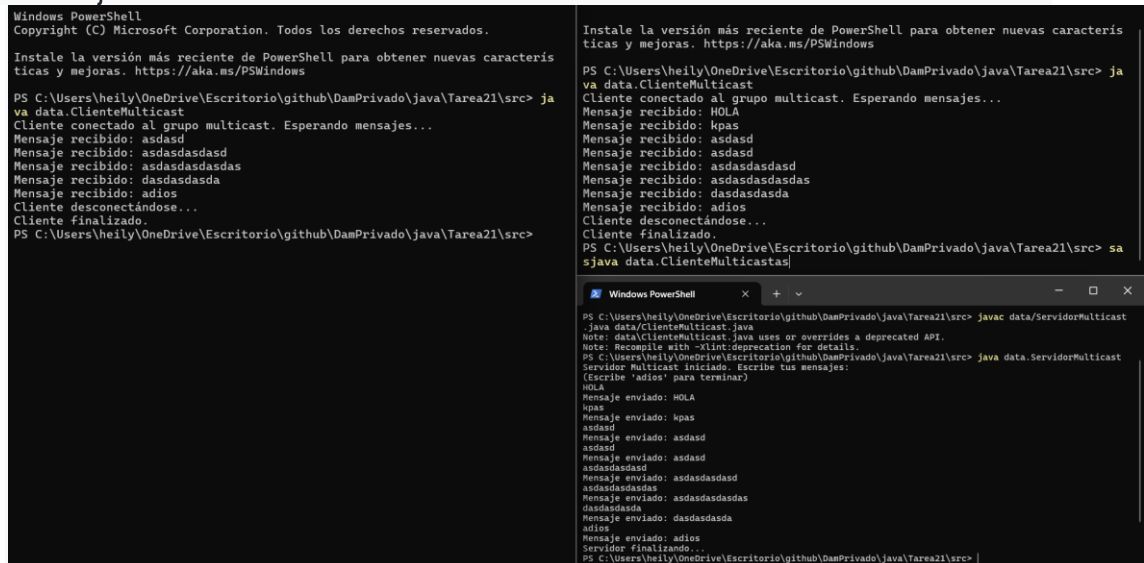


Tarea Individual 21 - Socket UDP Multicast

- La interacción entre el servidor y uno o más clientes al enviar y recibir mensajes.



```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras. https://aka.ms/PSWindows

PS C:\Users\heily\OneDrive\Escritorio\github\DamPrivado\java\Tarea21\src> java data.ClienteMulticast
Cliente conectado al grupo multicast. Esperando mensajes...
Mensaje recibido: asdasd
Mensaje recibido: asdasdasdasd
Mensaje recibido: asdasdasdasdas
Mensaje recibido: asdasdasdasda
Mensaje recibido: adios
Cliente desconectándose...
Cliente finalizado.
PS C:\Users\heily\OneDrive\Escritorio\github\DamPrivado\java\Tarea21\src>

Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras. https://aka.ms/PSWindows

PS C:\Users\heily\OneDrive\Escritorio\github\DamPrivado\java\Tarea21\src> java data.ServidorMulticast
Servidor Multicast iniciado. Escribe tus mensajes:
(Escribe 'adios' para terminar)
HOLA
Mensaje enviado: HOLA
kpas
Mensaje enviado: kpas
asdasd
Mensaje enviado: asdasd
asdasd
Mensaje enviado: asdasd
asdasdasdasd
Mensaje enviado: asdasdasdasd
asdasdasdasdas
Mensaje enviado: asdasdasdasdas
dasdasdasda
Mensaje enviado: dasdasdasda
adios
Mensaje enviado: adios
Servidor finalizando...
PS C:\Users\heily\OneDrive\Escritorio\github\DamPrivado\java\Tarea21\src>
```

package data;

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

```
public class ClienteMulticast {
```

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

```
        final String MULTICAST_IP = "225.0.0.1";
        final int PORT = 12345;
        final int BUFFER_SIZE = 1024;
```

```
        try (MulticastSocket multicastSocket = new MulticastSocket(PORT)) {
            InetAddress group = InetAddress.getByName(MULTICAST_IP);
```

```
            // Unirse al grupo multicast
            multicastSocket.joinGroup(group);
            System.out.println("Cliente conectado al grupo multicast. Esperando mensajes...");
```

```
            byte[] buffer = new byte[BUFFER_SIZE];
```

```
            while (true) {
                // Crear datagrama para recibir mensajes
                DatagramPacket packet = new DatagramPacket(buffer,
                    buffer.length);
```

```
                // Recibir mensaje del grupo multicast
```

```

        multicastSocket.receive(packet);

        // Convertir el mensaje de bytes a String
        String mensaje = new String(packet.getData(), 0, packet.getLength());

        // Mostrar el mensaje recibido
        System.out.println("Mensaje recibido: " + mensaje);

        // Finalizar si el mensaje es "adios"
        if ("adios".equalsIgnoreCase(mensaje.trim())) {
            System.out.println("Cliente desconectándose...");
            break;
        }
    }

    // Salir del grupo multicast
    multicastSocket.leaveGroup(group);
    System.out.println("Cliente finalizado.");

} catch (IOException e) {
    System.err.println("Error en el cliente multicast: " + e.getMessage());
    e.printStackTrace();
}
}
}

