     Leer x2
10    Escribir "Ingrese y2:"
11    Leer y2
12
13    dx ← abs(x2-x1)
14    dy ← abs(y2-y1)
15
16    Si (dx=2 Y dy=1) O (dx=1 Y dy=2) Entonces
17        Escribir "Movimiento Valido"
18    SiNo
19        Escribir "Movimiento Invalido"
20    FinSi
21 FinAlgoritmo
22
```

```
► PSeInt - Ejecutando proceso TAREA_8_03
*** Ejecución Iniciada. ***
Ingrese x1:
> 3
Ingrese y1:
> 2
Ingrese x2:
> 1
Ingrese y2:
> 3
Movimiento Valido
*** Ejecución Finalizada. ***
```



VERDAD, BELLEZA, PROBIDAD



Universidad
Autónoma de
TAMAULIPAS



```
1 import java.io.*;
2
3 public class tarea_8_03{
4     public static void main(String args[]) throws IOException{
5         BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
6         int x1,y1,x2,y2,dx,dy;
7
8         x1=Integer.parseInt(br.readLine());
9         y1=Integer.parseInt(br.readLine());
10        x2=Integer.parseInt(br.readLine());
11        y2=Integer.parseInt(br.readLine());
12
13        dx=Math.abs(x1-x2);
14        dy=Math.abs(y1-y2);
15
16        if((dx==2&dy==1)|| (dx==1&dy==2))
17            System.out.println("Movimiento valido");
18        else
19            System.out.println("Invalido");
20    }
21 }
22 }
```

Eclipse IDE interface showing the code in the editor and the terminal window below it.

4. División de Reclutas Descripción: Clasificar a un soldado según su edad, fuerza (0-100) y visión (0.0-1.0). • Lógica:
1. Si tiene entre 18 y 25 años: Fuerza > 80 y Visión >= 0.8 es "Fuerzas Especiales", de lo contrario si Fuerza > 50 es "Infantería". 2. Si tiene más de 25 años: Visión >= 0.9 es "Estratega". • Salida: Nombre de la división o "No apto".

```
1 Algoritmo tarea_8_04
2     Definir edad,fuerza Como Entero
3     Definir vision Como Real
4
5     Escribir "Ingrese edad:"
6     Leer edad
7     Escribir "Ingrese fuerza (0-100):"
8     Leer fuerza
9     Escribir "Ingrese vision (0-1):"
10    Leer vision
11
12    Si edad≥18 Y edad≤25 Entonces
13        Si fuerza>80 Y vision≥0.8 Entonces|
14            Escribir "Fuerzas Especiales"
15        SiNo
16            Si fuerza>50 Entonces
17                Escribir "Infanteria"
18            SiNo
19                Escribir "No apto"
20            FinSi
21        FinSi
22    SiNo
23        Si edad>25 Y vision≥0.9 Entonces
24            Escribir "Estratega"
25        SiNo
26            Escribir "No apto"
27        FinSi
28    FinSi
29 FinAlgoritmo
```



VERDAD, BELLEZA, PROBIDAD



Universidad
Autónoma de
TAMAULIPAS



```
3  public class tarea_8_04{
4      public static void main(String args[]) throws IOException{
5
6          BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
7          int edad,fuerza;
8          double vision;
9          System.out.println("Ingrese edad, fuerza y vision");
10         edad=Integer.parseInt(br.readLine());
11         fuerza=Integer.parseInt(br.readLine());
12         vision=Double.parseDouble(br.readLine());
13
14         if(edad>18&&edad<=25){
15             if(fuerza>80&&vision>=0.8)
16                 System.out.println("Fuerzas especiales");
17             else
18                 if(fuerza>50)
19                     System.out.println("Infanteria");
20                 else
21                     System.out.println("No apto");
22         }
23         else
24             if(edad>25){
25                 if(vision>0.9)
26                     System.out.println("Estratega");
27                 else
28                     System.out.println("No apto");
29         }
30     }
31 }
32
```

Problems @ Javadoc Declaration Console Chat Eclipse IDE for Java Developers 2026-03 M2

terminated> tarea_8_04 [Java Application] C:\Program Files\Java\jdk-24\bin\javaw.exe (15 feb 2026 15:30:51 – 15:31:06 elapsed 0:00:14) [pid: 11588]

Ingrese edad, fuerza y vision

20

80

0.5

Infanteria

*** Ejecución Iniciada. ***

Ingrese edad:

> 20

Ingrese fuerza (0-100):

> 80

Ingrese vision (0-1):

> 0.5

Infanteria

*** Ejecución Finalizada. ***



5. El Año de Oro Descripción: Determinar si un año Y es de "Oro" bajo reglas específicas. • Lógica: El año debe ser par, NO terminar en 0 (no divisible por 10), el residuo de dividirlo por 4 debe ser 2, y debe estar entre los años 2000 y 3000. • Salida: "Año de Oro" o "Año Ordinario".

```
1 Algoritmo tarea_8_05
2   Definir n Como Entero
3
4   Escribir "Ingrese el año:"
5   Leer n
6
7   Si (n MOD 2 = 0) Y (n MOD 10 ≠ 0) Y (n MOD 4 = 2) Y (n≥2000 Y n≤3000) Entonces
8       Escribir "Año de Oro"
9   SiNo
10      Escribir "Año Ordinario"
11  FinSi
12 FinAlgoritmo
13
```

► PSeInt - Ejecutando proceso TAREA_8_05

*** Ejecución Iniciada. ***

Ingresar el año:

> 2015

Año Ordinario

*** Ejecución Finalizada. ***



The screenshot shows the Eclipse IDE interface. The code editor displays a Java program named `tarea_8_05`. The console tab shows the execution of the program, where the user inputs the year `2000`, and the program outputs `Año Ordinario`.

