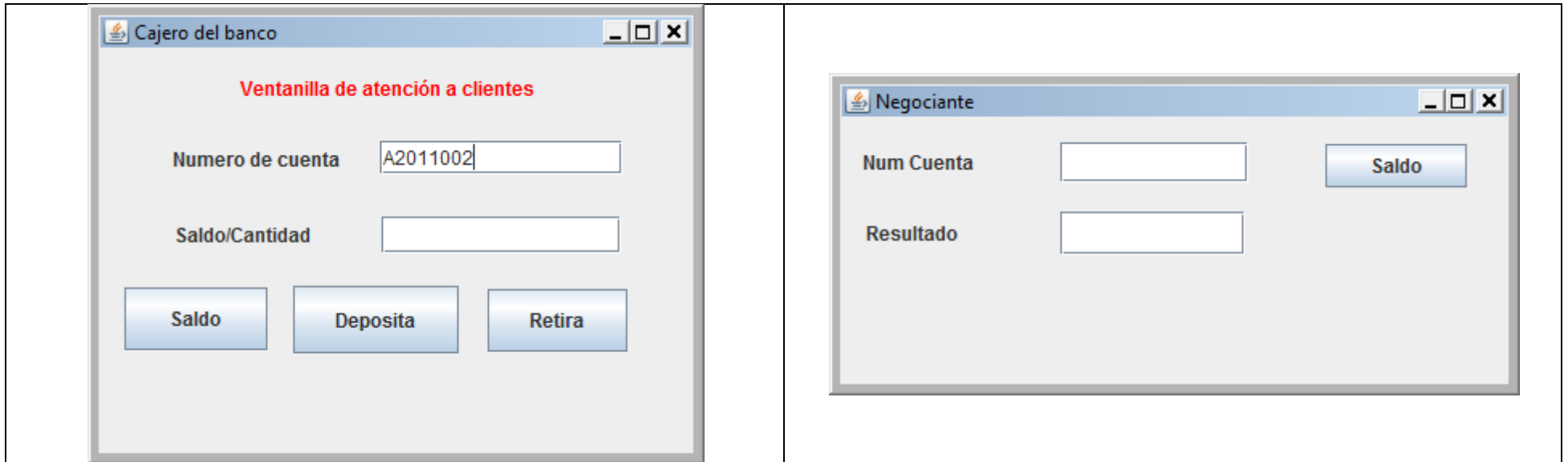


Programación Avanzada. Juan Manuel Fernández. Curso 2011  
Ejemplo de uso de sockets desde aplicaciones visuales. Usan un hilo en banco.

El software que se muestra es un ejemplo sin pulir donde se muestra el uso de sockets integrados en aplicaciones y no de manera autónoma. Partes han sido recicladas del trabajo de IBM y están en inglés. El resto es similar a los ejemplos vistos en clase, especialmente la parte del Banco y sus cuentas.

Interfaces



## Lado del banco

```
import javax.swing.SwingUtilities;
import java.awt.BorderLayout;
import javax.swing.JPanel;
import javax.swing.JFrame;
import java.awt.Dimension;
import javax.swing.JLabel;
import java.awt.Rectangle;
import java.awt.Color;
import java.awt.event.KeyEvent;
import javax.swing.SwingConstants;
import javax.swing.JTextField;
import javax.swing.JButton;

    /*
     *   Interfaz que simula un cajero bancario.
     *   Juan Manuel Fernández. Curso 2011
     */

public class IU Banco extends JFrame {

    private static final long serialVersionUID = 1L;
    private JPanel jPanelContentPane = null;
    private JLabel jLabel = null;
    private JLabel jLabel1 = null;
    private JTextField jTextField = null;
    private JLabel jLabel2 = null;
    private JTextField jTextField1 = null;
    private JButton jButton = null;
    private JButton jButton1 = null;
    private JButton jButton2 = null;

    private Banco bank; // @jve:decl-index=0:
    private Cuenta cue;
    private Remoto remo;

    private JTextField getJTextField() {
```

```

        if (jTextField == null) {
            jTextField = new JTextField();
            jTextField.setBounds(new Rectangle(174, 57, 151, 21));
        }
        return jTextField;
    }

    private JTextField getJTextField1() {
        if (jTextField1 == null) {
            jTextField1 = new JTextField();
            jTextField1.setBounds(new Rectangle(175, 104, 149, 23));
        }
        return jTextField1;
    }

    private JButton getJButton() {
        if (jButton == null) {
            jButton = new JButton();
            jButton.setBounds(new Rectangle(15, 148, 89, 39));
            jButton.setText("Saldo");
            jButton.addActionListener(new java.awt.event.ActionListener() {
                public void actionPerformed(java.awt.event.ActionEvent e) {
                    System.out.println("Saldo actionPerformed()");
                    kue = bank.buscaCuenta(jTextField.getText());
                    if (kue != null)
                        jTextField1.setText(""+kue.getSaldo());
                    else
                        jTextField1.setText("Cuenta inexistente");
                }
            });
        }
        return jButton;
    }

    private JButton getJButton1() {
        if (jButton1 == null) {
            jButton1 = new JButton();
            jButton1.setBounds(new Rectangle(120, 147, 103, 42));
            jButton1.setText("Deposita");
        }
    }