```

package data;

```

import java.io.IOException;
import java.net.DatagramPacket;
import java.net.InetAddress;
import java.net.MulticastSocket;
import java.util.Scanner;

public class ServidorMulticast {
    public static void main(String[] args) {
        final String MULTICAST_IP = "225.0.0.1";
        final int PORT = 12345;

        try (MulticastSocket multicastSocket = new MulticastSocket();
            Scanner scanner = new Scanner(System.in)) {

            InetAddress group = InetAddress.getByName(MULTICAST_IP);
            System.out.println("Servidor Multicast iniciado. Escribe tus mensajes:");
            System.out.println("(Escribe 'adios' para terminar)");

            String mensaje;
            while (true) {
                // Leer mensaje desde la entrada estándar
                mensaje = scanner.nextLine();

                // Convertir mensaje a bytes
                byte[] buffer = mensaje.getBytes();
            }
        }
    }
}

```

```

        // Crear datagrama para enviar al grupo
        DatagramPacket packet = new DatagramPacket(buffer, buffer.length,
group, PORT);

        // Enviar datagrama
        multicastSocket.send(packet);
        System.out.println("Mensaje enviado: " + mensaje);

        // Finalizar si el mensaje es "adios"
        if ("adios".equalsIgnoreCase(mensaje.trim())) {
            System.out.println("Servidor finalizando...");
            break;
        }
    } catch (IOException e) {
        System.err.println("Error en el servidor multicast: " + e.getMessage());
        e.printStackTrace();
    }
}
}
}

```

- El manejo de excepciones en el cliente y el servidor en caso de errores de conexión.

```

PS C:\Users\heily\OneDrive\Escritorio\github\DamPrivado\java\Tarea21\src> java data.ClienteMulticast
Cliente conectado al grupo multicast. Esperando mensajes...

PS C:\Users\heily\OneDrive\Escritorio\github\DamPrivado\java\Tarea21\src> java ServidorMulticast
Error: Could not find or load main class ServidorMulticast
Caused by: java.lang.ClassNotFoundException: ServidorMulticast

PS C:\Users\heily\OneDrive\Escritorio\github\DamPrivado\java\Tarea21\src> java data.ServidorMulticast
Servidor Multicast iniciado. Escribe tus mensajes:
(Escribe 'adios' para terminar)

Error en el servidor multicast: No route to host: sendto
java.net.NoRouteToHostException: No route to host: sendto
    at java.base/sun.nio.ch.DatagramChannelImpl.send0(Native Method)
    at java.base/sun.nio.ch.DatagramChannelImpl.sendFromNativeBuffer(DatagramChannelImpl.java:1005)
    at java.base/sun.nio.ch.DatagramChannelImpl.send(DatagramChannelImpl.java:966)
    at java.base/sun.nio.ch.DatagramChannelImpl.send(DatagramChannelImpl.java:889)
    at java.base/sun.nio.ch.DatagramChannelImpl.blockingSend(DatagramChannelImpl.java:952)
    at java.base/sun.nio.ch.DatagramSocketAdaptor.send(DatagramSocketAdaptor.java:193)
    at java.base/java.net.DatagramSocket.send(DatagramSocket.java:662)
    at data.ServidorMulticast.main(ServidorMulticast.java:33)

PS C:\Users\heily\OneDrive\Escritorio\github\DamPrivado\java\Tarea21\src>

```

Apagando el wifi abruptamente.

Las excepciones que hago en:

ClienteMulticast:

```

// Salir del grupo multicast
multicastSocket.leaveGroup(group);
System.out.println("Cliente finalizado.");
catch (SocketException e) {
    System.err.println("Error al crear el socket multicast: " + e.getMessage());
} catch (IOException e) {
    System.err.println("Error de red inesperado: " + e.getMessage());
} finally {
    if (multicastSocket != null && !multicastSocket.isClosed()) {
        multicastSocket.close();
    }
    System.out.println("Socket cerrado correctamente.");
}
}

```

-ServidorMulticast:

```
        }
    } catch (IOException e) {
        System.err.println("Error al enviar el mensaje: " + e.getMessage());
        e.printStackTrace();
    }
}
} catch (SocketException e) {
    System.err.println("Error al crear el socket multicast: " + e.getMessage());
} catch (IOException e) {
    System.err.println("Error de red inesperado: " + e.getMessage());
} finally {
    if (multicastSocket != null && !multicastSocket.isClosed()) {
        multicastSocket.close();
        System.out.println("Socket cerrado correctamente.");
    }
}
}
```