```
1 import java.io.*;
2
3 public class tarea_8_05{
4     public static void main(String args[]) throws IOException{
5
6         BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
7         int y;
8         System.out.println("Ingrese un año");
9         y=Integer.parseInt(br.readLine());
10
11         if(y%2==0 && y%10!=0 && y%4==2 && y>=2000 && y<=3000)
12             System.out.println("Año de Oro");
13         else
14             System.out.println("Año Ordinario");
15     }
16 }
17
```

Problems Declaration Console Chat Eclipse IDE for Java Developers 2026-03 M2
<terminated> tarea_8_05 [Java Application] C:\Program Files\Java\jdk-24\bin\javaw.exe (15 feb 2026 15:34:07 – 15:34:12 elapsed 0:00:05) [pid: 28588]
Ingrese un año
2000
Año Ordinario

6. Punto en el Rectángulo Descripción: Dadas las esquinas de un rectángulo (x_1, y_1) y (x_2, y_2) , y un punto (px, py) . • Lógica: Verificar si px está entre x_1 y x_2 Y si py está entre y_1 y y_2 . Si los valores son iguales a los límites, está en el "Borde". • Salida: "Dentro", "Borde" o "Fuera".

```
*** Ejecución Iniciada. ***
> 4
> 2
> 3
> 1
> 2
> 1
Fuera
*** Ejecución Finalizada. ***
```



```
1 Algoritmo tarea_8_06
2     Definir x1,y1,x2,y2,px,py Como Real
3
4     Leer x1,y1,x2,y2
5     Leer px,py
6
7     Si (px≥x1 Y px≤x2) Y (py≥y1 Y py≤y2) Entonces
8         Si (px=x1 O px=x2 O py=y1 O py=y2) Entonces
9             Escribir "Borde"
10        SiNo
11            Escribir "Dentro"
12        FinSi
13    SiNo
14        Escribir "Fuera"
15    FinSi
16 FinAlgoritmo
17
```

```
1 import java.io.*;
2
3 public class tarea_8_06{
4     public static void main(String args[]) throws IOException{
5
6         BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
7         int x1,y1,x2,y2,px,py;
8         System.out.println("Ingrese las coordenadas ");
9         x1=Integer.parseInt(br.readLine());
10        y1=Integer.parseInt(br.readLine());
11        x2=Integer.parseInt(br.readLine());
12        y2=Integer.parseInt(br.readLine());
13        px=Integer.parseInt(br.readLine());
14        py=Integer.parseInt(br.readLine());
15
16     if(px>=x1&&px<=x2&&py>=y1&&py<=y2){
17         if(px==x1||px==x2||py==y1||py==y2)
18             System.out.println("Borde");
19         else
20             System.out.println("Dentro");
21     }
22     else
23         System.out.println("Fuera");
24 }
25 }
```

```
Problems Javadoc Declaration Console X Chat Eclipse IDE for Java Developers 2026-03 M2
terminated> tarea_8_06 [Java Application] C:\Program Files\Java\jdk-24\bin\javaw.exe (15 feb 2026 15:37:17 – 15:37:24 elapsed 0:00:07) [pid: 29380]
2
1
2
3
4
Fuera
```



7. Círculos que Colisionan Descripción: Dados centros (x_1, y_1) , (x_2, y_2) y radios r_1, r_2 . • Lógica: Comparar la distancia entre centros con la suma de los radios. • Fórmula: $(x_2-x_1)^2 + (y_2-y_1)^2$ comparado contra $(r_1+r_2)^2$.
- Salida: "Traslanan", "Tangentes" o "No se tocan"

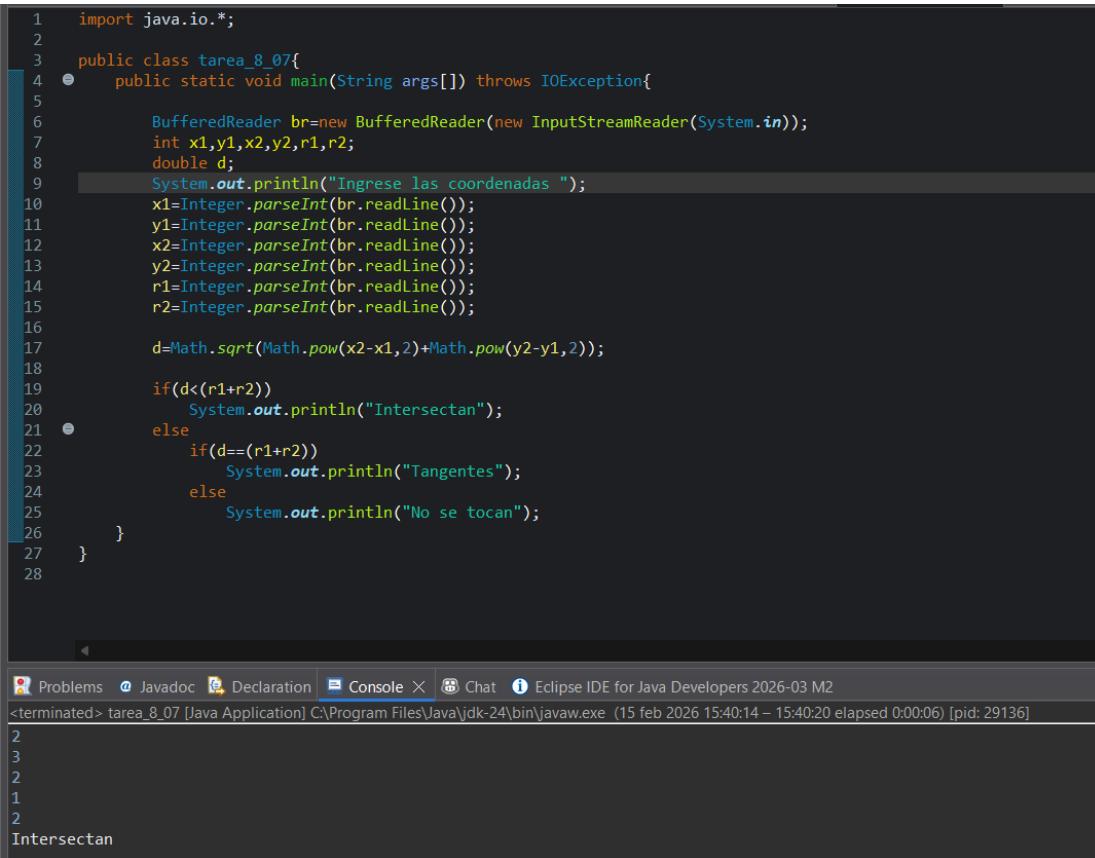
```
1 Algoritmo tarea_8_07
2     Definir x1,y1,x2,y2,r1,r2,d Como Real
3
4     Leer x1,y1
5     Leer x2,y2
6     Leer r1,r2
7
8     d ← (x2-x1)↑2 + (y2-y1)↑2
9
10    Si d < (r1+r2)↑2 Entonces
11        Escribir "Intersecan"
12    SiNo
13        Si d = (r1+r2)↑2 Entonces
14            Escribir "Tangentes"
15        SiNo
16            Escribir "No se tocan"
17        FinSi
18    FinSi
19 FinAlgoritmo
20
```

*** Ejecución Iniciada. ***

