



**FACULTAD DE INGENIERÍA  
UAEMéx**



**Universidad Autónoma del Estado de  
México**

**Facultad de Ingeniería**

**Tipos de sistemas operativos:  
Copia remota**

**Alumno:  
Jiménez Rosas Victor Manuel**

## **Problemática.**

El desafío que planteamos se centra en la imperativa necesidad de realizar copias de archivos de manera remota, lo cual implica una comunicación efectiva entre los procesos mediante el uso de sockets, siguiendo un esquema cliente-servidor riguroso.

En términos operativos, el proceso se llevará a cabo enviando la ubicación del archivo que se desea copiar o leer.

Una vez que el servidor reciba esta instrucción, se encargará de ejecutar la tarea solicitada por el cliente y proporcionará el archivo deseado o una confirmación de que la copia se ha realizado con éxito. Como resultado, tanto el archivo original como su réplica en la máquina remota serán idénticos.

## **Propuesta de solución.**

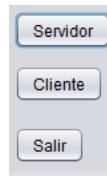
En busca de una solución a esta problemática, proponemos la implementación de un programa en lenguaje Java que aproveche las capacidades de los sockets para facilitar la copia remota de archivos.

Este programa ha sido diseñado para brindar una experiencia de usuario óptima, permitiendo la apertura del mismo archivo tanto en calidad de servidor como de cliente. Esto agrega un nivel adicional de comodidad y flexibilidad a la operación.

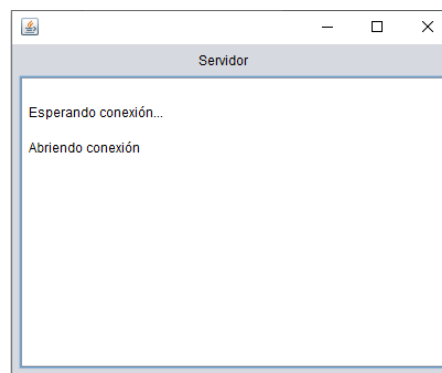
Además, la robustez inherente de Java y su capacidad para funcionar en sistemas operativos Windows garantizan un rendimiento sólido y confiable en todo momento.

## El algoritmo a seguir es simple:

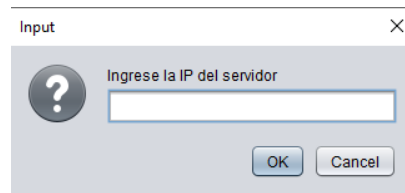
- Abrir el programa.
- Elegir entre las opciones “Servidor” y “cliente”.



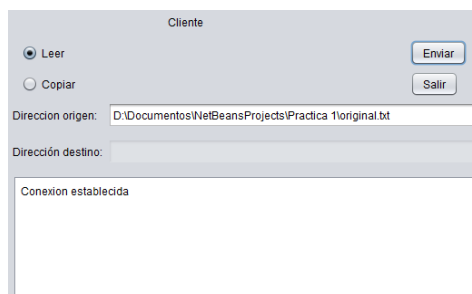
- En caso de elegir “Servidor” el programa desplegará una ventana con un log en blanco que mostrará la frase “Esperando conexión...” a la espera de que el cliente se conecte y en caso de ser contactado por un cliente mostrará el mensaje “Abriendo conexión”.



- En caso de elegir “Cliente” se pedirá ingresar la IP del servidor, que será usada posteriormente para realizar una prueba de conectividad.

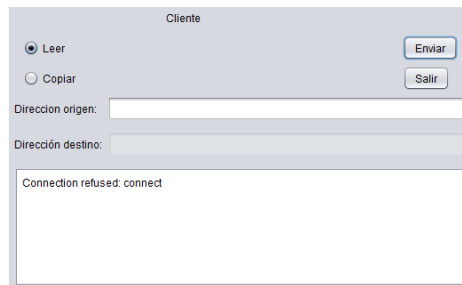


- Sabrá que la prueba fue un éxito cuando observe que la caja de texto de la ruta de origen otorga la ruta a un archivo de prueba y el log del cliente muestra el mensaje de “Conexión establecida.”

