



**Universidad Autónoma De Tamaulipas**

**Fundamentos de Programación**

**1-N**

**Tarea 10**

**Molina Meneses Diego**

## Filtro\_ICMP:

The screenshot shows the PSelnt IDE interface. On the left, the script file `Filtro_ICMP.psc` is open, containing the following pseudocode:

```
Algoritmo Filtro_ICMP
    Definir tam Como Entero
    Definir protocolo Como Entero
    ...
    Si protocolo = 1 Entonces
        Si tam > 65535 Entonces
            Escribir "Ataque Ping de la Muerte detectado"
        SiNo
            Escribir "Paquete ICMP normal"
        FinSi
    SiNo
        Si protocolo = 2 Entonces
            Escribir "Ignorar, no es ICMP"
        SiNo
            Escribir "Protocolo no valido"
        FinSi
    FinSi
FinAlgoritmo
```

On the right, a terminal window titled "PSelnt - Ejecutando proceso FILTRO\_ICMP" shows the execution results:

```
*** Ejecución Iniciada. ***
Tamaño del paquete (bytes):
> 15
Protocolo (1=ICMP, 2=TCP):
> 1
Paquete ICMP normal
*** Ejecución Finalizada. ***
```

The screenshot shows the Eclipse IDE interface. On the left, the project structure under "Projects" shows a folder `a2241330016_tarea10` containing several Java files.

The main file `Filtro_ICMP.java` is displayed in the editor:

```
package a2241330016_tarea10;
import java.util.Scanner;
public class Filtro_ICMP {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Tamano del paquete (bytes): ");
        int tam = sc.nextInt();
        System.out.print("Protocolo (1=ICMP, 2=TCP): ");
        int protocolo = sc.nextInt();
        if (protocolo == 1) { // ICMP
            if (tam > 65535) {
                System.out.println("Ataque Ping de la Muerte detectado");
            } else {
                System.out.println("Paquete ICMP normal");
            }
        } else if (protocolo == 2) { // TCP
        }
    }
}
```

On the right, the "Output" view shows the execution results:

```
run:
Tamano del paquete (bytes): 15
Protocolo (1=ICMP, 2=TCP): 1
Paquete ICMP normal
BUILD SUCCESSFUL (total time: 6 seconds)
```

## ClasificadorIP:

PSeInt

Archivo Editar Configurar Ejecutar Ayuda

Clasificador\_IP.psc X IPS.psc Validacion\_Red.psc

```

1 Algoritmo Clasificador_IP
2   Definir octeto1 Como Entero
3
4   Escribir "Primer octeto (0 a 255): "
5   Leer octeto1
6
7   Si octeto1 ≥ 1 Y octeto1 ≤ 126 Entonces
8     Escribir "Clase A"
9   SiNo
10    Si octeto1 ≥ 128 Y octeto1 ≤ 191 Entonces
11      Escribir "Clase B"
12    SiNo
13      Si octeto1 ≥ 192 Y octeto1 ≤ 223 Entonces
14        Escribir "Clase C"
15      SiNo
16        Si octeto1 = 127 Entonces
17          Escribir "Direccion de Loopback"
18        SiNo
19          Escribir "Clase especial o reservada"
20        FinSi
21      FinSi
22    FinSi
23  FinAlgoritmo

```

PSeInt - Ejecutando proceso CLASIFICADOR\_IP

\*\*\* Ejecución Iniciada. \*\*\*

Primer octeto (0 a 255):

> 115

Clase A

\*\*\* Ejecución Finalizada. \*\*\*

No cerrar esta ventana  Siempre visible

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help a22413300

Projects x Start Page x ClasificadorIP.java x IPS.java x ValidacionRed.java x

Source History

```

1 package a2241330016_tarea10;
2 import java.util.Scanner;
3 public class ClasificadorIP {
4   public static void main(String[] args) {
5     Scanner sc = new Scanner(System.in);
6
7     System.out.print("Primer octeto (0 a 255): ");
8     int octeto1 = sc.nextInt();
9
10    if (octeto1 >= 1 && octeto1 <= 126) {
11      System.out.println("Clase A");
12    } else if (octeto1 >= 128 && octeto1 <= 191) {
13      System.out.println("Clase B");
14    } else if (octeto1 >= 192 && octeto1 <= 223) {
15      System.out.println("Clase C");
16    } else if (octeto1 == 127) {
17      System.out.println("Direccion de Loopback");
18    } else {
19      System.out.println("Clase especial o reservada");
20    }
21  }
22 }