```
> 2
> 4
> 1
> 2
> 3
> 1
```

Intersecan

*** Ejecución Finalizada. ***



```
1 import java.io.*;
2
3 public class tarea_8_07{
4     public static void main(String args[]) throws IOException{
5
6         BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
7         int x1,y1,x2,y2,r1,r2;
8         double d;
9         System.out.println("Ingrese las coordenadas ");
10        x1=Integer.parseInt(br.readLine());
11        y1=Integer.parseInt(br.readLine());
12        x2=Integer.parseInt(br.readLine());
13        y2=Integer.parseInt(br.readLine());
14        r1=Integer.parseInt(br.readLine());
15        r2=Integer.parseInt(br.readLine());
16
17        d=Math.sqrt(Math.pow(x2-x1,2)+Math.pow(y2-y1,2));
18
19        if(d<(r1+r2))
20            System.out.println("Intersecan");
21        else
22            if(d==(r1+r2))
23                System.out.println("Tangentes");
24            else
25                System.out.println("No se tocan");
26    }
27 }
28
```

The screenshot shows the Eclipse IDE interface with the Java code for calculating the distance between two circles and determining if they intersect. The code uses BufferedReader to read coordinates and radii from the console. It calculates the distance d and compares it with the sum of the radii (r1+r2) to determine if the circles intersect, are tangent, or do not touch.

Console output:

```
2
3
2
1
2
Intersecan
```

8. Detector de Números Capicúa Descripción: Verificar si un número de 4 cifras se lee igual al revés. • Lógica: Extraer dígitos usando / y %. Comparar si el primer dígito es igual al cuarto y el segundo al tercero. • Salida: "Es Capicúa" o "No lo es".



```
1 Algoritmo tarea_8_08
2     Definir n,d1,d2,d3,d4 Como Entero
3
4     Leer n
5
6     Si n≥1000 Y n≤9999 Entonces
7
8         d1 ← n DIV 1000
9         d2 ← (n DIV 100) MOD 10
10        d3 ← (n DIV 10) MOD 10
11        d4 ← n MOD 10
12
13        Si d1=d4 Y d2=d3 Entonces
14            Escribir "Es capicua"
15        SiNo
16            Escribir "No lo es"
17        FinSi
18
19    SiNo
20        Escribir "El numero debe ser de 4 cifras"
21    FinSi
22
23 FinAlgoritmo
24
```

```
1 import java.io.*;
2
3 public class tarea_8_08{
4     public static void main(String args[]) throws IOException{
5
6         BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
7         int n,d1,d4;
8         System.out.println("Ingrese un numero de 4 digitos");
9         n=Integer.parseInt(br.readLine());
10
11         d1=n/1000;
12         d4=n%10;
13
14         if(d1==d4)
15             System.out.println("Capicua");
16         else
17             System.out.println("No lo es");
18     }
19 }
```

```
Problems Javadoc Declaration Console X Chat Eclipse IDE for Java Developers 2026-03 M2
<terminated> tarea_8_08 [Java Application] C:\Program Files\Java\jdk-24\bin\javaw.exe (15 feb 2026 15:43:41 - 15:43:45 elapsed 0:00:03) [pid: 23508]
Ingrese un numero de 4 digitos
1212
No lo es
```



9. El Dilema de Pitágoras Descripción: Dados los lados A, B y C (donde C es el mayor). • Lógica: Si $A^2 + B^2 == C^2$ es "Rectángulo". Si $A^2 + B^2 > C^2$ es "Acutángulo". Si $A^2 + B^2 < C^2$ es "Obtusángulo". • Salida: Tipo de triángulo.

```
1 Algoritmo tarea_8_09
2     Definir a,b,c Como Real
3
4     Leer a,b,c
5
6     Si a↑2 + b↑2 = c↑2 Entonces
7         Escribir "Rectangulo"
8     SiNo
9         Si a↑2 + b↑2 > c↑2 Entonces
10            Escribir "Acutangulo"
11        SiNo
12            Escribir "Obtusangulo"
13        FinSi
14    FinSi
15 FinAlgoritmo
16
```

*** Ejecución Iniciada. ***

> 3

> 4

> 2

Acutangulo

*** Ejecución Finalizada. ***



```
1 import java.io.*;
2
3 public class tarea_8_09{
4     public static void main(String args[]) throws IOException{
5
6         BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
7         double a,b,c;
8         System.out.println("Ingrese los lados del triangulo");
9         a=Double.parseDouble(br.readLine());
10        b=Double.parseDouble(br.readLine());
11        c=Double.parseDouble(br.readLine());
12
13        if(a*a+b*b==c*c)
14            System.out.println("Rectangulo");
15        else
16            if(a*a+b*b>c*c)
17                System.out.println("Acutangulo");
18            else
19                System.out.println("Obtusangulo");
20    }
21 }
22
```

Screenshot of Eclipse IDE showing Java code for triangle classification and its execution output.

The code reads three double values from the console and prints "Rectangulo" if they satisfy the Pythagorean theorem ($a^2 + b^2 = c^2$). Otherwise, it prints "Acutangulo" if $a^2 + b^2 > c^2$ and "Obtusangulo" if $a^2 + b^2 < c^2$.

Execution output:

```
Problems Javadoc Declaration Console Chat Eclipse IDE for Java Developers 2026-03 M
<terminated> tarea_8_09 [Java Application] C:\Program Files\Java\jdk-24\bin\javaw.exe (15 feb 2026 15:46:31 – 15:46:36 elapsed 0:00:05)
Ingrese los lados del triangulo
2
3
1
Acutangulo
```

10. Validador de Movimiento de Torre Descripción: Posición inicial (f_1, c_1) y final (f_2, c_2). • Lógica: La torre es válida si $f_1 == f_2$ o $c_1 == c_2$, siempre que no sea la misma posición exacta. • Salida: "Válido" o "Inválido".