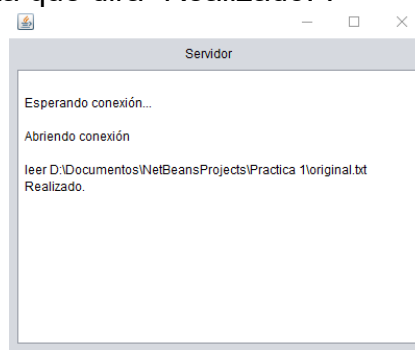


- Por otro lado, se dará a conocer que existe un error en la conexión con el mensaje de error “Connection refused: connect” que por lo general significará

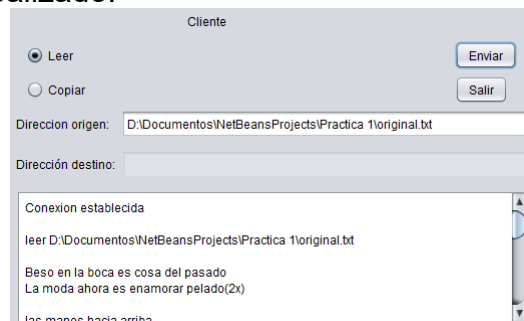
que el cliente no logró encontrar el servidor, ya sea por fallas en la conexión de internet o porque el servidor no se encuentra abierto.



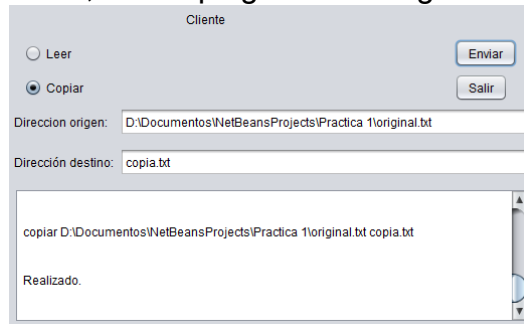
- Una vez que se ha iniciado correctamente el cliente y el servidor se puede proceder a la lectura y/o copia de los archivos, el programa genera de forma automática un archivo de prueba llamado “original.txt” que se encontrará de forma nativa en la carpeta donde se ejecute el programa, cabe aclarar que el programa solo maneja archivos en formato txt, tal vez sea necesario abrir dos veces el programa en caso de aparecer la pantalla de permitir acceso a la red.
- Para realizar una lectura se debe seleccionar la opción “Leer” del panel superior, posteriormente se deberá ingresar la ruta del archivo remoto (o usar el que proporciona por defecto el programa) y dar clic en el botón “Enviar”.
- Una vez que se ha enviado el comando el servidor recibirá la orden que será mostrada en el panel de logs, mostrando la información esencial para la realización de la tarea y un mensaje de confirmación una vez que se haya enviado la respuesta que dirá “Realizado.”.



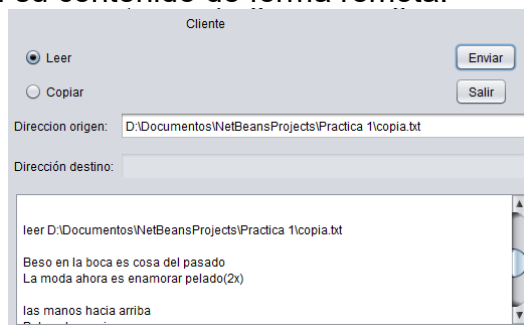
- Una vez que la respuesta ha sido recibida por el cliente, se desplegará en el log el archivo leído seguido de un mensaje de confirmación que dirá “Realizado.” O en su defecto, si se trata de una copia solo mostrara el mensaje de “Realizado.”



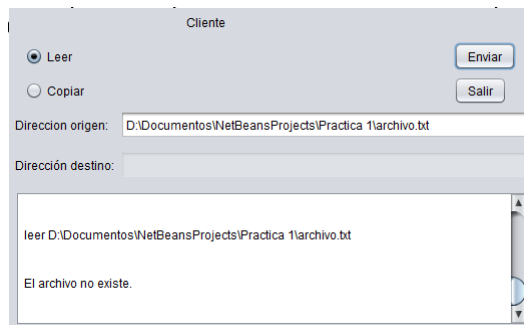
- Al seleccionar la opción de copia se habilitará una segunda caja de texto que permitirá al usuario ingresar el nombre del archivo que será la copia, al dar clic en el botón enviar, se desplegará en el log un mensaje de confirmación.



