



---

# **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. Filtro de Paquetes ICMP (Ping de la muerte) Escribe un programa que reciba el tamaño de un paquete (en bytes) y el protocolo (1 para ICMP, 2 para TCP). • Si el protocolo es ICMP: o Si el tamaño es mayor a 65535, imprime: "Ataque Ping de la Muerte detectado". o Si no, imprime: "Paquete ICMP normal". • Si es TCP, imprime: "Ignorar, no es ICMP".

```
1 Algoritmo tarea_10_1
2     definir paquete ,protocolo Como Entero
3     Escribir "ingrese el tamaño del paquete en bytes"
4     leer paquete
5     escribir "1.ICMP  2.TCP"
6     Escribir "ingrese el tipo de protocolo"
7     leer protocolo
8     si protocolo=1 Entonces
9         si paquete ≥ 65535 Entonces
10            Escribir "ataque ping de la muerte"
11        sino
12            Escribir "ping normal  "
13        FinSi
14    SiNo
15        si protocolo=2 Entonces
16            Escribir "paquete  ICMP normal"
17        FinSi
18    FinSi
19 FinAlgoritmo
20
```

\*\*\* Ejecución Iniciada. \*\*\*

ingrese el tamaño del paquete en bytes

> 456789

1.ICMP 2.TCP

ingrese el tipo de protocolo

> 1

ataque ping de la muerte

\*\*\* Ejecución Finalizada. \*\*\*

```
1 import java.io.*;
2 public class tarea_10_1 {
3
4     public static void main(String[] args) throws IOException {
5         BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
6         int paquete, protocolo;
7
8         System.out.println("Ingrese el tamaño del paquete en bytes");
9         paquete = Integer.parseInt(br.readLine());
10
11         System.out.println("1.ICMP 2.TCP");
12         System.out.println("Ingrese el tipo de protocolo");
13         protocolo = Integer.parseInt(br.readLine());
14
15         if (protocolo == 1) {
16             if (paquete > 65535)
17                 System.out.println("Ataque ping de la muerte");
18             else
19                 System.out.println("Ping normal");
20         }
21         else if (protocolo == 2) {
22             System.out.println("Paquete TCP normal");
23         }
24     }
25 }
```

Problems Javadoc Declaration Console Chat Conectarse a una zona Wi-Fi

<terminated> tarea\_10\_1 [Java Application] C:\Program Files\Java\jdk-24\bin\javaw.exe (24 feb 2026 16:08:12 – 16:08:17 elapsed 0:00:05) [pid: 11612]

Ingrese el tamaño del paquete en bytes  
23456789  
1.ICMP 2.TCP  
Ingrese el tipo de protocolo  
1  
Ataque ping de la muerte

2. Clasificador de Direcciones IP (Clases) Crea un programa que reciba el primer octeto de una dirección IP (un número entre 0 y 255). • Si está entre 1 y 126, imprime: "Clase A". • Si está entre 128 y 191, imprime: "Clase B". • Si está entre 192 y 223, imprime: "Clase C". • Si es 127, imprime: "Dirección de Loopback". • Para cualquier otro rango, imprime: "Clase especial o reservada".

```
1 Algoritmo tarea_10_2
2   definir ip Como Entero
3   definir salida Como Caracter
4   escribir "ingrese la ip"
5   leer ip
6   si (ip ≥ 1 Y ip ≤ 126 )Entonces
7       escribir "clase A"
8   SiNo
9       si (ip ≥ 123 Y ip ≤ 191) Entonces
10           Escribir "clase B"
11       SiNo
12           si (ip ≥ 192 Y ip ≤ 223) Entonces
13               Escribir "clase C"
14           SiNo
15               si (ip = 127) Entonces
16                   escribir "direccion de loopback"
17               SiNo
18                   si (ip > 223) Entonces
19                       escribir "clase especial o reservada"
20                   FinSi
21               FinSi
22           FinSi
23       FinSi
24   FinSi
25 FinAlgoritmo
26
```