```
1 Algoritmo tarea_8_10
2     Definir f1,c1,f2,c2 Como Entero
3
4     Leer f1,c1
5     Leer f2,c2
6
7     Si (f1=f2 O c1=c2) Y (f1≠f2 O c1≠c2) Entonces
8         Escribir "Valido"
9     SiNo
10        Escribir "Invalido"
11    FinSi
12 FinAlgoritmo
13
```

*** Ejecución Iniciada. ***

```
> 3
> 2
> 3
> 2
```

Invalido

*** Ejecución Finalizada. ***



```
1 import java.io.*;
2
3 public class tarea_8_10{
4     public static void main(String args[]) throws IOException{
5
6         BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
7         int f1,c1,f2,c2;
8         System.out.println("Ingrese las coordenadas de la primera torre");
9         f1=Integer.parseInt(br.readLine());
10        c1=Integer.parseInt(br.readLine());
11        f2=Integer.parseInt(br.readLine());
12        c2=Integer.parseInt(br.readLine());
13
14        if(f1==f2||c1==c2)
15            System.out.println("Valido");
16        else
17            System.out.println("Invalido");
18    }
19 }
20
```

The screenshot shows the Eclipse IDE interface with the following details:

- Toolbar: Problems, Javadoc, Declaration, Console (highlighted), Chat.
- Console tab: Shows the output of the Java application "tarea_8_10".
- Output:

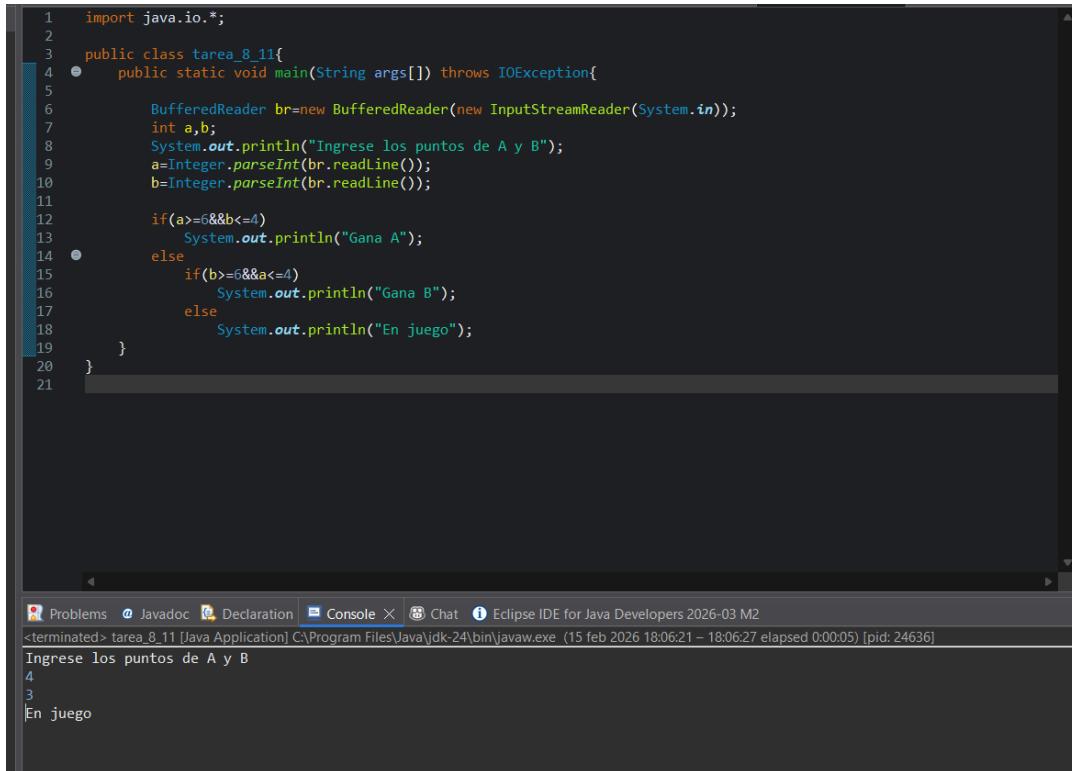
```
<terminated> tarea_8_10 [Java Application] C:\Program Files\Java\jdk-24\bin\javaw.exe (15 feb 2026 15:49:59 – 15:50:05 elapsed 0:00:05) [pid: 11048]
Ingrese las coordenadas de la primera torre
1
2
2
1
Invalido
```

11. Tie-break de Tenis Descripción: Puntos de Jugador A y Jugador B. • Lógica: Si alguien tiene 6 y el otro 4 o menos, gana. Si están 5-5, deben llegar a 7. Si están 6-6, se define por muerte súbita. • Salida: "Gana A", "Gana B" o "En juego".



```
1 Algoritmo tarea_8_11
2     Definir a,b Como Entero
3     Escribir "ingrese datos"
4     Leer a,b
5
6     Si (a≥7 O b≥7) Y abs(a-b)≥2 Entonces
7         Si a>b Entonces
8             Escribir "Gana A"
9         SiNo
10            Escribir "Gana B"
11        FinSi
12    SiNo
13        Escribir "En juego"
14    FinSi
15 FinAlgoritmo
16 |
```