```

Output - a2241330016\_tarea10 (run)

run:  
Primer octeto (0 a 255): 115  
Clase A  
BUILD SUCCESSFUL (total time: 4 seconds)

## IPS:

The screenshot shows the PSeInt IDE interface. On the left, the code editor displays a PSC script named 'IPS.psc' with the title 'Validacion\_Red.psc'. The script contains pseudocode for an 'IPS' algorithm. On the right, a terminal window titled 'PSeInt - Ejecutando proceso IPS' shows the execution results. The output indicates that the process started, checked for failed attempts (2), checked if the IP was known (1), allowed a retry, and completed successfully.

```

1 Algoritmo IPS
2   Definir intentos_fallidos Como Entero
3   Definir es_ip_conocida Como Logico
4   Escribir "Intentos fallidos: "
5   Leer intentos_fallidos
6   Escribir "?Es IP conocida? (1=Si, 0=No): "
7   Leer es_ip_conocida
8   Si es_ip_conocida = Falso Entonces
9     Si intentos_fallidos > 3 Entonces
10       Escribir "Bloquear IP permanentemente"
11     SiNo
12       Si intentos_fallidos ≥ 1 Y intentos_fallidos < 3 Entonces
13         Escribir "Solicitar Captcha"
14       SiNo
15         Escribir "Permitir reintento"
16       FinSi
17     FinSi
18   SiNo
19     Si intentos_fallidos > 10 Entonces
20       Escribir "Bloquear temporalmente por 15 min"
21     SiNo
22       Si intentos_fallidos > 5 Entonces
23         Escribir "Alertar al admin"
24       SiNo
25         Escribir "Permitir reintento"
26       FinSi
27     FinSi
28 FinSi

```

The screenshot shows an IDE interface with multiple tabs: 'IPS.java', 'ValidacionRed.java', and 'Start Page'. The 'IPS.java' tab is active, displaying Java code for the 'IPS' class. The code uses a Scanner to read input from System.in, prints a message asking for failed attempts, reads the value, and then checks if the IP is known. It handles three cases based on the number of failed attempts: blocking permanently for more than 10 attempts, temporarily blocking for 15 minutes for more than 5 attempts, and alerting the admin for more than 3 attempts. The 'Output' tab at the bottom shows the successful execution of the program, printing the user's input and the system's response.

```

1 package a2241330016_tarea10;
2 import java.util.Scanner;
3 public class IPS {
4   public static void main(String[] args) {
5     Scanner sc = new Scanner(System.in);
6
7     System.out.print("Intentos fallidos: ");
8     int intentos = sc.nextInt();
9
10    System.out.print("?Es IP conocida? (1=Si, 0=No): ");
11    int conocidaNum = sc.nextInt();
12    boolean esConocida = (conocidaNum == 1);
13
14    if (!esConocida) {
15      if (intentos > 3) {
16        System.out.println("Bloquear IP permanentemente");
17      } else if (intentos ≥ 1 && intentos ≤ 3) {
18        System.out.println("Solicitar Captcha");
19      } else {
20        System.out.println("Permitir reintento");
21      }
22    }
23  }
24 }

```

Output - a2241330016\_tarea10 (run)

```

run:
Intentos fallidos: 2
?Es IP conocida? (1=Si, 0=No): 1
Permitir reintento
BUILD SUCCESSFUL (total time: 6 seconds)