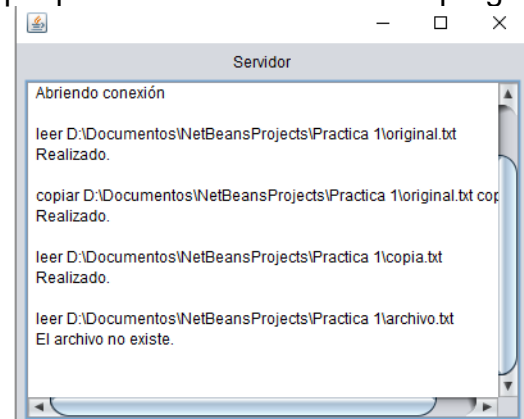
- Para confirmar que la copia se realizó con éxito se puede leer el archivo, confirmando así su contenido de forma remota.



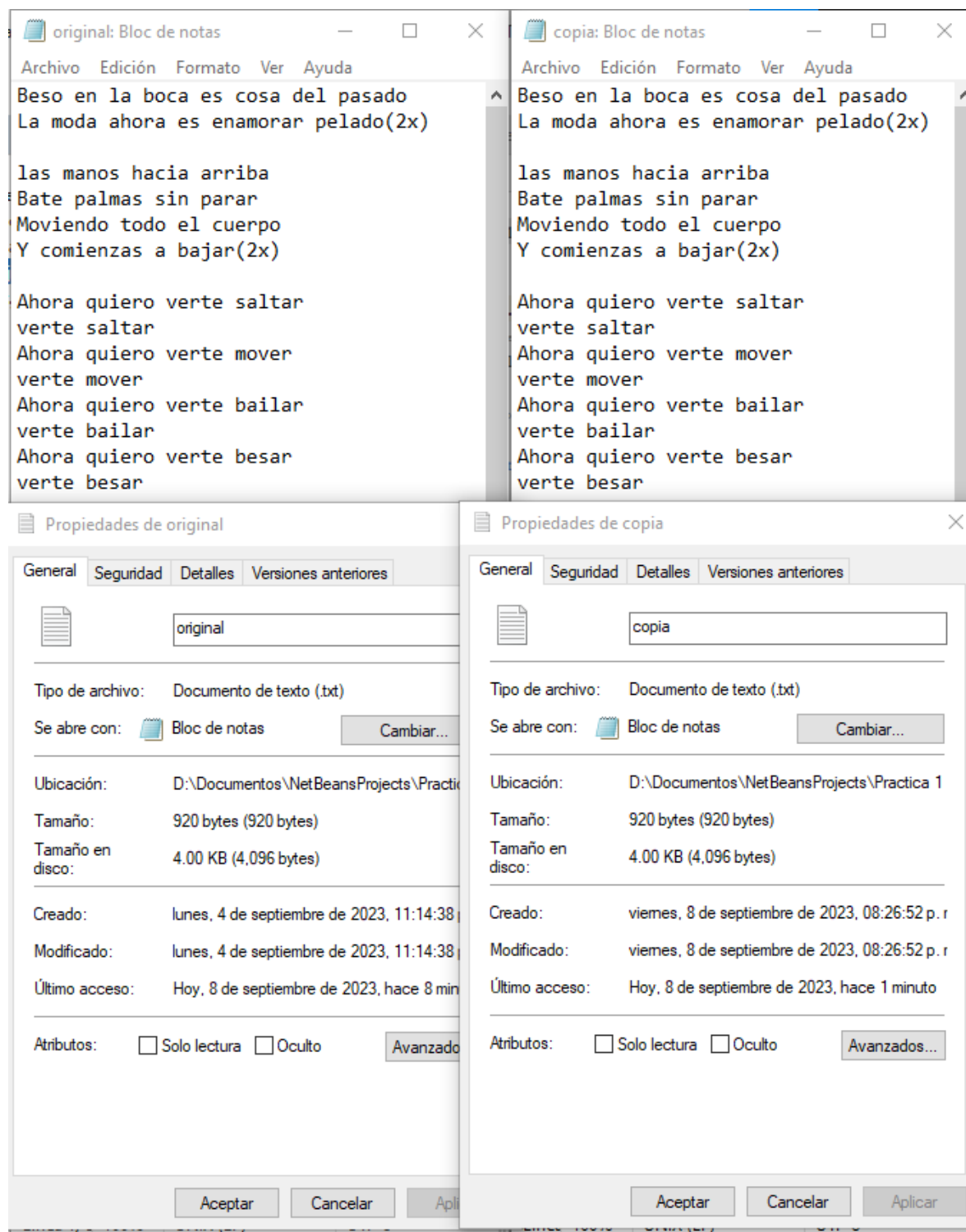
- En caso de intentar leer un archivo que no exista se desplegará un mensaje indicándolo.