```
*** Ejecución Iniciada. ***
ingrese datos
> 7
> 1
Gana A
*** Ejecución Finalizada. ***
```

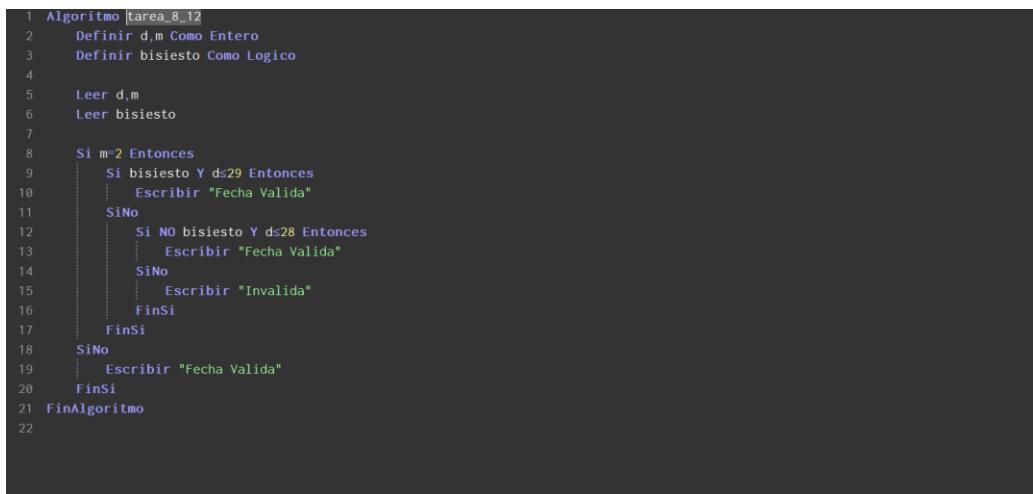


The screenshot shows the Eclipse IDE interface with the following details:

- Code Area:** Displays Java code for a game logic program named "tarea_8_11". The code reads two integers from the user (a and b) and prints the result based on their values.
- Status Bar:** Shows the project name "Problems", JavaDoc, Declaration, Console (with the message "Ingresar los puntos de A y B"), Chat, and the status "Eclipse IDE for Java Developers 2026-03 M2".
- Console Tab:** Shows the terminal output:

```
<terminated> tarea_8_11 [Java Application] C:\Program Files\Java\jdk-24\bin\javaw.exe (15 feb 2026 18:06:21 – 18:06:27 elapsed 0:00:05) [pid: 24636]
Ingresar los puntos de A y B
4
3
En juego
```

12. Validador de Fecha con Febrero Descripción: Dado un día, mes y si el año es bisiesto (booleano). • Lógica: Usar if anidados para meses de 31, 30 y el caso especial de febrero (28 o 29 días). • Salida: "Fecha Válida" o "Inválida".



The screenshot shows pseudocode for a leap year date validator program named "tarea_8_12". The code uses nested if statements to validate the day, month, and leap year status to determine if the date is valid or invalid.

```
1 Algoritmo |tarea_8_12
2   Definir d,m Como Entero
3   Definir bisiesto Como Logico
4
5   Leer d,m
6   Leer bisiesto
7
8   Si m=2 Entonces
9     Si bisiesto Y d≤29 Entonces
10       Escribir "Fecha Valida"
11     SiNo
12       Si NO bisiesto Y d≤28 Entonces
13         Escribir "Fecha Valida"
14       SiNo
15         Escribir "Invalida"
16       FinSi
17     FinSi
18   SiNo
19     Escribir "Fecha Valida"
20   FinSi
21 FinAlgoritmo
22
```



*** Ejecución Iniciada. ***

> 2

> 2

> 0

Fecha Valida

*** Ejecución Finalizada. ***

```
1 import java.io.*;
2
3 public class tarea_8_12{
4     public static void main(String args[]) throws IOException{
5
6         BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
7         int d,m;
8         boolean b;
9         System.out.println("Ingrese el dia, mes y si el año es bisiesto");
10        d=Integer.parseInt(br.readLine());
11        m=Integer.parseInt(br.readLine());
12        b=Boolean.parseBoolean(br.readLine());
13
14        if(m==2){
15            if((b&&d<=29)||(!b&&d<=28))
16                System.out.println("Fecha valida");
17            else
18                System.out.println("Invalida");
19        }
20        else
21            if(d<=31)
22                System.out.println("Fecha valida");
23            else
24                System.out.println("Invalida");
25    }
26 }
27
```

Problems Javadoc Declaration Console Chat Eclipse IDE for Java Developers 2026-03 M2
<terminated> tarea_8_12 [Java Application] C:\Program Files\Java\jdk-24\bin\javaw.exe (15 feb 2026 18:09:41 - 18:09:47 elapsed 0:00:05) [pid: 30024]
Ingrese el dia, mes y si el año es bisiesto
2
2
1
Fecha valida



13. Categoría de Boxeo Descripción: Clasificar por peso en kg. • Lógica: < 52: Mosca, 52-63: Ligero, 64-75: Mediano, 76-91: Pesado, > 91: Súper Pesado. • Salida: Nombre de la categoría.

```
1 Algoritmo tarea_8_13
2   Definir peso Como Real
3
4   Leer peso
5
6   Si pesos≤52 Entonces
7     Escribir "Mosca"
8   SiNo
9     Si peso≤63 Entonces
10    Escribir "Ligero"
11   SiNo
12     Si peso≤75 Entonces
13       Escribir "Mediano"
14     SiNo
15       Si pesos≤91 Entonces
16         Escribir "Pesado"
17       SiNo
18         Escribir "Super Pesado"
19       FinSi
20     FinSi
21   FinSi
22 FinAlgoritmo
23
```

*** Ejecución Iniciada. ***

> 23

Mosca

*** Ejecución Finalizada. ***



```
1 import java.io.*;
2
3 public class tarea_8_13{
4     public static void main(String args[]) throws IOException{
5
6         BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
7         int kg;
8         System.out.println("Ingrese el peso del boxeador");
9         kg=Integer.parseInt(br.readLine());
10
11         if(kg<=52)
12             System.out.println("Mosca");
13         else
14             if(kg<=63)
15                 System.out.println("Ligero");
16             else
17                 if(kg<=75)
18                     System.out.println("Mediano");
19                 else
20                     if(kg<=91)
21                         System.out.println("Pesado");
22                     else
23                         System.out.println("Super pesado");
24     }
25 }
26
```

Problems Javadoc Declaration Console X Chat Eclipse IDE for Java Developers 2026-03 M2
<terminated> tarea_8_13 [Java Application] C:\Program Files\Java\jdk-24\bin\javaw.exe (15 feb 2026 18:13:02 – 18:13:07 elapsed 0:00:04) [pid: 17156]
Ingrese el peso del boxeador
78
Pesado

14. Clima y Humedad Descripción: Entrada de temperatura y humedad. • Lógica: Si Temp > 30 y Humedad > 80: "Sofocante". Si Temp < 15 y Humedad > 80: "Frío Húmedo". • Salida: Clasificación climática.