```

## ValidacionRed:

The screenshot shows the PSeInt environment. On the left, the pseudocode editor displays a program named 'Validacion\_Red.psc' with the following code:

```
1 Algoritmo Validacion_Red
2     Definir octeto2, octeto3 Como Entero
3
4     Escribir "Segundo octeto: "
5     Leer octeto2
6     Escribir "Tercer octeto: "
7     Leer octeto3
8
9     Si octeto2 = 0 Entonces
10        Si octeto3 = 5 Entonces
11            Escribir "Acceso concedido a Contabilidad"
12        SiNo
13            Escribir "Acceso denegado: Subred incorrecta"
14        FinSi
15    SiNo
16        Escribir "Acceso denegado: Red externa"
17    FinSi
18 FinAlgoritmo
```

On the right, a terminal window titled 'PSeInt - Ejecutando proceso VALIDACION\_RED' shows the execution results:

```
*** Ejecución Iniciada. ***
Segundo octeto:
> 0
Tercer octeto:
> 5
Acceso concedido a Contabilidad
*** Ejecución Finalizada. ***
```

The screenshot shows the Apache NetBeans IDE. On the left, the 'Projects' tool window shows a Java project structure under 'a2241330016\_tarea10'. The 'Source Packages' node contains a package named 'a2241330016\_tarea10' which includes four files: ClasificadorIP.java, Filtro\_ICMP.java, IPS.java, and ValidacionRed.java. The 'ValidacionRed.java' file is selected and its code is visible in the main editor:

```
1 package a2241330016_tarea10;
2 import java.util.Scanner;
3 public class ValidacionRed {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6
7         System.out.print("Segundo octeto: ");
8         int octeto2 = sc.nextInt();
9
10        System.out.print("Tercer octeto: ");
11        int octeto3 = sc.nextInt();
12
13        if (octeto2 == 0) {
14            if (octeto3 == 5) {
15                System.out.println("Acceso concedido a Contabilidad");
16            } else {
17                System.out.println("Acceso denegado: Subred incorrecta");
18            }
19        } else {
20        }
21    }
22}
```

At the bottom, the 'Output' tool window shows the terminal output of the run command:

```
run:
Segundo octeto: 0
Tercer octeto: 5
Acceso concedido a Contabilidad
BUILD SUCCESSFUL (total time: 4 seconds)
```

## Alerta\_Trafico:

PSeInt

Archivo Editar Configurar Ejecutar Ayuda

Alerta\_Trafico.psc X

```

1 Algoritmo Alerta_Trafico
2   Definir hora Como Entero
3   Definir volumen Como Real
4
5   Escribir "Ingrese la hora (0-23): "
6   Leer hora
7
8   Escribir "Ingrese el volumen de datos (GB): "
9   Leer volumen
10
11  Si hora ≥ 2 Y hora ≤ 5 Entonces
12    Si volumen > 50 Entonces
13      Escribir "Alerta: Posible exfiltracion de datos"
14    SiNo
15      Escribir "Trafico nocturno normal"
16    FinSi
17  SiNo
18    Si volumen > 500 Entonces
19      Escribir "Alerta: Congestion de red"
20    SiNo
21      Escribir "Trafico bajo control"
22    FinSi
23  FinSi
24 FinAlgoritmo

```

PSeInt - Ejecutando proceso ALERTA\_TRAFICO

\*\*\* Ejecución Iniciada. \*\*\*  
Ingrese la hora (0-23):> 3  
Ingrese el volumen de datos (GB):> 30  
Trafico nocturno normal  
\*\*\* Ejecución Finalizada. \*\*\*

No cerrar esta ventana  Siempre visible

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help a2241330016\_practica4 - Apache NetBeans

