

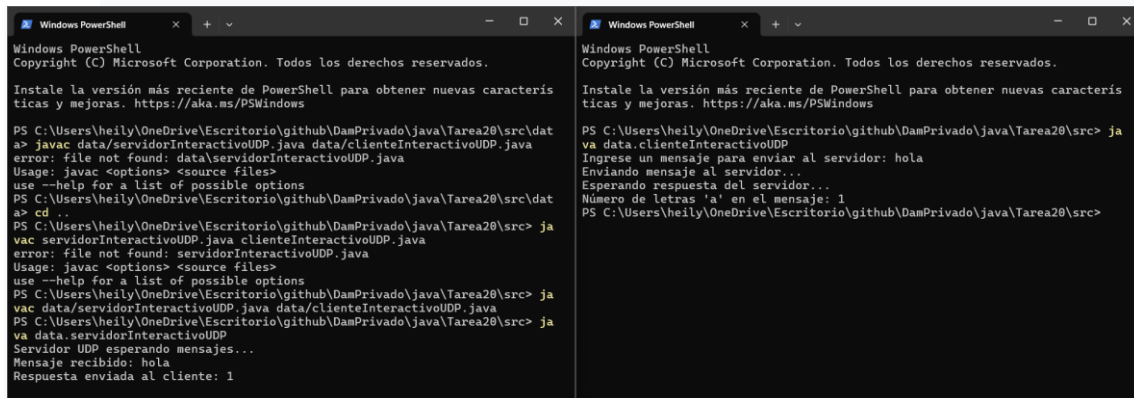
# Tarea Individual 20 - Socket UDP Interactivo

## Objetivo

Desarrollar una aplicación cliente-servidor utilizando sockets UDP que permita el envío de un mensaje desde el cliente al servidor, y que el servidor cuente cuántas veces aparece la letra 'a' en el mensaje recibido y devuelva este valor al cliente. El propósito es comprender el uso de sockets UDP, la interacción cliente-servidor y el manejo de paquetes datagrama.

## Evaluación

- La interacción entre el cliente y el servidor al enviar un mensaje y recibir la respuesta.



```
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\Tarea20\src> javac data/servidorInteractivoUDP.java data/clienteInteractivoUDP.java
error: file not found: data/servidorInteractivoUDP.java
Usage: javac <options> <source files>
use --help for a list of possible options
PS C:\Users\heily\OneDrive\Escritorio\github\DamPrivado\java\Tarea20\src> cd ..
PS C:\Users\heily\OneDrive\Escritorio\github\DamPrivado\java\Tarea20\src> java -cp . servidorInteractivoUDP.java clienteInteractivoUDP.java
error: file not found: servidorInteractivoUDP.java
Usage: javac <options> <source files>
use --help for a list of possible options
PS C:\Users\heily\OneDrive\Escritorio\github\DamPrivado\java\Tarea20\src> java -cp data/servidorInteractivoUDP.java data/clienteInteractivoUDP.java
PS C:\Users\heily\OneDrive\Escritorio\github\DamPrivado\java\Tarea20\src> java -cp data/servidorInteractivoUDP.java data/clienteInteractivoUDP.java
Servidor UDP esperando mensajes...
Mensaje recibido: hola
Respuesta enviada al cliente: 1

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

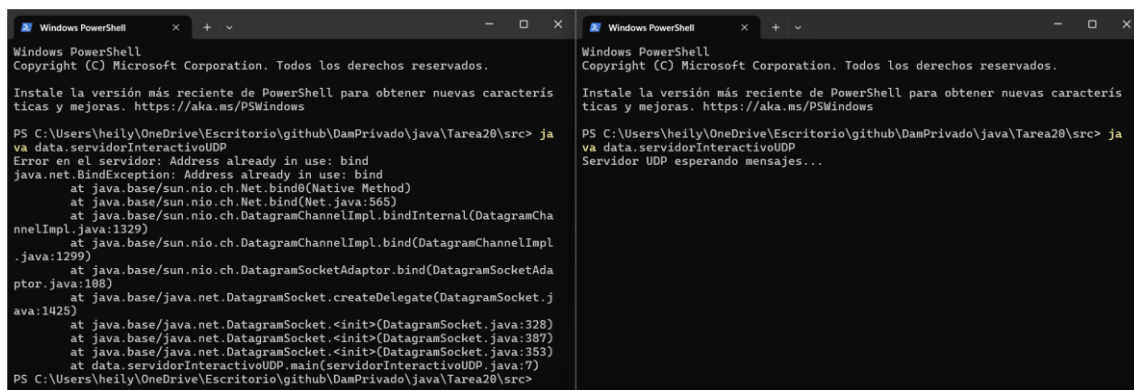
PS C:\Users\heily\OneDrive\Escritorio\github\DamPrivado\java\Tarea20\src> java data.clienteInteractivoUDP
Ingrese un mensaje para enviar al servidor: hola
Enviando mensaje al servidor...
Esperando respuesta del servidor...
Número de letras 'a' en el mensaje: 1
PS C:\Users\heily\OneDrive\Escritorio\github\DamPrivado\java\Tarea20\src>
```