▶ PSeInt - Ejecutando proceso TAREA\_10\_2

```
*** Ejecución Iniciada. ***
ingrese la ip
> 78
clase A
*** Ejecución Finalizada. ***
```

```
1
2
3  import java.util.Scanner;
4  public class tarea_10_2 {
5
6      public static void main(String[] args) {
7          // TODO Auto-generated method stub
8          Scanner leer = new Scanner(System.in);
9
10         int ip;
11         System.out.println("Ingrese la ip");
12         ip=leer.nextInt();
13         if (ip >= 1 && ip <= 126) {
14             System.out.println("Clase A");
15         } else if (ip >= 123 && ip <= 191) {
16             System.out.println("Clase B");
17         } else if (ip >= 192 && ip <= 223) {
18             System.out.println("Clase C");
19         } else if (ip == 127) {
20             System.out.println("Direccion de loopback");
21         } else if (ip > 223) {
22             System.out.println("Clase especial o reservada");
23         }
24
25     }
26
27 }
28
29 }
30
```

Problems Javadoc Declaration Console Chat Conectarse a una zona Wi-Fi

<terminated> tarea\_10\_2 [Java Application] C:\Program Files\Java\jdk-24\bin\javaw.exe (24 feb 2026 16:10:18 – 16:10:25 elapsed 0:00:06) [pid: 5276]

Ingrese la ip  
178  
Clase B

3. Sistema de Prevención de Intrusos (IPS) El programa recibe dos variables: intentos\_fallidos (entero) y es\_ip\_conocida (booleano/1 o 0). • Si la IP no es conocida: o Si los intentos son mayores a 3, imprime: "Bloquear IP permanentemente". o Si son entre 1 y 3, imprime: "Solicitar Captcha". • Si la IP sí es conocida: o Si los intentos son mayores a 10, imprime: "Bloquear temporalmente por 15 min". o Si no, imprime: "Permitir reintento".

```
1 Algoritmo tarea_10_3
2
3   Definir intentos Como Entero
4   Definir conocida Como Entero
5
6   Escribir "Intentos fallidos:"
7   Leer intentos
8
9   Escribir "IP conocida? (1=Si,0=No):"
10  Leer conocida
11
12  Si conocida = 0 Entonces
13      Si intentos > 3 Entonces
14          Escribir "Bloquear IP permanentemente"
15      SiNo
16          Escribir "Solicitar Captcha"
17      FinSi
18  SiNo
19      Si intentos > 10 Entonces
20          Escribir "Bloquear temporalmente por 15 min"
21      SiNo
22          Escribir "Permitir reintento"
23      FinSi
24  FinSi
25
26 FinAlgoritmo
```

PSelnt - Ejecutando proceso TAREA\_10\_3

\*\*\* Ejecución Iniciada. \*\*\*

Intentos fallidos:

> 3

IP conocida? (1=Si,0=No):

> 1

Permitir reintento

\*\*\* Ejecución Finalizada. \*\*\*

```
1
2 import java.util.Scanner;
3 public class tarea_10_3 {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         Scanner leer = new Scanner(System.in);
8
9         int intentos,conocida;
10        System.out.println("Intentos fallidos:");
11        intentos=leer.nextInt();
12
13        System.out.println("IP conocida? (1=Si,0=No):");
14        conocida=leer.nextInt();
15
16        if (conocida == 0) {
17            if (intentos > 3) {
18                System.out.println("Bloquear IP permanentemente");
19            } else {
20                System.out.println("Solicitar Captcha");
21            }
22        } else {
23            if (intentos > 10) {
24                System.out.println("Bloquear temporalmente por 15 min");
25            } else {
26                System.out.println("Permitir reintento");
27            }
28        }
29    }
30
31 }
```

Problems Javadoc Declaration Console Chat Conectarse a una zona Wi-Fi

<terminated> tarea\_10\_3 [Java Application] C:\Program Files\Java\jdk-24\bin\javaw.exe (24 feb 2026 16:14:24 – 16:14:27 elapsed 0:00:03)

Intentos fallidos:  
9  
IP conocida? (1=Si,0=No):  
0  
Bloquear IP permanentemente