Projects x Start Page x Alerta\_Trafico.java x

Source History

```

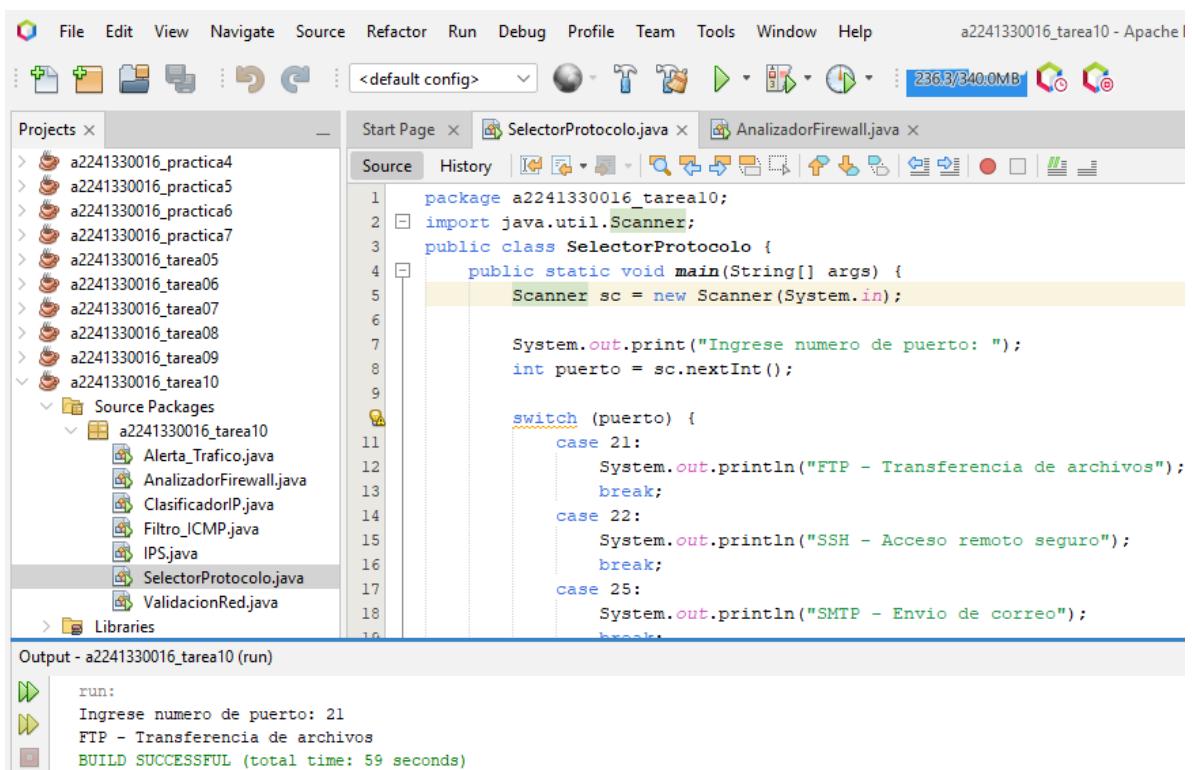
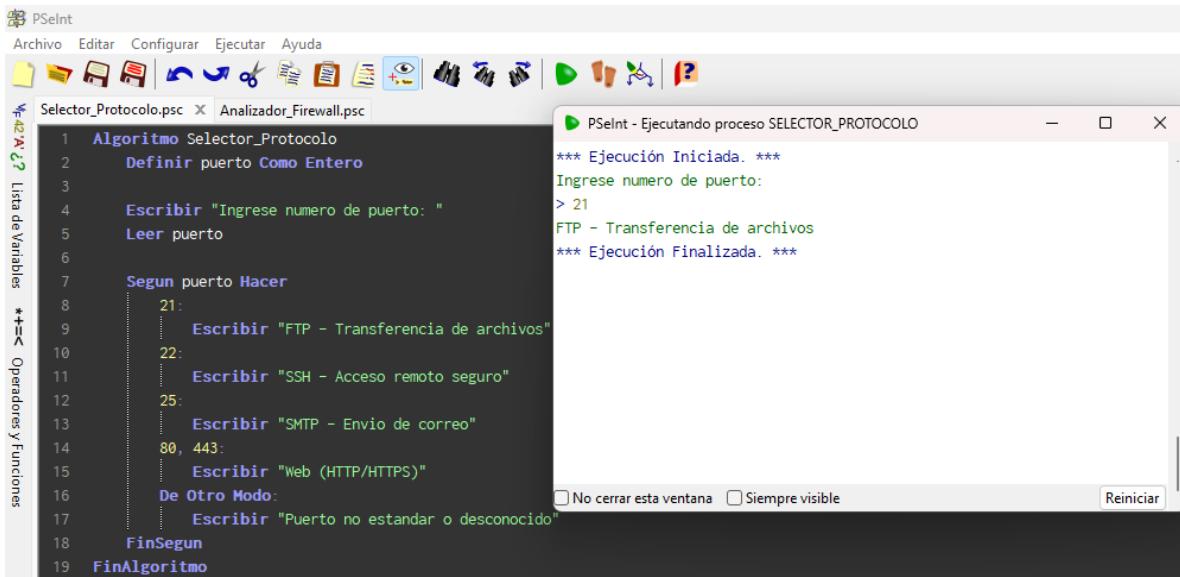
1 package a2241330016_tarea10;
2 import java.util.Scanner;
3 public class Alerta_Trafico {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6
7         System.out.print("Ingrese la hora (0-23): ");
8         int hora = sc.nextInt();
9
10        System.out.print("Ingrese el volumen de datos (GB): ");
11        double volumen = sc.nextDouble();
12
13        if (hora >= 2 && hora <= 5) { // Madrugada
14            if (volumen > 50) {
15                System.out.println("Alerta: Posible exfiltracion de datos");
16            } else {
17                System.out.println("Trafico nocturno normal");
18            }
19        } else { // Cualquier otra hora
20
21
22
23
24
25
26
27
28
29
2