- El manejo adecuado de excepciones en caso de errores en la conexión.

## 1. Excepción: BindException (Error al enlazar el puerto)

### Descripción:

- Ocurre cuando el puerto que el servidor intenta usar ya está ocupado por otro proceso.



```
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\Tarea20\src> java -cp data/servidorInteractivoUDP
Error en el servidor: Address already in use: bind
java.net.BindException: Address already in use: bind
    at java.base/sun.nio.ch.Net.bind0(Native Method)
    at java.base/sun.nio.ch.Net.bind(Net.java:565)
    at java.base/sun.nio.ch.DatagramChannelImpl.bindInternal(DatagramChannelImpl.java:1329)
    at java.base/sun.nio.ch.DatagramChannelImpl.bind(DatagramChannelImpl.java:1299)
    at java.base/sun.nio.ch.DatagramSocketAdaptor.bind(DatagramSocketAdaptor.java:168)
    at java.base/java.net.DatagramSocket.createDelegate(DatagramSocket.java:1425)
    at java.base/java.net.DatagramSocket.<init>(DatagramSocket.java:328)
    at java.base/java.net.DatagramSocket.<init>(DatagramSocket.java:387)
    at java.base/java.net.DatagramSocket.<init>(DatagramSocket.java:353)
    at data.servidorInteractivoUDP.main(servidorInteractivoUDP.java:7)
PS C:\Users\heily\OneDrive\Escritorio\github\DamPrivado\java\Tarea20\src>

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\Tarea20\src> java -cp data/servidorInteractivoUDP
Servidor UDP esperando mensajes...
```

## 2. Excepción: SocketException (Error con el socket)

## Descripción:

- Se lanza cuando hay un problema con el socket, como una desconexión abrupta.

```
run:
Ingrese un mensaje para enviar al servidor: lll
Enviando mensaje al servidor...
Esperando respuesta del servidor...

❖ERROR: El servidor no responde!
El servidor puede estar caído o no disponible.
Por favor, verifica que el servidor esté ejecutándose.
La conexión ha expirado después de 1 segundo de espera.
BUILD SUCCESSFUL (total time: 7 seconds)
```

### 3. Excepción: IOException (Error de entrada/salida)

## Descripción:

- Aparece cuando hay un problema al enviar o recibir datos entre el cliente y el servidor.

```
run:
Ingrese un mensaje para enviar al servidor: wqsda
Enviando mensaje al servidor...
Esperando respuesta del servidor...

❖ERROR: El servidor no responde!
El servidor puede estar caído o no disponible.
Por favor, verifica que el servidor esté ejecutándose.
Error al enviar o recibir datos: Connection refused
Por favor, verifica que el servidor esté ejecutándose.
BUILD SUCCESSFUL (total time: 4 seconds)
```

### 4. Excepción: UnknownHostException (Error al resolver el host)

## Descripción:

- Se lanza cuando el cliente intenta conectarse a un servidor cuya dirección no puede ser resuelta.

```
Ingrese un mensaje para enviar al servidor: jjaa
Error de E/S: Host desconocido (host-invalido)
BUILD SUCCESSFUL (total time: 10 seconds)
```

```
InetAddress address = InetAddress.getByName("host-invalido"); // Host inválido para provocar UnknownHostException
```

### 5. Excepción: PortUnreachableException (Puerto no alcanzable)

### Descripción:

- Se lanza si el cliente envía un mensaje a un puerto donde no hay un servidor escuchando.

```
run:
Ingrese un mensaje para enviar al servidor: aslkdsldlasdkasd
Enviando mensaje al servidor...
Esperando respuesta del servidor...
Error al enviar o recibir datos: Connection refused
BUILD SUCCESSFUL (total time: 4 seconds)
```

## 6. Excepción: Exception (Error inesperado)

### Descripción:

- Se utiliza para capturar errores genéricos o casos no previstos.