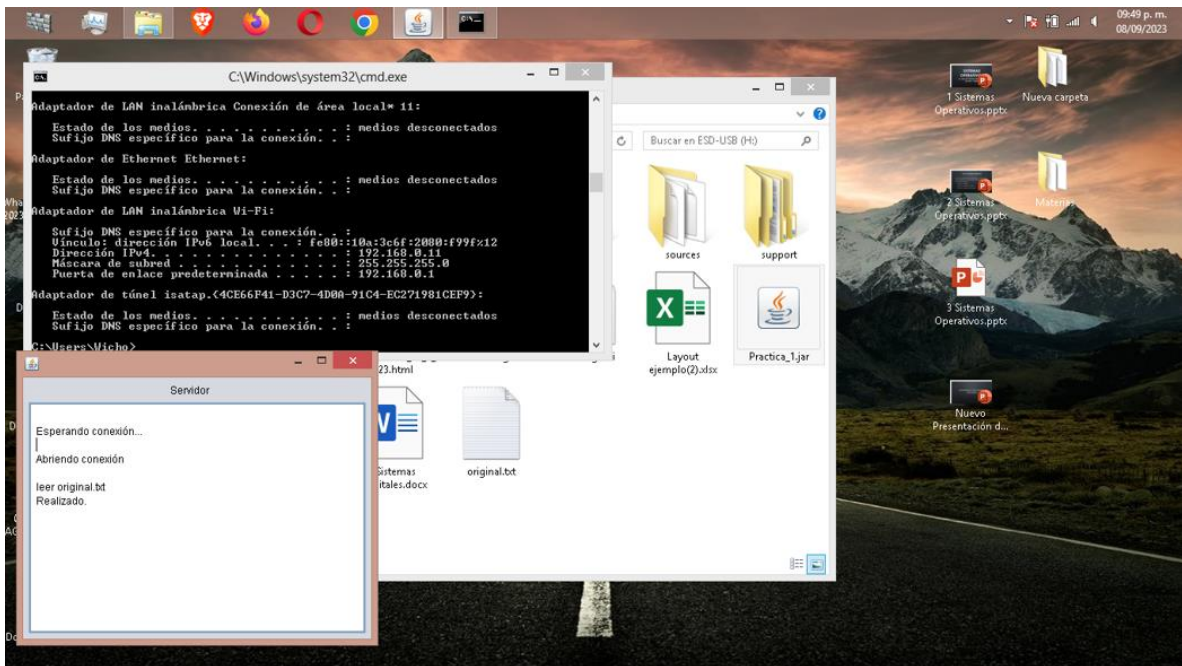
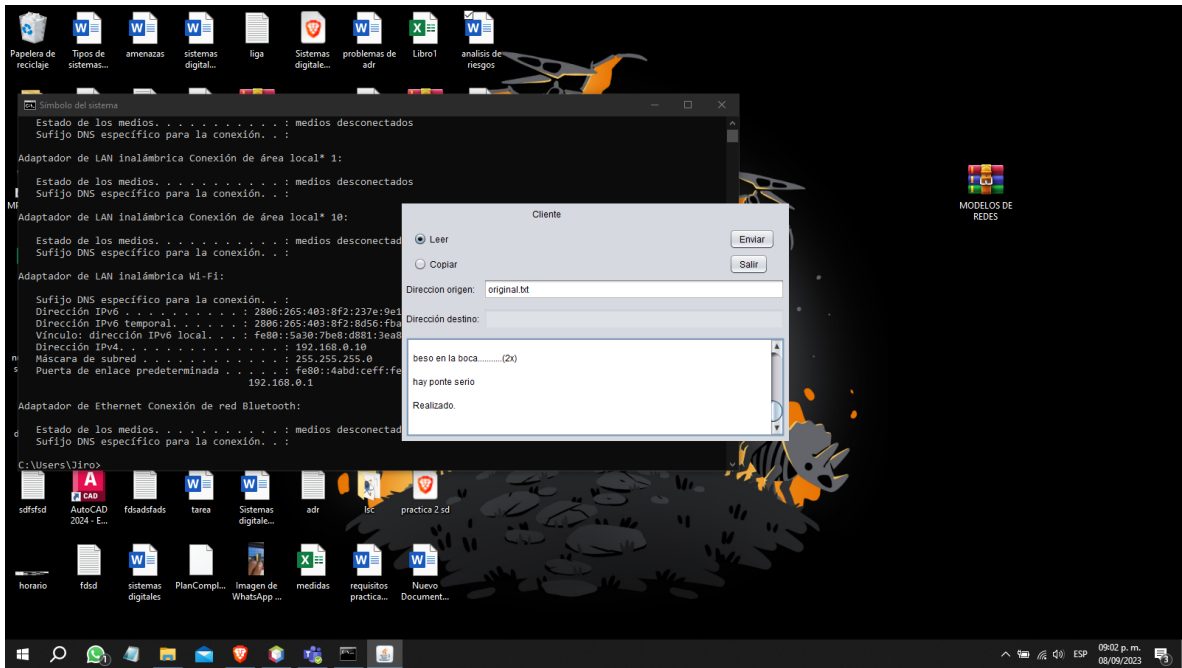