```

Output - a2241330016\_tarea10 (run)

run:  
Ingrese la hora (0-23): 3  
Ingrese el volumen de datos (GB): 30  
Trafico nocturno normal  
BUILD SUCCESSFUL (total time: 7 seconds)

## SelectorProtocolo:



## Analizador\_Firewall:

PSelint

Archivo Editar Configurar Ejecutar Ayuda

Selector\_Protocolo.psc Analizador\_Firewall.psc

```

1 Algoritmo Analizador_Firewall
2   Definir codigo Como Entero
3   Definir nivel_riesgo Como Entero
4
5   Escribir "Codigo de accion (1=Permitir, 2=Bloquear, 3=Cuarentena): "
6   Leer codigo
7
8   Escribir "Nivel de riesgo (1-10): "
9   Leer nivel_riesgo
10
11   Segun codigo Hacer
12     1:
13       Si nivel_riesgo > 7 Entonces
14         |   Escribir "Advertencia: Trafico riesgoso permitido"
15       SiNo
16         |   Escribir "Trafico seguro"
17       FinSi
18     2:
19       Escribir "Paquete descartado"
20     3:
21       Escribir "Paquete aislado para analisis"
22     De Otro Modo:
23       Escribir "Codigo invalido"
24   FinSegun
25 FinAlgoritmo

```

PSelint - Ejecutando proceso ANALIZADOR\_FIREWALL

\*\*\* Ejecución Iniciada. \*\*\*

Código de acción (1=Permitir, 2=Bloquear, 3=Cuarentena):

> 1  
 Nivel de riesgo (1-10):  
 > 5  
 Trafico seguro  
 \*\*\* Ejecución Finalizada. \*\*\*

No cerrar esta ventana  Siempre visible

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help a2241330016\_tarea10 - Apache NetBeans IDE 20

Projects x Start Page x SelectorProtocolo.java x AnalizadorFirewall.java x

Source History

```

1 package a2241330016_tarea10;
2 import java.util.Scanner;
3 public class AnalizadorFirewall {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6
7         System.out.print("Codigo (1=Permitir, 2=Bloquear, 3=Cuarentena): ");
8         int codigo = sc.nextInt();
9
10        System.out.print("Nivel de riesgo (1-10): ");
11        int nivel = sc.nextInt();
12
13        switch (codigo) {
14            case 1:
15                if (nivel > 7) {
16                    System.out.println("Advertencia: Trafico riesgoso permitido");
17                } else {
18                    System.out.println("Trafico seguro");
19                }
20        }
21    }
22 }

```

Output - a2241330016\_tarea10 (run)

run:  
 Código (1=Permitir, 2=Bloquear, 3=Cuarentena): 1  
 Nivel de riesgo (1-10): 6  
 Trafico seguro  
 BUILD SUCCESSFUL (total time: 6 seconds)

QoS:

PSelInt

Archivo Editar Configurar Ejecutar Ayuda

QoS.psc X Detector\_Exensiones.psc

```

1 Algoritmo QoS
2 Definir tipo Como Caracter
3 Definir tamMB Como Real
4
5 Escribir "Tipo de trafico (V=Voz, D=Datos, G=Gaming):"
6 Leer tipo
7
8 Segun tipo Hacer
9     "D", "d":
10        Escribir "Tamaño del trafico en MB: "
11        Leer tamMB
12        Si tamMB > 1000 Entonces
13            Escribir "Prioridad: Baja"
14        SiNo
15            Escribir "Prioridad: Media"
16        FinSi
17    "V", "v":
18        Escribir "Prioridad: Critica"
19    "G", "g":
20        Escribir "Prioridad: Alta"
21 De Otro Modo:
22        Escribir "Tipo de trafico no valido"
23 FinSegun
24 FinAlgoritmo

```

PSelInt - Ejecutando proceso QoS

\*\*\* Ejecución Iniciada. \*\*\*

Tipo de trafico (V=Voz, D=Datos, G=Gaming):

> D

Tamaño del trafico en MB:

> 15

Prioridad: Media

\*\*\* Ejecución Finalizada. \*\*\*

No cerrar esta ventana  Siempre visible  Reiniciar

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help a2241330016\_tarea10 - Apache NetBeans

Projects X Start Page X QoS.java X DetectorExtensiones.java X

Source History

```

1 package a2241330016_tarea10;
2 import java.util.Scanner;
3 public class QoS {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6
7         System.out.print("Tipo de trafico (V=Voz, D=Datos, G=Gaming): ");
8         char tipo = sc.next().charAt(0);
9
10        switch (Character.toUpperCase(tipo)) {
11            case 'D':
12                System.out.print("Tamaño del trafico en MB: ");
13                double tamMB = sc.nextDouble();
14
15                if (tamMB > 1000) {
16                    System.out.println("Prioridad: Baja");
17                } else {
18                    System.out.println("Prioridad: Media");
19                }
20        }
21    }
22 }

```

Output - a2241330016\_tarea10 (run)

run:  
 Tipo de trafico (V=Voz, D=Datos, G=Gaming): D  
 Tamaño del trafico en MB: 15  
 Prioridad: Media  
 BUILD SUCCESSFUL (total time: 9 seconds)

## DetectorExensiones:

PSelnt

Archivo Editar Configurar Ejecutar Ayuda

QoS.psc Detector\_Extensiones.psc

```

1 Algoritmo Detector_Extensiones
2 Definir codigo Como Entero
3 Definir es_admin Como Logico
4
5 Escribir "Codigo de extension (1=.exe, 2=.docx, 3=.sh): "
6 Leer codigo
7
8 Segun codigo Hacer
9   1, 3:
10    Escribir "¿Tiene permisos de admin? (1=Si, 0=No):"
11    Leer es_admin
12    Escribir "Archivo potencialmente ejecutable. ¿Tiene"
13    Si es_admin = Verdadero Entonces
14      Escribir "ALERTA: Ejecutable con permisos de a"
15    SiNo
16      Escribir "Ejecutable detectado sin permisos de a"
17    FinSi
18   2:
19     Escribir "Documento de texto seguro"
20   De Otro Modo:
21     Escribir "Extension no reconocida"
22 FinSegun
23 FinAlgoritmo

```

PSelnt - Ejecutando proceso DETECTOR\_EXTENSIONES

\*\*\* Ejecución Iniciada. \*\*\*  
 Codigo de extension (1=.exe, 2=.docx, 3=.sh):  
 > 2  
 Documento de texto seguro  
 \*\*\* Ejecución Finalizada. \*\*\*

No cerrar esta ventana  Siempre visible

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help a2241330016\_tarea10 - Apache NetBeans IDE

Projects x

- a2241330016\_practica4
- a2241330016\_practica5
- a2241330016\_practica6
- a2241330016\_practica7
- a2241330016\_tarea05
- a2241330016\_tarea06
- a2241330016\_tarea07
- a2241330016\_tarea08
- a2241330016\_tarea09
- a2241330016\_tarea10
  - Source Packages
    - a2241330016\_tarea10
      - Alerta\_Trafico.java
      - AnalizadorFirewall.java
      - ClasificadorIP.java
      - DetectorExtensiones.java
      - Filtro\_ICMP.java
      - IPS.java
      - QoS.java
      - SelectorProtocolo.java