---

```
run:
Ingrese un mensaje para enviar al servidor: a
Enviando mensaje al servidor...
Error inesperado en el cliente: Address not set
BUILD SUCCESSFUL (total time: 1 second)
```

Como solo pones “El manejo adecuado de excepciones en caso de errores en la conexión” y no aclaras si aparte de las imágenes te añadimos el código no te he pegado las imágenes a medida que he ido modificando el código para provocar excepciones, así que pongo el código final. (Es que me he dado cuenta que no lo he ido documentado hasta casi cuando lo acabo, entonces también ponerme a buscar ahora es como -\_- , pero si me dices que lo ponga cuando me lo corrijas lo hago):

```
package data;
```

```
import java.net.*;
```

```
import java.io.*;
```

```
import java.util.concurrent.*;
```

```

public class servidorInteractivoUDP {

    public static void main(String[] args) {

        try (DatagramSocket socket = new DatagramSocket(12345)) {

            byte[] buffer = new byte[256];

            System.out.println("Servidor UDP esperando mensajes...");

            while (true) {

                try {

                    DatagramPacket paqueteRecepcion = new DatagramPacket(buffer, buffer.length);

                    System.out.println("Esperando nuevo mensaje...");

                    socket.receive(paqueteRecepcion);

                    CompletableFuture.runAsync(() -> {

                        try {

                            String mensaje = new String(paqueteRecepcion.getData(), 0,
                                paqueteRecepcion.getLength());

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

                            int contadorA = contarLetraA(mensaje);

                            String respuesta = Integer.toString(contadorA);

                            byte[] bufferRespuesta = respuesta.getBytes();

                            InetAddress address = paqueteRecepcion.getAddress();

                            int port = paqueteRecepcion.getPort();

                            DatagramPacket paqueteEnvio = new DatagramPacket(bufferRespuesta,
                                bufferRespuesta.length, address, port);

                            socket.send(paqueteEnvio);

                            System.out.println("Respuesta enviada al cliente: " + respuesta);

                        } catch (Exception e) {

                            System.err.println("Error en el procesamiento: " + e.getMessage());
                        }
                    });
                }
            }
        }
    }
}

```

```

        }
    });

    } catch (IOException e) {
        System.err.println("Error al recibir datos: " + e.getMessage());
    }
}

} catch (SocketException e) {
    System.err.println("Problema con el socket: " + e.getMessage());
}
}

private static int contarLetraA(String mensaje) {
    int contador = 0;
    for (char c : mensaje.toCharArray()) {
        if (c == 'a' || c == 'A') {
            contador++;
        }
    }
    return contador;
}
}

package data;

import java.net.*;
import java.io.*;
import java.util.Scanner;

public class clienteInteractivoUDP {
    public static void main(String[] args) {

```

```
Scanner scanner = new Scanner(System.in);

DatagramSocket socket = null;

try {

    socket = new DatagramSocket();

    socket.setSoTimeout(2000); // Timeout de 2 segundos


    System.out.print("Ingrese un mensaje para enviar al servidor: ");

    String mensaje = scanner.nextLine();


    // Cambiar la dirección para provocar UnknownHostException
    // InetAddress address = InetAddress.getByName("host-invalid");


    InetAddress address = InetAddress.getByName("localhost");

    byte[] buffer = mensaje.getBytes();


    // Cambiar el puerto para provocar PortUnreachableException
    // DatagramPacket paqueteEnvio = new DatagramPacket(buffer, buffer.length, address,
54321);


    DatagramPacket paqueteEnvio = new DatagramPacket(buffer, buffer.length, address,
12345);

    System.out.println("Enviando mensaje al servidor...");

    socket.send(paqueteEnvio);


    buffer = new byte[256];

    DatagramPacket paqueteRecepcion = new DatagramPacket(buffer, buffer.length);


    System.out.println("Esperando respuesta del servidor...");

    socket.receive(paqueteRecepcion);
```

```

        String respuesta = new String(paqueteRecepcion.getData(), 0,
paqueteRecepcion.getLength());

        System.out.println("Respuesta del servidor: " + respuesta);

    } catch (UnknownHostException e) {

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

        System.err.println("Error: El servidor no responde (Timeout)");
    } catch (SocketException e) {

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

        System.err.println("Error de E/S: " + e.getMessage());
    } finally {

        if (socket != null && !socket.isClosed()) {

            socket.close();

        }

        scanner.close();

    }

}

```