- Por último, todos los movimientos que se realicen serán captados por el log del servidor, lo que permitirá ver el avance del programa.



## Comparación entre archivos:





# Código

## Interfaz

```
import javax.swing.JOptionPane;

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this
license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GuiForms/JFrame.java to edit this template
 */

/**
 *
 * @author Jiro
 */
public class Interfaz extends javax.swing.JFrame {
    Cliente c;
    Servidor s;
    /**
     * Creates new form Interfaz
     */
    public Interfaz() {
        initComponents();
        this.setLocationRelativeTo(null);
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jButton1 = new javax.swing.JButton();
        jButton2 = new javax.swing.JButton();
        jButton3 = new javax.swing.JButton();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setUndecorated(true);

        jButton1.setText("Cliente");
        jButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton1ActionPerformed(evt);
            }
        });

        jButton2.setText("Servidor");
        jButton2.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton2ActionPerformed(evt);
            }
        })
    }
}
```



```

});

jButton3.setText("Salir");
jButton3.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton3ActionPerformed(evt);
    }
});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(12, 12, 12)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jButton1)
                .addComponent(jButton3)
                .addComponent(jButton2))
            .addGap(18, 18, 18)
        )
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(12, 12, 12)
            .addComponent(jButton2)
            .addGap(18, 18, 18)
            .addComponent(jButton1)
            .addGap(18, 18, 18)
            .addComponent(jButton3)
        )
);

pack();
} // </editor-fold>

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    String ip="";
    ip = JOptionPane.showInputDialog("Ingrese la IP del servidor");
    if(ip.matches("^(25[0-5]|(2[0-4]|1[0-9]|[1-9])?[0-9])(\\.(?!$)|$)){4}$")){
        this.c=new Cliente();

        this.setVisible(false);
        c.ip(ip);
        c.setVisible(true);
    }else{
        JOptionPane.showMessageDialog(null, "La IP no es válida", "Error",
JOptionPane.ERROR_MESSAGE);
    }
}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    System.exit(0);
}

```

```

    }

    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        this.s=new Servidor();
        this.setVisible(false);
        s.setVisible(true);
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
         * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
                javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {

            java.util.logging.Logger.getLogger(Interfaz.class.getName()).log(java.util.logging.Level.SEVERE,
            null, ex);
        } catch (InstantiationException ex) {
            java.util.logging.Logger.getLogger(Interfaz.class.getName()).log(java.util.logging.Level.SEVERE,
            null, ex);
        } catch (IllegalAccessException ex) {
            java.util.logging.Logger.getLogger(Interfaz.class.getName()).log(java.util.logging.Level.SEVERE,
            null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

            java.util.logging.Logger.getLogger(Interfaz.class.getName()).log(java.util.logging.Level.SEVERE,
            null, ex);
        }
        //</editor-fold>

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new Interfaz().setVisible(true);
            }
        });
    }
    /**/
    // Variables declaration - do not modify
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton2;
    private javax.swing.JButton jButton3;
    // End of variables declaration
}