4. Validación de Segmentación de Red Un administrador quiere saber si una IP pertenece a la subred de "Contabilidad". La subred es 10.0.5.x. El programa recibe el segundo octeto y el tercer octeto. • Si el segundo octeto es 0: o Si el tercer octeto es 5, imprime: "Acceso concedido a Contabilidad". o Si no, imprime: "Acceso denegado: Subred incorrecta". • Si el segundo octeto no es 0, imprime: "Acceso denegado: Red externa".

```
1 Algoritmo tarea_10_4
2     definir octeto1,octeto2 Como Entero
3     escribir "ingrese el segundo octeto"
4     leer octeto1
5     escribir "ingrese el tercer octeto"
6     leer octeto2
7     si octeto1=0 Entonces
8         si octeto2=5 Entonces
9             Escribir "acceso concedido a contabilidad"
10        SiNo
11            escribir "acceso denegado: subred incorrecta"
12        FinSi
13    SiNo
14        si octeto1≠0 Entonces
15            Escribir "acceso denegado: red externa "
16        FinSi
17    FinSi
18
19 FinAlgoritmo
20 |
```

▶ PSeInt - Ejecutando proceso TAREA\_10\_4

\*\*\* Ejecución Iniciada. \*\*\*

ingrese el segundo octeto

> 0

ingrese el tercer octeto

> 5

acceso concedido a contabilidad

\*\*\* Ejecución Finalizada. \*\*\*



```
tarea_10_4.java x tarea_10_5.java tarea_10_6.java tarea_10_7.java tarea_10_8.java tarea_10_9.java
3 import java.util.Scanner;
4
5 public class tarea_10_4 {
6
7     public static void main(String[] args) {
8         Scanner leer = new Scanner(System.in);
9
10        int octeto1, octeto2;
11        System.out.println("Ingrese el segundo octeto:");
12        octeto1=leer.nextInt();
13
14        System.out.println("Ingrese el tercer octeto:");
15        octeto2=leer.nextInt();
16
17        if (octeto1 == 0) {
18            if (octeto2 == 5) {
19                System.out.println("Acceso concedido a contabilidad");
20            } else {
21                System.out.println("Acceso denegado: subred incorrecta");
22            }
23        } else {
24            if (octeto1 != 0) {
25                System.out.println("Acceso denegado: red externa");
26            }
27        }
28        // TODO Auto-generated method stub
29
30    }
31
32 }
33
```

Problems Javadoc Declaration Console Chat Conectarse a una zona Wi-Fi

<terminated> tarea\_10\_4 [Java Application] C:\Program Files\Java\jdk-24\bin\javaw.exe (24 feb 2026 16:18:16 – 16:18:23 elapsed 0)

Ingrese el segundo octeto:  
0  
Ingrese el tercer octeto:  
3  
Acceso denegado: subred incorrecta

5. Alerta de Tráfico Inusual Recibes la hora (0-23) y el volumen\_datos (en GB). • Si la hora es entre las 02:00 y las 05:00 (madrugada): o Si el volumen es mayor a 50 GB, imprime: "Alerta: Posible exfiltración de datos". o Si no, imprime: "Tráfico nocturno normal". • Si es en cualquier otra hora: o Si el volumen es mayor a 500 GB, imprime: "Alerta: Congestión de red". o Si no, imprime: "Tráfico bajo control".

```
1 Algoritmo tarea_10_5
2   definir hora,vold Como Entero
3   escribir "ingrese la hora"
4   leer hora
5   escribir"ingrese el tamaño del volumen en GB"
6   leer vold
7   si (hora≥2 Y horas≤5 ) Entonces
8       si vold>50 Entonces
9           escribir "Alerta: posible exfiltracion de datos"
10      sino
11          escribir"trafico normal"
12      FinSi
13  SiNo
14      si vold>50 Entonces
15          escribir "Alerta: congestion de red"
16      sino
17          escribir"trafico bajo control"
18      FinSi
19  FinSi
20
21 FinAlgoritmo
22
```