Start Page x QoS.java x DetectorExtensiones.java

Source History

```

1 package a2241330016_tarea10;
2 import java.util.Scanner;
3 public class DetectorExtensiones {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6
7         System.out.print("Codigo de extension (1=.exe, 2=.docx, 3=.sh): ");
8         int codigo = sc.nextInt();
9
10        switch (codigo) {
11            case 1: // .exe
12            case 3: // .sh
13                System.out.println("¿Tiene permisos de admin? (1=Si, 0=No): ");
14                int adminNum = sc.nextInt();
15                boolean esAdmin = (adminNum == 1);
16
17                System.out.println("Archivo potencialmente ejecutable. ¿Tiene");
18                if (esAdmin) {
19                    System.out.println("ALERTA: Ejecutable con permisos de ad");
20                }
21            }
22        }
23    }

```

Output - a2241330016\_tarea10 (run)

run:  
 Codigo de extension (1=.exe, 2=.docx, 3=.sh): 2  
 Documento de texto seguro  
 BUILD SUCCESSFUL (total time: 3 seconds)

## Menu Herramientas Red:

PSelnt

Archivo Editar Configurar Ejecutar Ayuda

Menu\_Herramientas\_Red.psc x

```

1 Algoritmo Menu_Herramientas_Red
2 Definir opcion Como Entero
3 Definir rango Como Entero
4 Definir nivel Como Entero
5
6 Escribir "==" MENU HERRAMIENTAS DE RED ==="
7 Escribir "1. Escanear Red"
8 Escribir "2. Verificar Firewall"
9 Escribir "3. Cifrar Mensaje"
10 Escribir "4. Salir"
11 Escribir "Elige una opcion (1-4): "
12 Leer opcion
13
14 Segun opcion Hacer
15   1:
16     Escribir "Ingrese un rango (ej. > 0): "
17     Leer rango
18     Si rango > 0 Entonces
19       Escribir "Escaneando red en rango: ", rango
20     SiNo
21       Escribir "Rango invalido"
22     FinSi
23   2:

```

PSelnt - Ejecutando proceso MENU\_HERRAMIENTAS\_RED

\*\*\* Ejecución Iniciada. \*\*\*

== MENU HERRAMIENTAS DE RED ==

1. Escanear Red  
2. Verificar Firewall  
3. Cifrar Mensaje  
4. Salir  
Elige una opcion (1-4):  
> 1  
Ingrese un rango (ej. > 0):  
> 2  
Escaneando red en rango: 2  
\*\*\* Ejecución Finalizada. \*\*\*

No cerrar esta ventana  Siempre visible Reiniciar

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help a2241330016\_tarea10 - A

Projects x Start Page x MenuHerramientasRed.java x

Source History

```

1 package a2241330016_tarea10;
2 import java.util.Scanner;
3 public class MenuHerramientasRed {
4   public static void main(String[] args) {
5     Scanner sc = new Scanner(System.in);
6
7     System.out.println("== MENU HERRAMIENTAS DE RED ===");
8     System.out.println("1. Escanear Red");
9     System.out.println("2. Verificar Firewall");
10    System.out.println("3. Cifrar Mensaje");
11    System.out.println("4. Salir");
12    System.out.print("Elige una opcion (1-4): ");
13    int opcion = sc.nextInt();
14
15    switch (opcion) {
16      case 1:
17        System.out.print("Ingrese un rango (ej. > 0): ");
18        int rango = sc.nextInt();
19        if (rango > 0) {

```

Output - a2241330016\_tarea10 (run)

run:  
== MENU HERRAMIENTAS DE RED ===  
1. Escanear Red  
2. Verificar Firewall  
3. Cifrar Mensaje  
4. Salir  
Elige una opcion (1-4): 1  
Ingrese un rango (ej. > 0): 2  
Escaneando red en rango: 2  
BUILD SUCCESSFUL (total time: 13 seconds)