```

## Cliente

```
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.IOException;
import java.net.ConnectException;
import java.net.Socket;
import java.net.UnknownHostException;
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this
license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
 */
/**
 *
 * @author Jiro
 */
public class Cliente extends javax.swing.JFrame {
    int puerto = 9000;
    String ip = "";
    String tarea = "leer$", msg;

    /**
     * Creates new form Cliente
     */
    public Cliente() {
        initComponents();
        this.TxtDestino.setEnabled(false);
        this.setLocationRelativeTo(null);
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        accion = new javax.swing.ButtonGroup();
        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        TxtOrigen = new javax.swing.JTextField();
        jButton1 = new javax.swing.JButton();
        jButton2 = new javax.swing.JButton();
        jLabel3 = new javax.swing.JLabel();
        TxtDestino = new javax.swing.JTextField();
        jScrollPane1 = new javax.swing.JScrollPane();
        TxtLog = new javax.swing.JTextArea();
        jButton3 = new javax.swing.JButton();
        jButton4 = new javax.swing.JButton();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
```

```

setUndecorated(true);

jLabel1.setText("Cliente");

jLabel2.setText("Direccion origen:");

TxtOrigen.setToolTipText("Dirección del archivo");
TxtOrigen.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        TxtOrigenActionPerformed(evt);
    }
});

accion.add(jRadioButton1);
jRadioButton1.setText("Copiar");
jRadioButton1.setActionCommand("copiar");
jRadioButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jRadioButton1ActionPerformed(evt);
    }
});

accion.add(jRadioButton2);
jRadioButton2.setSelected(true);
jRadioButton2.setText("Leer");
jRadioButton2.setActionCommand("leer");
jRadioButton2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jRadioButton2ActionPerformed(evt);
    }
});

jLabel3.setText("Dirección destino:");

TxtLog.setColumns(20);
TxtLog.setRows(5);
jScrollPane1.setViewportView(TxtLog);

jButton1.setText("Enviar");
jButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton1ActionPerformed(evt);
    }
});

jButton2.setText("Salir");
jButton2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton2ActionPerformed(evt);
    }
});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

```

```

        .addGroup(layout.createSequentialGroup())
        .addGap(177, 177, 177)
        .addComponent(jLabel1)
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        .addGroup(layout.createSequentialGroup())
        .addContainerGap()
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 513,
Short.MAX_VALUE)
            .addGroup(layout.createSequentialGroup())

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(jLabel3)
    .addComponent(jLabel2))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(TxtDestino)
    .addComponent(TxtOrigen))))
    .addContainerGap()
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup())
    .addGap(16, 16, 16)
    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jRadioButton2)
        .addComponent(jRadioButton1))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jButton1)
        .addComponent(jButton2))
    .addGap(19, 19, 19))
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup())
    .addContainerGap()
    .addComponent(jLabel1)
    .addGap(12, 12, 12)
    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(jRadioButton2)
        .addComponent(jButton1))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(jRadioButton1)
        .addComponent(jButton2))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(TxtOrigen, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jLabel2))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(jLabel3)
        .addComponent(TxtDestino, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

```

[illegible]

[illegible]

```

        this.TxtDestino.setText("");
        this.TxtDestino.setEnabled(false);
    }

    /**
     * @param args the command line arguments
     */
    /* public static void main(String args[]) {

        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
                javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {

            java.util.logging.Logger.getLogger(Cliente.class.getName()).log(java.util.logging.Level.SEVERE,
            null, ex);
        } catch (InstantiationException ex) {

            java.util.logging.Logger.getLogger(Cliente.class.getName()).log(java.util.logging.Level.SEVERE,
            null, ex);
        } catch (IllegalAccessException ex) {

            java.util.logging.Logger.getLogger(Cliente.class.getName()).log(java.util.logging.Level.SEVERE,
            null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

            java.util.logging.Logger.getLogger(Cliente.class.getName()).log(java.util.logging.Level.SEVERE,
            null, ex);
        }

        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new Cliente().setVisible(true);
            }
        });
    }

    /**
     // Variables declaration - do not modify
    private javax.swing.JTextField TxtDestino;
    private javax.swing.JTextArea TxtLog;
    private javax.swing.JTextField TxtOrigen;
    private javax.swing.ButtonGroup accion;
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton2;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JRadioButton jRadioButton1;
    private javax.swing.JRadioButton jRadioButton2;
    private javax.swing.JScrollPane jScrollPane1;
    // End of variables declaration
    }