▶ PSeInt - Ejecutando proceso TAREA\_10\_5

\*\*\* Ejecución Iniciada. \*\*\*

ingrese la hora

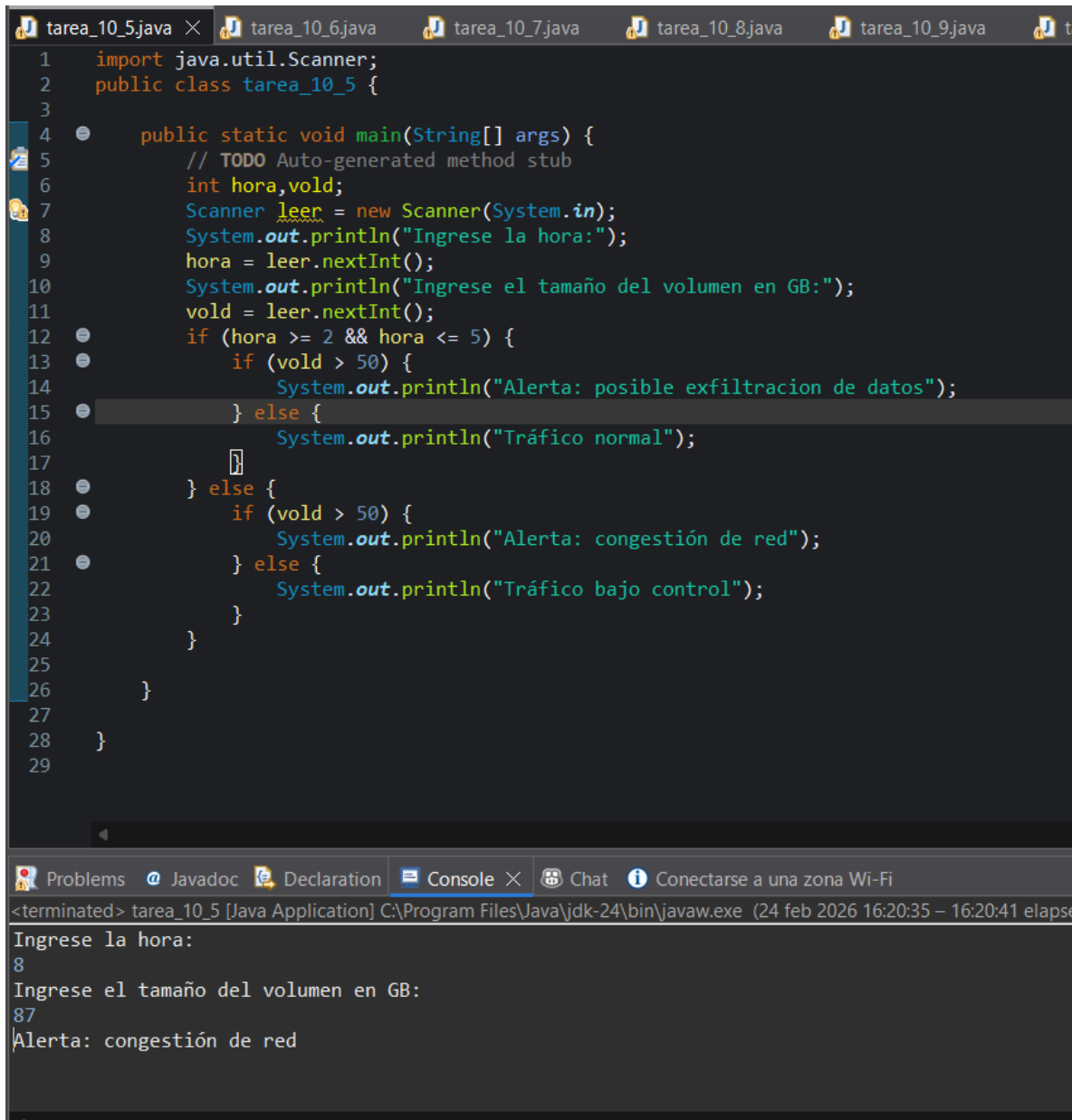
> 4

ingrese el tamaño del volumen en GB

> 78

Alerta: posible exfiltracion de datos

\*\*\* Ejecución Finalizada. \*\*\*



```

1  import java.util.Scanner;
2  public class tarea_10_5 {
3
4      public static void main(String[] args) {
5          // TODO Auto-generated method stub
6          int hora,vold;
7          Scanner leer = new Scanner(System.in);
8          System.out.println("Ingrese la hora:");
9          hora = leer.nextInt();
10         System.out.println("Ingrese el tamaño del volumen en GB:");
11         vold = leer.nextInt();
12         if (hora >= 2 && hora <= 5) {
13             if (vold > 50) {
14                 System.out.println("Alerta: posible exfiltracion de datos");
15             } else {
16                 System.out.println("Tráfico normal");
17             }
18         } else {
19             if (vold > 50) {
20                 System.out.println("Alerta: congestión de red");
21             } else {
22                 System.out.println("Tráfico bajo control");
23             }
24         }
25     }
26 }
27
28 }
29

```

Problems Javadoc Declaration Console Chat Conectarse a una zona Wi-Fi

<terminated> tarea\_10\_5 [Java Application] C:\Program Files\Java\jdk-24\bin\javaw.exe (24 feb 2026 16:20:35 – 16:20:41 elapsed)

Ingrese la hora:  
8  
Ingrese el tamaño del volumen en GB:  
87  
Alerta: congestión de red

6. Selector de Protocolo de Capa de Aplicación El programa recibe un número de puerto. Usa un switch para identificar:
- Case 21: Imprime "FTP - Transferencia de archivos".
  - Case 22: Imprime "SSH - Acceso remoto seguro".
  - Case 25: Imprime "SMTP - Envío de correo".
  - Case 80, Case 443: Imprime "Web (HTTP/HTTPS)".
  - Default: Imprime "Puerto no estándar o desconocido".

```
1 Algoritmo tarea_10_6
2   definir puerto como entero