```
1 Algoritmo tarea_8_14
2     Definir temp,hum Como Real
3
4     Leer temp,hum
5
6     Si temp>30 Y hum>80 Entonces
7         Escribir "Sofocante"
8     SiNo
9         Si temp<15 Y hum>80 Entonces
10            Escribir "Frio Humedo"
11        SiNo
12            Escribir "Clima Normal"
13        FinSi
14    FinSi
15 FinAlgoritmo
16 |
```



*** Ejecución Iniciada. ***

ingrese temperatura

> 45

> 12

Clima Normal

*** Ejecución Finalizada. ***

```
1 import java.io.*;
2
3 public class tarea_8_14{
4     public static void main(String args[]) throws IOException{
5
6         BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
7         int t,h;
8         System.out.println("Ingrese la temperatura y la humedad");
9         t=Integer.parseInt(br.readLine());
10        h=Integer.parseInt(br.readLine());
11
12        if(t>30&&h>80)
13            System.out.println("Sofocante");
14        else
15            if(t<15&&h>80)
16                System.out.println("Frio Humedo");
17    }
18
19 }
```

15. Sistema de Medallas con Honores Descripción: Tres puntajes de exámenes. • Lógica: Promedio > 90 es "Oro". Pero si el promedio es > 90 y alguno de los exámenes es 100, es "Oro con Honores". Si el promedio está entre 80-90 es "Plata". • Salida: Tipo de medalla.



Algoritmo tarea_8_TS

```
Definir e1, e2, e3, prom Como Real
Escribir "Ingrese la calificación del examen 1:"
Leer e1
Escribir "Ingrese la calificación del examen 2:"
Leer e2
Escribir "Ingrese la calificación del examen 3:"
Leer e3
prom ← (e1 + e2 + e3) / 3
Si prom ≥ 90 Entonces
    Si e1 = 100 O e2 = 100 O e3 = 100 Entonces
        Escribir "Oro con Honores"
    SiNo
        Escribir "Oro"
    FinSi
SiNo
    Si prom ≥ 80 Y prom < 90 Entonces
        Escribir "Plata"
    SiNo
        Escribir "Sin medalla"
    FinSi
FinSi
FinAlgoritmo
```

*** Ejecución Iniciada. ***

Ingrese la calificación del examen 1:

> 78

Ingrese la calificación del examen 2:

> 89

Ingrese la calificación del examen 3:

> 78

Plata

*** Ejecución Finalizada. ***



```
1 import java.io.*;
2
3 public class tarea_8_15{
4     public static void main(String args[]) throws IOException{
5
6         BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
7         double e1,e2,e3,p;
8         System.out.println("Ingrese las calificaciones de los tres exámenes");
9         e1=Double.parseDouble(br.readLine());
10        e2=Double.parseDouble(br.readLine());
11        e3=Double.parseDouble(br.readLine());
12
13        p=(e1+e2+e3)/3;
14
15        if(p>90&&(e1==100||e2==100||e3==100))
16            System.out.println("Oro con Honores");
17        else
18            if(p>90)
19                System.out.println("Oro");
20            else
21                if(p>=80)
22                    System.out.println("Plata");
23                else
24                    System.out.println("Sin medalla");
25    }
26 }
27
```

```
Problems @ Javadoc Declaration Console X Chat Eclipse IDE for Java Developers 2026-03 M2
<terminated> tarea_8_15 [Java Application] C:\Program Files\Java\jdk-24\bin\javaw.exe (15 feb 2026 18:21:20 - 18:21:31 elapsed 0:00:10) [pid: 302]
Ingrese las calificaciones de los tres exámenes
78
67
90
Sin medalla
```

- 16 Filtro de Firewall Corporativo (Ciberseguridad) Descripción: Un firewall debe decidir si deja pasar un paquete de datos basado en su IP, Puerto y Protocolo (TCP/UDP). • Lógica: 1. Si la IP de origen es "192.168.1.50", está en la Lista Negra (Bloquear). 2. Si no, si el Puerto es 80 o 443, permitir solo si el Protocolo es "TCP". 3. Si el Puerto es 53, permitir solo si el Protocolo es "UDP" (DNS). 4. Cualquier otro caso debe ser "Bloqueado por defecto". • Salida: "Tráfico Permitido" o "Tráfico Bloqueado".

*** Ejecución Iniciada. ***

Ingrese IP de origen:

> 90

Ingrese puerto:

> 78

Ingrese protocolo (TCP/UDP):

> 9

Tráfico Bloqueado

*** Ejecución Finalizada. ***



```
1 Algoritmo tarea_8_16
2   Definir ip Como Cadena
3   Definir puerto Como Entero
4   Definir protocolo Como Cadena
5   Escribir "Ingrese IP de origen:"
6   Leer ip
7   Escribir "Ingrese puerto:"
8   Leer puerto
9   Escribir "Ingrese protocolo (TCP/UDP):"
10  Leer protocolo
11  Si ip = "192.168.1.50" Entonces
12    Escribir "Tráfico Bloqueado"
13  SiNo
14    Si puerto = 80 O puerto = 443 Entonces
15      Si protocolo = "TCP" Entonces
16        Escribir "Tráfico Permitido"
17      SiNo
18        Escribir "Tráfico Bloqueado"
19    FinSi
20  SiNo
21    Si puerto = 53 Entonces
22      Si protocolo = "UDP" Entonces
23        Escribir "Tráfico Permitido"
24      SiNo
25        Escribir "Tráfico Bloqueado"
26    FinSi
27  SiNo
28    Escribir "Tráfico Bloqueado"
29  FinSi