```



## Servidor

```
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.IOException;
import java.net.ConnectException;
import java.net.ServerSocket;
import java.net.Socket;
import java.net.UnknownHostException;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.util.ArrayList;

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this
license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GuiForms/JFrame.java to edit this template
 */
/**
 *
 * @author Jiro
 */
public class Servidor extends javax.swing.JFrame implements Runnable {

    int puerto = 9000, c = 0;
    String log, tarea;
    File file = null;
    FileInputStream original = null;
    FileOutputStream copia = null;
    FileInputStream copiaLectura = null;
    String[] nodos = null;
    String ruta = "";
    String rutaA = "";
    String cont = "";

    /**
     * Creates new form Servidor
     */
    public Servidor() {
        initComponents();
        try {
            String contenido = "Beso en la boca es cosa del pasado\nLa moda ahora es enamorar
pelado(2x)\n\nlas manos hacia arriba\nBate palmas sin parar\nMoviendo todo el cuerpo\nY
comienzas a bajar(2x)\n\nAhora quiero verte saltar\nverte saltar\nAhora quiero verte mover\nverte
mover\nAhora quiero verte bailar\nverte bailar\nAhora quiero verte besar\nverte besar\n\nBeso en la
boca es cosa del pasado\nLa moda ahora es enamorar pelado(2x)\n\nlas manos hacia arriba\nBate
palmas sin parar\nMoviendo todo el cuerpo\nY comienzas a bajar(2x)\n\nAhora quiero verte
saltar\nverte saltar\nAhora quiero verte mover\nverte mover\nAhora quiero verte bailar\nverte
bailar\nAhora quiero verte besar\nverte besar\n\nBeso en la boca es cosa del pasado\nLa moda
ahora es enamorar pelado(2x)\n\nhay ponte serio\n\nlas manos hacia arriba\nBate palmas sin
parar\nMoviendo todo el cuerpo\nY comienzas a bajar(2x)\n\nbeso en la boca.....(2x)\n\nquero by
quero by\n\nbeso en la boca.....(2x)\n\nhay ponte serio";
            file = new File("original.txt");
```

```

        if (!file.exists()) {
            FileOutputStream original = new FileOutputStream(file);
            original.write(contenido.getBytes());
            original.close();
        }
        ruta = file.getAbsolutePath();
        rutaA = ruta.replace("original.txt", "");
        System.out.println(ruta);
    } catch (IOException e) {
        this.TxtLog.append("no se creo el archivo original");
    }
    this.setLocationRelativeTo(null);
    Thread hilo = new Thread(this);
    hilo.start();
}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jLabel3 = new javax.swing.JLabel();
    jScrollPane1 = new javax.swing.JScrollPane();
    TxtLog = new javax.swing.JTextArea();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

    jLabel3.setText("Servidor");

    TxtLog.setColumns(20);
    TxtLog.setRows(5);
    jScrollPane1.setViewportView(TxtLog);

    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(169, 169, 169)
                .addComponent(jLabel3)
                .addGap(Short.MAX_VALUE)
                .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 388, Short.MAX_VALUE)
                .addGap(Short.MAX_VALUE)
            )
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(169, 169, 169)
                .addComponent(jLabel3)
                .addGap(Short.MAX_VALUE)
            )
    );
}

```

```

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 266,
Short.MAX_VALUE)
        .addContainerGap()
    );

    pack();
} // </editor-fold>

/*public static void main(String args[]) {

    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

        java.util.logging.Logger.getLogger(Servidor.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (InstantiationException ex) {

        java.util.logging.Logger.getLogger(Servidor.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (IllegalAccessException ex) {

        java.util.logging.Logger.getLogger(Servidor.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

        java.util.logging.Logger.getLogger(Servidor.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    }
}

    java.awt.EventQueue.invokeLater(new Runnable() {

        public void run() {
            new Servidor().setVisible(true);

        }
    });
} // */

// Variables declaration - do not modify
private javax.swing.JTextArea TxtLog;
private javax.swing.JLabel jLabel3;
private javax.swing.JScrollPane jScrollPane1;
// End of variables declaration

@Override
public void run() {