3   escribir "ingrese el numero del puerto "
4   leer puerto
5   segun puerto Hacer
6       caso 21 :
7           escribir "FTP: transferencia de archivos"
8       caso 22 :
9           escribir "SSH: acceso remoto seguro"
10      caso 25 :
11          escribir "SMTP: envio de correo"
12      caso 80 :
13          escribir "WEB http/https"
14      caso 443:
15          escribir "WEB http/https"
16      De Otro Modo:
17          escribir "puerto no reconocido"
18      FinSegun
19 FinAlgoritmo
20
```

▶ PSeInt - Ejecutando proceso TAREA\_10\_6

```
*** Ejecución Iniciada. ***
ingrese el numero del puerto
> 25
SMTP: envio de correo
*** Ejecución Finalizada. ***
```

```
1
2 import java.util.Scanner;
3 public class tarea_10_6 {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         int puerto;
8         Scanner leer = new Scanner(System.in);
9         System.out.println("Ingrese el numero del puerto:");
10        puerto = leer.nextInt();
11        switch (puerto) {
12            case 21:
13                System.out.println("FTP: transferencia de archivos");
14                break;
15            case 22:
16                System.out.println("SSH: acceso remoto seguro");
17                break;
18            case 25:
19                System.out.println("SMTP: envio de correo");
20                break;
21            case 80:
22                System.out.println("WEB http/https");
23                break;
24            case 443:
25                System.out.println("WEB http/https");
26                break;
27            default:
28                System.out.println("Puerto no reconocido");
29                break;
30        }
31    }
32 }
```

Problems Javadoc Declaration Console Chat Conectarse a una zona Wi-Fi

terminated> tarea\_10\_6 [Java Application] C:\Program Files\Java\jdk-24\bin\javaw.exe (24 feb 2026 16:22:48 – 16:22:56 elapsed 0:00:07) [pid: 1543]

Ingrese el numero del puerto:

143

WEB http/https

7. Analizador de Logs de Firewall Recibes un código de acción del firewall (1, 2 o 3) y una variable nivel\_riesgo (1-10). Usa un switch para la acción:
- Case 1 (Permitir): Si el riesgo es > 7, imprime "Advertencia: Tráfico riesgoso permitido". Si no, "Tráfico seguro".
  - Case 2 (Bloquear): Imprime "Paquete descartado".
  - Case 3 (Cuarentena): Imprime "Paquete aislado para análisis".

```
1  Algoritmo tarea_10_7
2      definir accion,riesgo Como entero
3      escribir "ingrese "
4      leer accion
5      escribir"riesgo"
6      leer riesgo
7      segun accion Hacer
8          caso 1:
9              si riesgo>7 Entonces
10                 escribir "advertencia trafico peligroso"
11             SiNo
12                 escribir "trafico seguro"
13             FinSi
14         caso 2:
15             escribir "paquete descartado"
16         caso 3:
17             escribir "paquete aislado para analisis"
18     FinSegun
19 FinAlgoritmo
20 |
```

▶ PSeInt - Ejecutando proceso TAREA\_10\_7

\*\*\* Ejecución Iniciada. \*\*\*

ingrese

> 2

riesgo

> 6

paquete descartado

\*\*\* Ejecución Finalizada. \*\*\*

```
1
2 import java.util.Scanner;
3 public class tarea_10_7 {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         Scanner leer = new Scanner(System.in);
8
9         int accion,riesgo;
10        System.out.println("Ingrese la accion (1=Permitir, 2=Descartar, 3=Aislar):");
11        accion=leer.nextInt();
12
13        System.out.println("Ingrese el nivel de riesgo (1-10):");
14        riesgo=leer.nextInt();
15
16        switch (accion) {
17            case 1:
18                if (riesgo > 7) {
19                    System.out.println("Advertencia: tráfico peligroso");
20                } else {
21                    System.out.println("Tráfico seguro");
22                }
23                break;
24            case 2:
25                System.out.println("Paquete descartado");
26                break;
27            case 3:
28                System.out.println("Paquete aislado para análisis");
29                break;
30            default:
31                System.out.println("Acción no válida");
32        }
33    }
34 }
```

Problems Javadoc Declaration Console Chat Conectarse a una zona Wi-Fi

<terminated> tarea\_10\_7 [Java Application] C:\Program Files\Java\jdk-24\bin\javaw.exe (24 feb 2026 16:24:52 – 16:25:11 elapsed)

Ingrese la accion (1=Permitir, 2=Descartar, 3=Aislar):  
1  
Ingrese el nivel de riesgo (1-10):  
7  
Tráfico seguro

8. Priorización de Calidad de Servicio (QoS) Recibes el tipo de tráfico como un carácter ('V' para Voz, 'D' para Datos, 'G' para Gaming). • Usa un switch para el tipo. • Dentro de Case 'D': Usa un if para ver si el tamaño es > 1000MB. Si sí, prioridad "Baja"; si no, prioridad "Media". • Case 'V': Prioridad "Crítica" (sin ifs). • Case 'G': Prioridad "Alta" (sin ifs).



```
1  Algoritmo tarea_10_8
2      definir tipo Como Caracter
3      definir dato Como Entero
4
5      escribir "tipo de trafico V(voz), D(datos), G(gaming)"
6      leer tipo
7      Segun tipo Hacer
8          Caso "d":
9              escribir "ingrese el tamaño del dato"
10             leer dato
11             si dato>1000 Entonces
12                 escribir "prioridad alta"
13             SiNo
14                 escribir "prioridad baja o media"
15             FinSi
16         Caso "v":
17             escribir "prioridad critica"
18         Caso "g":
19             escribir "prioridad alta"
20         De Otro Modo:
21             escribir "ERROR"
22     FinSegun
23 FinAlgoritmo
```

▶ PSeInt - Ejecutando proceso TAREA\_10\_8

\*\*\* Ejecución Iniciada. \*\*\*

tipo de trafico V(voz), D(datos), G(gaming)

