

# TAREA 1 - UD3

EJERCICIO 1.....	2
EJERCICIO 2.....	4
EJERCICIO 3.....	6
EJERCICIO 4.....	9

# EJERCICIO 1

/\*Escribe un programa que muestre información de todas las interfaces de red presentes en el sistema. No hace falta que muestre información de su configuración para IPv4 o IPv6 (es decir, direcciones IP, máscaras de red, direcciones de broadcast, etc.). Antes de ejecutar el programa, asegúrate de que el ordenador tenga conexión a internet, o al menos a una red local. ¿En qué cambia la salida del programa cuando se desconecta el cable de red o se desactiva la interfaz de red desde el sistema operativo? ¿Para qué tipos de interfaces se muestra información? ¿Hay alguna que esté siempre activa? ¿Cuál?\*/

```
import java.net.NetworkInterface;
import java.net.SocketException;
import java.util.Collections;
import java.util.Enumeration;

public class Main {

    public static void main(String[] args) {
        try {

            Enumeration<NetworkInterface> interfaces =
NetworkInterface.getNetworkInterfaces();

            if (interfaces == null) {
                System.out.println("No se encontraron interfaces de
red.");
                return;
            }

            System.out.println("=== LISTADO DE INTERFACES DE RED DEL
SISTEMA ===\n");

            for (NetworkInterface networkInterface :
Collections.list(interfaces)) {

                System.out.println("Nombre (Sistema): " +
networkInterface.getName());
                System.out.println("Nombre (Mostrar): " +
networkInterface.getDisplayName());
```

```

        System.out.println("¿Está activa?: " +
(networkInterface.isUp() ? "SÍ" : "NO"));
        System.out.println("¿Es Loopback?:      " +
(networkInterface.isLoopback() ? "SÍ" : "NO"));
        System.out.println("¿Es Virtual?:      " +
(networkInterface.isVirtual() ? "SÍ" : "NO"));
        System.out.println("MTU:              " +
networkInterface.getMTU());

        byte[] mac = networkInterface.getHardwareAddress();
        if (mac != null) {
            System.out.print("Dirección MAC:      ");
            StringBuilder sb = new StringBuilder();
            for (int i = 0; i < mac.length; i++) {
                sb.append(String.format("%02X%s", mac[i], (i <
mac.length - 1) ? "-" : ""));
            }
            System.out.println(sb.toString());
        } else {
            System.out.println("Dirección MAC:      No
disponible ");
        }

System.out.println("-----");
    }

    } catch (SocketException e) {
        System.err.println("Error al obtener las interfaces: " +
e.getMessage());
    }
}
}

```

## EJERCICIO 2

```

/*3.3.
Amplia el programa anterior para mostrar información de la
configuración IP para cada
interfaz, al menos la dirección o direcciones IP, y las
correspondientes máscaras de red y
direcciones de broadcast. Puedes utilizar el operador instanceof para
distinguir cuándo
una dirección IP es IPv4 (clase Inet4Address) o IPv6 (clase
Inet6Address). */

import java.net.Inet4Address;
import java.net.Inet6Address;
import java.net.InetAddress;
import java.net.InterfaceAddress;
import java.net.NetworkInterface;
import java.net.SocketException;
import java.util.Collections;
import java.util.Enumeration;
import java.util.List;

public class Main_2 {

    public static void main(String[] args) {
        try {
            Enumeration<NetworkInterface> interfaces =
NetworkInterface.getNetworkInterfaces();

            if (interfaces == null) {
                System.out.println("No se encontraron interfaces.");
                return;
            }

            System.out.println("=== CONFIGURACIÓN IP DE INTERFACES
===\n");

            for (NetworkInterface networkInterface :
Collections.list(interfaces)) {

                System.out.println("INTERFAZ: " +
networkInterface.getDisplayName() + " [" + networkInterface.getName() +
"]");

                System.out.println("-----");

                List<InterfaceAddress> listaDirecciones =
networkInterface.InterfaceAddresses();

```

```

        if (listaDirecciones.isEmpty()) {
            System.out.println("    (Sin configuración IP)");
        }

        for (InterfaceAddress interfaceAddress :
listaDirecciones) {
            InetAddress address = interfaceAddress.getAddress();

            if (address instanceof Inet4Address) {
                System.out.println("    [IPv4]");
                System.out.println("        Dirección IP:         " +
address.getHostAddress());

                System.out.println("        Máscara (Prefijo):  /" +
interfaceAddress.getNetworkPrefixLength());

                if (interfaceAddress.getBroadcast() != null) {
                    System.out.println("        Broadcast:
" + interfaceAddress.getBroadcast().getHostAddress());
                } else {
                    System.out.println("        Broadcast:
No aplica");
                }

            } else if (address instanceof Inet6Address) {
                System.out.println("    [IPv6]");

                System.out.println("        Dirección IP:         " +
address.getHostAddress());
                System.out.println("        Prefijo de Red:       /" +
interfaceAddress.getNetworkPrefixLength());

            }
            System.out.println();
        }

        System.out.println("=====\\n"
);
    }

    } catch (SocketException e) {
        System.err.println("Error: " + e.getMessage());
    }
}
}