1 import java.io.*;
2
3 public class tarea_8_16 {
4     public static void main(String args[]) throws IOException {
5
6         BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
7         String ip,protocolo;
8         int puerto;
9
10        System.out.print("Ingrese IP: ");
11        ip = br.readLine();
12
13        System.out.print("Ingrese puerto: ");
14        puerto = Integer.parseInt(br.readLine());
15
16        System.out.print("Ingrese protocolo: ");
17        protocolo = br.readLine();
18
19        if(ip.equals("192.168.1.50"))
20            System.out.println("Tráfico Bloqueado");
21        else
22            if((puerto==80 || puerto==443) && protocolo.equals("TCP"))
23                System.out.println("Tráfico Permitido");
24            else
25                if(puerto==53 && protocolo.equals("UDP"))
26                    System.out.println("Tráfico Permitido");
27                else
28                    System.out.println("Tráfico Bloqueado");
29    }
30 }
```

```
Problems @ Javadoc Declaration Console X Chat Eclipse IDE for Java Developers 2026-03 M2
<terminated> tarea_8_16 [Java Application] C:\Program Files\Java\jdk-24\bin\javaw.exe (15 feb 2026 18:26:03 – 18:26:23 elapsed 0:00:19) [pid: 1856]
Ingrese IP: 99
Ingrese puerto: 80
Ingrese protocolo: 53
Tráfico Bloqueado
```



17. Validador de Fortaleza de Contraseñas (Ciberseguridad) Descripción: Analizar una contraseña basándose en su longitud y presencia de caracteres especiales (puedes simular esto con variables booleanas: tieneMayus, tieneNum, tieneEspecial). • Lógica: 1. Si tiene menos de 8 caracteres: "Insegura". 2. Si tiene 8 o más: - Si tiene los tres tipos de caracteres: "Muy Fuerte". - Si tiene al menos dos: "Fuerte". - Si solo tiene uno: "Débil". • Salida: Nivel de seguridad de la contraseña.

```
1 Algoritmo tarea_8_17
2   Definir long Como Entero
3   Definir tieneMayus, tieneNum, tieneEsp Como Logico
4   Escribir "Ingrese longitud:"
5   Leer long
6   Escribir "Tiene mayúsculas? (1=SI,0=NO)"
7   Leer tieneMayus
8   Escribir "Tiene números? (1=SI,0=NO)"
9   Leer tieneNum
10  Escribir "Tiene caracteres especiales? (1=SI,0=NO)"
11  Leer tieneEsp
12  Si long < 8 Entonces
13    Escribir "Insegura"
14  SiNo
15    Si tieneMayus Y tieneNum Y tieneEsp Entonces
16      Escribir "Muy Fuerte"
17    SiNo
18      Si tieneMayus O tieneNum O tieneEsp Entonces
19        Escribir "Fuerte"
20      SiNo
21        Escribir "Débil"
22      FinSi
23    FinSi
24  FinSi
25
26 FinAlgoritmo
27
```

► PSeInt - Ejecutando proceso TAREA_8_17

*** Ejecución Iniciada. ***

Ingrese longitud:

> 23

Tiene mayúsculas? (1=SI,0=NO)

> 0

Tiene números? (1=SI,0=NO)

> 1

Tiene caracteres especiales? (1=SI,0=NO)

> 1

Fuerte

*** Ejecución Finalizada. ***



```
3  public class tarea_8_17 {
4      public static void main(String args[]) throws IOException {
5
6          BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
7          int longitud, mayus, num, esp;
8
9          System.out.print("Longitud: ");
10         longitud = Integer.parseInt(br.readLine());
11
12         System.out.print("Tiene mayusculas (1/0): ");
13         mayus = Integer.parseInt(br.readLine());
14
15         System.out.print("Tiene numeros (1/0): ");
16         num = Integer.parseInt(br.readLine());
17
18         System.out.print("Tiene especiales (1/0): ");
19         esp = Integer.parseInt(br.readLine());
20
21         if(longitud<8)
22             System.out.println("Insegura");
23         else
24             if(mayus==1 && num==1 && esp==1)
25                 System.out.println("Muy Fuerte");
26             else
27                 if(mayus==1 || num==1 || esp==1)
28                     System.out.println("Fuerte");
29                 else
30                     System.out.println("Debil");
31     }
32 }
33 }
```

```
Problems Javadoc Declaration Console Chat Eclipse IDE for Java Developers 2026-03 M2
<terminated> tarea_8_17 [Java Application] C:\Program Files\Java\jdk-24\bin\javaw.exe (15 feb 2026 18:29:26 – 18:29:45 elapsed 0:00:19) [pid: 2648]
Longitud: 12
Tiene mayusculas (1/0): 1
Tiene numeros (1/0): 1
Tiene especiales (1/0): 0
Fuerte
```

18. Detección de Anomalías en Sensores (Análisis de Datos) Descripción: Un sistema recibe 3 lecturas de temperatura de un servidor. Debe detectar si hay un fallo en los sensores o un sobrecalentamiento. • Lógica: 1. Si la diferencia entre cualquier par de sensores es mayor a 20 grados: "Error de Sensor" (Lecturas inconsistentes). 2. Si no hay error: - Si el promedio de las 3 lecturas es > 80: "Alerta de Incendio". - Si el promedio es > 60: "Alerta de Sobrecaleamiento". - Si no: "Estado Normal". • Salida: Diagnóstico del sistema.



```
1 Algoritmo tarea_8_18
2     Definir t1,t2,t3,prom Como Real
3     Escribir "ingrese datos"
4     Leer t1,t2,t3
5
6     Si Abs(t1-t2)>20 O Abs(t1-t3)>20 O Abs(t2-t3)>20 Entonces
7         Escribir "Error de Sensor"
8     SiNo
9         prom ← (t1+t2+t3)/3
10        Si prom ≥ 80 Entonces
11            Escribir "Alerta de Incendio"
12        SiNo
13            Si prom ≥ 60 Entonces
14                Escribir "Sobrecalentamiento"
15            SiNo
16                Escribir "Estado Normal"
17            FinSi
18        FinSi
19    FinSi
20
21 FinAlgoritmo
22
```

*** Ejecución Iniciada. ***

ingrese datos

> 20

> 60

> 80

Error de Sensor

*** Ejecución Finalizada. ***