> g

prioridad alta

\*\*\* Ejecución Finalizada. \*\*\*

```
1
2 import java.util.Scanner;
3 public class tarea_10_8 {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         char tipo;
8         int dato;
9         Scanner leer = new Scanner(System.in);
10        System.out.println("tipo de trafico V(voz), D(datos), G(gaming)");
11        tipo = leer.next().charAt(0);
12        switch (tipo) {
13            case 'd':
14                System.out.println("ingrese el tamaño del dato");
15                dato = leer.nextInt();
16                if (dato > 1000) {
17                    System.out.println("prioridad alta");
18                } else {
19                    System.out.println("prioridad baja o media");
20                }
21                break;
22            case 'v':
23                System.out.println("prioridad critica");
24                break;
25            case 'g':
26                System.out.println("prioridad alta");
27                break;
28            default:
29                System.out.println("ERROR");
30                break;
31        }
32    }
33 }
```

Problems Javadoc Declaration Console Chat Conectarse a una zona Wi-Fi

<terminated> tarea\_10\_8 [Java Application] C:\Program Files\Java\jdk-24\bin\javaw.exe (24 feb 2026 16:27:15 -

tipo de trafico V(voz), D(datos), G(gaming)

v

prioridad critica

9. Detector de Extensiones Maliciosas Recibes un código entero que representa una extensión de archivo (1: .exe, 2: .docx, 3: .sh). • Usa un switch: • Case 1, Case 3: Imprime "Archivo potencialmente ejecutable. ¿Tiene permisos de admin?". (Aquí puedes anidar un if que reciba una variable es\_admin). • Case 2: Imprime "Documento de texto seguro". • Default: Imprime "Extensión no reconocida".

\*\*\* Ejecución Iniciada. \*\*\*

ingrese codigo, 1:.exe 2:.docx 3:.sh

> 1

Archivo potencialmente ejecutable. ¿Tiene permisos de admin? 1.si 2.no

> 1

tienes permiso

\*\*\* Ejecución Finalizada. \*\*\*

```
1 Algoritmo tarea_10_9
2   definir codigo,admin Como Entero
3   escribir "ingrese codigo, 1:.exe 2:.docx 3:.sh"
4   leer codigo
5   segun codigo Hacer
6       caso 1:
7           escribir "Archivo potencialmente ejecutable. ¿Tiene permisos de admin? 1.si 2.no"
8           leer admin
9           si admin =1 Entonces
10              Escribir "tienes permiso"
11          SiNo
12              si admin = 2 Entonces
13                  Escribir "no tienes permiso papu :v"
14              FinSi
15          FinSi
16       caso 2:
17           Escribir "Documento de texto seguro"
18       caso 3:
19           escribir "Archivo potencialmente ejecutable. ¿Tiene permisos de admin? 1.si 2.no"
20           leer admin
21           si admin =1 Entonces
22              Escribir "tienes permiso"
23          SiNo
24              si admin = 2 Entonces
25                  Escribir "no tienes permiso papu :v"
26              FinSi
27          FinSi
28       De Otro Modo:
29           Escribir "extension no reconocida."
```

```
2 public class tarea_10_9 {
3
4     public static void main(String[] args) {
5         // TODO Auto-generated method stub
6         Scanner leer = new Scanner(System.in);
7
8         int codigo,admin;
9         System.out.println("ingrese codigo, 1:.exe 2:.docx 3:.sh");
10        codigo=leer.nextInt();
11        switch (codigo) {
12            case 1:
13                System.out.println("Archivo potencialmente ejecutable. ¿Tiene permisos de admin? 1.si 2.no");
14                admin=leer.nextInt();
15                if (admin == 1) {
16                    System.out.println("tienes permiso");
17                } else if (admin == 2) {
18                    System.out.println("no tienes permiso papu :v");
19                }
20                break;
21            case 2:
22                System.out.println("Documento de texto seguro");
23                break;
24            case 3:
25                System.out.println("Archivo potencialmente ejecutable. ¿Tiene permisos de admin? 1.si 2.no");
26                admin=leer.nextInt();
27                if (admin == 1) {
28                    System.out.println("tienes permiso");
29                } else if (admin == 2) {
30                    System.out.println("no tienes permiso papu :v");
31                }
32                break;
33        }
34    }
35}
```