```

## EJERCICIO 3

```

/*Probablemente, tu ordenador está situado en una red local con
direcciones privadas de tipo

C con máscara de red de 24 bits. Si es así, las direcciones son de tipo
192.168.X.Y, con

0<=X<=255 y constante para todos los hosts de la red, e Y variable con
0<=Y<=254, la

máscara de red es 255.255.255.0, y la dirección de broadcast es
192.168.X.255. Puedes

verificarlo ejecutando el comando ifconfig (en linux) o ipconfig (en
Windows) y analizando la información que muestra.

Escribe un programa que permita determinar qué direcciones dentro de la
red local son

alcanzables, es decir, cuáles están asignadas a una interfaz de red
activa. Se puede escribir

el programa específicamente para la red particular. Es decir, no hace
falta que sea genérico

y capaz de detectar automáticamente la red. Pero si hay que explicar
cómo has averiguado rango de direcciones que va a probar, y escribirá
después las direcciones IP alcanzables en

ese rango, utilizando el método isReachable. Si la máscara de red es de
24 bits, bastará

con un bucle para los valores del último byte para probar todas las
direcciones.

Si tu ordenador no está en una red local con direcciones privadas de
tipo C, seguramente

estará en una de tipo B, lo que puede complicar un poco las cosas, pero
no mucho más. En

lugar de un bucle para el valor del último byte de la dirección IP,
habrá que hacer un bucle

anidado para los dos últimos bytes. Si la máscara de red no es de 24 o
16 bits, la cosa se

complica bastante más.

```

Para crear las `InetAddress` se puede utilizar el método `static InetAddress`

`getByAddress(byte[] addr)` de `InetAddress`. Ten en cuenta que la primera dirección

del rango (bits para host todos a cero) es la dirección de la red, y que la última (bits para

host todos a uno) es la dirección de broadcast, y que ninguna de ellas corresponde

ningún host.\*/

```
import java.io.IOException;
import java.net.InetAddress;

public class Main_3 {

    public static void main(String[] args) {

        int redByte1 = 192;
        int redByte2 = 168;
        int redByte3 = 1;

        int timeout = 100;

        System.out.println("Iniciando escaneo de la red " + redByte1 +
            "." + redByte2 + "." + redByte3 + ".0/24");
        System.out.println("Por favor espere, esto puede tardar unos
momentos...");

        for (int i = 1; i < 255; i++) {

            try {

                byte[] ipAddr = new byte[] {
                    (byte) redByte1,
                    (byte) redByte2,
                    (byte) redByte3,
                    (byte) i
                };

                InetAddress address = InetAddress.getByAddress(ipAddr);

                if (address.isReachable(timeout)) {
                    System.out.println("Host activo: " +
address.getHostAddress() + " (" + address.getHostName() + ")");
                }
            }
        }
    }
}
```

```
        } catch (IOException e) {  
            System.err.println("Error al escanear " + redByte1 + "."  
+ redByte2 + "." + redByte3 + "." + i);  
        }  
    }  
  
    System.out.println("Escaneo finalizado.");  
}  
}
```



## EJERCICIO 4

```

/*Lo más normal es que tu ordenador de clase tenga una interfaz de red
cableada. Si no,
podría tener una interfaz de red inalámbrica. En cualquier caso,
escribe un programa
que obtenga la dirección de broadcast para la red local en la que está
esa interfaz.
Esto puede ser útil para algunos programas para UDP que se crearán
después. Verifica
la información de todos los NetworkInterface que se pueden obtener con
un método
estático de esta misma clase visto con anterioridad. Debe seleccionarse
una interfaz para
la que exista una dirección IP para una red local, para lo que se puede
utilizar el método
isSiteLocalAddress.*/

import java.net.InetAddress;
import java.net.InterfaceAddress;
import java.net.NetworkInterface;
import java.net.SocketException;
import java.util.Collections;
import java.util.Enumeration;
import java.util.List;

public class Main_4 {

    public static void main(String[] args) {
        System.out.println("BUSCANDO DIRECCIÓN DE BROADCAST PARA RED
LOCAL \n");

        try {

            Enumeration<NetworkInterface> interfaces =
NetworkInterface.getNetworkInterfaces();

            if (interfaces == null) {
                System.out.println("No se encontraron interfaces de
red.");
                return;
            }

            boolean encontrado = false;

            for (NetworkInterface networkInterface :
Collections.list(interfaces)) {

```

```

        if (!networkInterface.isUp() ||
networkInterface.isLoopback()) {
            continue;
        }

        List<InterfaceAddress> interfaceAddresses =
networkInterface.getInterfaceAddresses();

        for (InterfaceAddress ia : interfaceAddresses) {
            InetAddress address = ia.getAddress();

            if (address.isSiteLocalAddress()) {

                InetAddress broadcast = ia.getBroadcast();

                if (broadcast != null) {
                    System.out.println(";INTERFAZ ENCONTRADA!");
                    System.out.println(" Nombre Interfaz: " +
networkInterface.getDisplayName());
                    System.out.println(" Tu IP Local: " +
address.getHostAddress());
                    System.out.println(" DIRECCIÓN BROADCAST: "
+ broadcast.getHostAddress());

                    System.out.println("-----"
);

                    encontrado = true;
                }
            }
        }

        if (!encontrado) {
            System.out.println("No se encontro ninguna interfaz
activa con dirección de red local .");
            System.out.println("Asegurate de estar conectado a una
red Wi-Fi o Ethernet.");
        }

    } catch (SocketException e) {
        System.err.println("Error al acceder a las interfaces de
red: " + e.getMessage());
    }
}
}

```