```

```
//throw new UnsupportedOperationException("Not supported yet."); // Generated from
nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/GeneratedMethodBody
log = "\nEsperando conexión...\n";
this.TxtLog.append(log);
```

```
try {
    ServerSocket servidor = new ServerSocket(this.puerto);
    while (true) {
        Socket socket = servidor.accept();
        DataInputStream paquete = new DataInputStream(socket.getInputStream());

        this.log = paquete.readUTF();
        this.tarea = this.log;
        if (this.log.equals("Abriendo conexión")) {
            //mensaje de bienvenida servidor abre primero
            c = 1;
            this.TxtLog.append("\n" + this.log + "\n\n");
            DataOutputStream enviar_datos = new DataOutputStream(socket.getOutputStream());
            enviar_datos.writeUTF(ruta);
        } else if (c == 0) {
            String msg = "\nAbriendo conexión\n\n";
            c = 1;
            this.TxtLog.append(msg + log.replace("$", " ") + "\n");
            //tarea y mensaje de bienvenida cliente abre primero
            tarea(1, socket);
            //respuesta
        } else {
            System.out.println(this.tarea);
            this.TxtLog.append(log.replace("$", " ") + "\n");
            //tarea happy path
            tarea(0, socket);
        }
        socket.close();
    }
} catch (ConnectException e1) {
    this.TxtLog.append(e1.getMessage() + "\n");
} catch (UnknownHostException e2) {
    System.out.println(e2.getMessage() + "\n");
} catch (IOException e3) {
    System.out.println(e3.getMessage() + "\n");
}
}
}/**/
```

```
private void tarea(int o, Socket socket) {
    try {
        int error = 0;
        nodos = log.split("\\$");
        System.out.println(nodos[0]);
        System.out.println(nodos[1]);
        if (nodos.length == 2) {
            //lectura
            ruta = nodos[1].trim();
            file = new File(ruta);
            if (file.exists()) {
                original = new FileInputStream(ruta);
                int content;
```

```

        while ((content = original.read()) != -1) {
            cont += (char) content;
        }
        cont += "\n";
        //System.out.println("contenido");
        //System.out.println(c);
    } else {
        error = 1;
        cont = "";
        DataOutputStream enviar_datos = new DataOutputStream(socket.getOutputStream());
        enviar_datos.writeUTF(cont + "\nEl archivo no existe.");
        this.TxtLog.append("El archivo no existe.\n\n");
        cont = "";
    }
} else {
    //copia
    System.out.println(nodos[2]);
    ruta = nodos[1].trim();
    file = new File(ruta);
    String ruta2 = nodos[2].trim();
    if (file.exists()) {
        original = new FileInputStream(ruta);

        int content;

        while ((content = original.read()) != -1) {
            cont += (char) content;
        }
        //realizar copia
        file = new File(ruta2);
        if (!file.exists()) {
            file.delete();
        }
        FileOutputStream copia = new FileOutputStream(file);
        copia.write(cont.getBytes());
        copia.close();
        cont = "";
    } else {
        error = 1;
        cont = "";
        DataOutputStream enviar_datos = new DataOutputStream(socket.getOutputStream());
        enviar_datos.writeUTF(cont + "\nEl archivo de origen no existe.");
        this.TxtLog.append("El archivo de origen no existe.\n\n");
        cont = "";
    }
}
}
if (error == 0) {
    if (o == 0) {
        //respuesta happy path
        DataOutputStream enviar_datos = new DataOutputStream(socket.getOutputStream());
        enviar_datos.writeUTF(cont + "\nRealizado.");
        this.TxtLog.append("Realizado.\n\n");
        cont = "";
    } else {

```

```
        DataOutputStream enviar_datos = new DataOutputStream(socket.getOutputStream());
        enviar_datos.writeUTF("Conexión establecida$\n" + cont + "\nRealizado.");
        this.TxtLog.append("Realizado.\n\n");
        cont = "";
    }
}
error = 0;
} catch (ConnectException e1) {
    this.TxtLog.append(e1.getMessage() + "\n");
} catch (UnknownHostException e2) {
    System.out.println(e2.getMessage() + "\n");
} catch (IOException e3) {
    System.out.println(e3.getMessage() + "\n");
}
}
}
```