Problems Javadoc Declaration Console X Chat Conectarse a una zona Wi-Fi

<terminated> tarea\_10\_9 [Java Application] C:\Program Files\Java\jdk-24\bin\javaw.exe (24 feb 2026 16:30:05 - 16:30:09 elapsed 0:00:03) [pid: 19640]

ingrese codigo, 1:.exe 2:.docx 3:.sh

1

Archivo potencialmente ejecutable. ¿Tiene permisos de admin? 1.si 2.no

2

no tienes permiso papu :v

10. Menú de Herramientas de Red Crea un menú donde el usuario elige una opción (1-4). Usa un switch: 1. Escanear Red: Pide un rango (usa un if para ver si el rango es válido, ej. > 0). 2. Verificar Firewall: Imprime "Estado: Activo". 3. Cifrar Mensaje: Pide un nivel (1 o 2). Si es 2, imprime "Cifrado AES", si es 1, "Cifrado César". 4. Salir: Imprime "Cerrando sesión".

```
1 Algoritmo tarea_10_10
2   definir op ,rango,cifra Como Entero
3   Escribir "MENU DE HERRAMIENTAS"
4   Escribir "1.Escanear Red"
5   Escribir "2.Verificar Firewall"
6   Escribir "3.Cifrar Mensaje"
7   Escribir "4.Salir"
8   leer op
9   Segun op Hacer
10      caso 1:
11         Escribir "escribe un rango "
12         leer rango
13         si rango >0 Entonces
14             escribir "escaneo"
15         SiNo
16             escribir "error"
17         FinSi
18      caso 2:
19         Escribir "Estado: Activo"
20      caso 3:
21         Escribir "ingrese cifra 1 o 2"
22         leer cifra
23         si cifra=1 Entonces
24             Escribir "Cifrado César"
25         SiNo
26             si cifra=2 Entonces
27                 Escribir "Cifrado AES"
28             FinSi
29         FinSi
```

▶ PSeInt - Ejecutando proceso TAREA\_10\_10

\*\*\* Ejecución Iniciada. \*\*\*

MENU DE HERRAMIENTAS

1.Escanear Red

2.Verificar Firewall

3.Cifrar Mensaje

4.Salir

> 3

ingrese cifra 1 o 2

> 2

Cifrado AES

\*\*\* Ejecución Finalizada. \*\*\*

```
2 public class tarea_10_10 {
3
4     public static void main(String[] args) {
5         // TODO Auto-generated method stub
6         Scanner leer = new Scanner(System.in);
7
8         int op,rango,cifra;
9         System.out.println("MENU DE HERRAMIENTAS");
10        System.out.println("1.Escanear Red");
11        System.out.println("2.Verificar Firewall");
12        System.out.println("3.Cifrar Mensaje");
13        System.out.println("4.Salir");
14        op=leer.nextInt();
15        switch (op) {
16            case 1:
17                System.out.println("escribe un rango ");
18                rango=leer.nextInt();
19                if (rango >0) {
20                    System.out.println("escaneo");
21                } else {
22                    System.out.println("error");
23                }
24                break;
25            case 2:
26                System.out.println("Estado: Activo");
27                break;
28            case 3:
29                System.out.println("ingrese cifra 1 o 2");
30                cifra=leer.nextInt();
31                if (cifra==1) {
32                    System.out.println("Cifrado César");
33                }
34            }
35        }
36    }
37 }
```

Problems Javadoc Declaration Console Chat Conectarse a una zona Wi-Fi

<terminated> tarea\_10\_10 [Java Application] C:\Program Files\Java\jdk-24\bin\javaw.exe (24 feb 2026 16:31:44 – 16:31:50 elapsed 0:00:05)

2.Verificar Firewall  
3.Cifrar Mensaje  
4.Salir  
3  
ingrese cifra 1 o 2  
1  
Cifrado César