```
1 import java.io.*;
2
3 public class tarea_8_18 {
4     public static void main(String args[]) throws IOException {
5
6         BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
7         double t1,t2,t3,prom;
8         System.out.println("Ingrese las temperaturas de los tres sensores");
9         t1 = Double.parseDouble(br.readLine());
10        t2 = Double.parseDouble(br.readLine());
11        t3 = Double.parseDouble(br.readLine());
12
13        if(Math.abs(t1-t2)>20 || Math.abs(t1-t3)>20 || Math.abs(t2-t3)>20)
14            System.out.println("Error de Sensor");
15        else{
16            prom = (t1+t2+t3)/3;
17            if(prom>=80)
18                System.out.println("Alerta de Incendio");
19            else
20                if(prom>=60)
21                    System.out.println("Sobrecalentamiento");
22                else
23                    System.out.println("Estado Normal");
24        }
25    }
26 }
27 }
```

The screenshot shows the Eclipse IDE interface with the following details:

- Toolbar: Problems, Javadoc, Declaration, Console (highlighted), Chat, Eclipse IDE for Java Developers 2026-03 M2.
- Status Bar: <terminated> tarea_8_18 [Java Application] C:\Program Files\Java\jdk-24\bin\javaw.exe (15 feb 2026 18:33:26 – 18:33:37 elapsed 0:00:10) [pid: 30520]
- Console Output:

```
Ingrese las temperaturas de los tres sensores
80
87
90
Alerta de Incendio
```

19. Enrutamiento de Subredes (Redes de Datos) Descripción: Dada una dirección IP (simplificada a un primer octeto) y una máscara de subred, determinar a qué clase de red pertenece y si es una IP privada. • Lógica: 1. Si el octeto está entre 1 y 127: Clase A. - Si el octeto es 10: "Privada". 2. Si está entre 128 y 191: Clase B. - Si el octeto es 172: "Privada". 3. Si está entre 192 y 223: Clase C. - Si el octeto es 192: "Privada". • Salida: "Clase [A/B/C] - [Privada/Pública]".



```
1 Algoritmo tarea_8_19
2   Definir octeto Como Entero
3   Escribir "Ingrese primer octeto:"
4   Leer octeto
5   Si octeto≥1 Y octeto≤127 Entonces
6     Si octeto=10 Entonces
7       Escribir "Clase A - Privada"
8     SiNo
9       Escribir "Clase A - Publica"
10    FinSi
11  SiNo
12    Si octeto≥128 Y octeto≤191 Entonces
13      Si octeto=172 Entonces
14        Escribir "Clase B - Privada"
15      SiNo
16        Escribir "Clase B - Publica"
17      FinSi
18    SiNo
19      Si octeto≥192 Y octeto≤223 Entonces
20        Si octeto=192 Entonces
21          Escribir "Clase C - Privada"
22        SiNo
23          Escribir "Clase C - Publica"
24        FinSi
25      FinSi
26    FinSi
27  FinSi
28 FinAlgoritmo
```

*** Ejecución Iniciada. ***

Ingrese primer octeto:

> 67

Clase A - Publica

*** Ejecución Finalizada. ***



```
1 import java.io.*;
2
3 public class tarea_8_19 {
4     public static void main(String args[]) throws IOException {
5
6         BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
7         int octeto;
8         System.out.print("Ingrese el primer octeto de la dirección IP: ");
9         octeto = Integer.parseInt(br.readLine());
10
11     if(octeto>=1 && octeto<=127){
12         if(octeto==10)
13             System.out.println("Clase A - Privada");
14         else
15             System.out.println("Clase A - Pública");
16     }
17     else
18         if(octeto>=128 && octeto<=191){
19             if(octeto==172)
20                 System.out.println("Clase B - Privada");
21             else
22                 System.out.println("Clase B - Pública");
23         }
24     else
25         if(octeto>=192 && octeto<=223){
26             if(octeto==192)
27                 System.out.println("Clase C - Privada");
28             else
29                 System.out.println("Clase C - Pública");
30         }
31     }
}
```

```
Problems @ Javadoc Declaration Console X Chat Eclipse IDE for Java Developers 2026-03 M2
terminated> tarea_8_19 [Java Application] C:\Program Files\Java\jdk-24\bin\javaw.exe (15 feb 2026 18:36:02 – 18:36:10 elapsed 0:00:08) [pid: 14716]
Ingrese el primer octeto de la dirección IP: 70
Clase A - Pública
```

20. Clasificación de Calidad de Servicio (QoS) en Redes Descripción: Un router prioriza el tráfico según el tipo de aplicación y el retraso (latencia) en milisegundos. • Lógica: 1. Si el tipo de tráfico es "Voz" o "Video": - Si latencia < 50ms: "Prioridad Alta". - Si latencia >= 50ms: "Prioridad Media" (Possible lag). 2. Si el tipo es "Descarga": - Si latencia < 150ms: "Prioridad Baja". - Si no: "Prioridad Mínima". • Salida: Nivel de prioridad asignado al paquete.



```
1 Algoritmo tarea_8_20
2     Definir tipo Como Cadena
3     Definir latencia Como Real
4     Escribir "Tipo de trafico (Voz/Video/Descarga):"
5     Leer tipo
6     Escribir "Latencia en ms:"
7     Leer latencia
8     Si tipo="Voz" O tipo="Video" Entonces
9         Si latencia<50 Entonces
10            Escribir "Prioridad Alta"
11        SiNo
12            Escribir "Prioridad Media"
13        FinSi
14    SiNo
15        Si tipo="Descarga" Entonces
16            Si latencia<150 Entonces
17                Escribir "Prioridad Baja"
18            SiNo
19                Escribir "Prioridad Minima"
20            FinSi
21        FinSi
22    FinSi
23 FinAlgoritmo
```

```
import java.io.*;
public class tarea_8_20 {
    public static void main(String args[]) throws IOException {
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        String tipo;
        double latencia;

        System.out.print("Tipo: ");
        tipo = br.readLine();

        System.out.print("Latencia: ");
        latencia = Double.parseDouble(br.readLine());

        if(tipo.equals("Voz") || tipo.equals("Video")){
            if(latencia<50)
                System.out.println("Prioridad Alta");
            else
                System.out.println("Prioridad Media");
        }
        else
            if(tipo.equals("Descarga")){
                if(latencia<150)
                    System.out.println("Prioridad Baja");
                else
                    System.out.println("Prioridad Minima");
            }
    }
}
```