```

```

        jButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent e) {
                System.out.println("Deposita actionPerformed()");
                if (bank.deposita(jTextField.getText(), Double.parseDouble(jTextField1.getText())))
                    jTextField1.setText("Depósito realizado");
                else
                    jTextField1.setText("No se pudo realizar depósito");
            }
        });
    }
    return jButton1;
}

private JButton getJButton2() {
    if (jButton2 == null) {
        jButton2 = new JButton();
        jButton2.setBounds(new Rectangle(241, 149, 87, 39));
        jButton2.setText("Retira");
        jButton2.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent e) {
                System.out.println(" Retira actionPerformed()");
                if (bank.retira(jTextField.getText(), Double.parseDouble(jTextField1.getText())))
                    jTextField1.setText("Retiro realizado");
                else
                    jTextField1.setText("No se pudo realizar retiro");
            }
        });
    }
    return jButton2;
}

public static void main(String[] args) {
    // TODO Auto-generated method stub
    SwingUtilities.invokeLater(new Runnable() {
        public void run() {
            IUBanco thisClass = new IUBanco();
            thisClass.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
            thisClass.setVisible(true);
        }
    });
}

```

```

    });
}

public IUBanco() {
    super();
    initialize();
}

private void initialize() {
    this.setSize(384, 286);
    this.setContentPane(getJContentPane());
    this.setTitle("Cajero del banco");
    this.addWindowListener(new java.awt.event.WindowAdapter() {
        public void windowClosing(java.awt.event.WindowEvent e) {
            System.out.println("windowClosing()"); // TODO Auto-generated Event stub windowClosing()
            remo.apaga();
        }
    });
    bank = new Banco();
    remo = new Remoto(bank);
    remo.start();
}

private JPanel getJContentPane() {
    if (jContentPane == null) {
        jLabel2 = new JLabel();
        jLabel2.setBounds(new Rectangle(46, 106, 121, 19));
        jLabel2.setText("Saldo/Cantidad");
        jLabel1 = new JLabel();
        jLabel1.setBounds(new Rectangle(45, 60, 121, 17));
        jLabel1.setText("Numero de cuenta");
        jLabel = new JLabel();
        jLabel.setBounds(new Rectangle(78, 15, 201, 18));
        jLabel.setForeground(Color.red);
        jLabel.setBackground(new Color(255, 204, 153));
        jLabel.setDisplayedMnemonic(KeyEvent.VK_UNDEFINED);
        jLabel.setHorizontalTextPosition(SwingConstants.CENTER);
        jLabel.setHorizontalAlignment(SwingConstants.CENTER);
    }
}

```

```

        jLabel.setText("Ventanilla de atención a clientes");
        jContentPane = new JPanel();
        jContentPane.setLayout(null);
        jContentPane.add(jLabel, null);
        jContentPane.add(jLabel1, null);
        jContentPane.add(getJTextField(), null);
        jContentPane.add(jLabel2, null);
        jContentPane.add(getJTextField1(), null);
        jContentPane.add(getJButton(), null);
        jContentPane.add(getJButton1(), null);
        jContentPane.add(getJButton2(), null);
    }
    return jContentPane;
}
} // @jve:decl-index=0:visual-constraint="10,10"

```

```
import java.util.LinkedList;
```

```

/**
 * Clase que representa al banco, el cual maneja una lista de cuentas
 * Inicialmente crea tres cuentas A2011001 hasta 003
 * Juan Manuel Fernández. Curso 2011
 */

```

```
public class Banco {
```

```

    private LinkedList<Cuenta> misCuentas;
    private final static String razónSocial="Banco de la Ilusión";

```

```

    public Banco(){
        //System.out.println("Creando banco");
        misCuentas = new LinkedList<Cuenta>();
        temporal();
    }

```

```

    private void temporal(){
        creaCuenta("A2011001", 1000.0);
    }

```

```

    creaCuenta("A2011002", 5000.0);
    creaCuenta("A2011003", 500.0);
}

public Cuenta buscaCuenta(String nn){
    //System.out.println("solicitan buscar cuenta "+nn);
    Cuenta cue=null;
    for (int ix=0; ix<misCuentas.size();ix++){
        if (misCuentas.get(ix).getNumcta().equals(nn)){
            cue = misCuentas.get(ix);
        }
    }
    //System.out.println("saliendo de buscacuenta");
    return cue;
}

public void creaCuenta(String nc, double sa){
    misCuentas.add(new Cuenta(nc, sa));
}

public boolean retira(String nc, double ka){
    System.out.println("Solicitan retiro "+nc);
    boolean resp = false;
    Cuenta kue = buscaCuenta(nc);
    if (kue != null)
        resp = kue.retira(ka);
    return resp;
}

public boolean deposita(String nc, double ka){
    boolean resp = false;
    Cuenta kue = buscaCuenta(nc);
    if (kue != null)
        resp = kue.deposita(ka);
    return resp;
}
}

```

```
/**  
 * Clase que representa una cuenta, como las usadas antes  
 *  
 * Juan Manuel Fernández. Curso 2011  
 */
```

```
public class Cuenta {  
  
    private double saldo;  
    private String numcta;  
  
    public Cuenta(String nc, double sa){  
        numcta = nc; saldo = sa;  
    }  
  
    public synchronized boolean deposita(double kan){  
        boolean resp = false;  
        if (kan >0.0){  
            saldo += kan;  
            resp = true;  
        }  
        return resp;  
    }  
  
    public synchronized boolean retira(double kan){  
        boolean resp=false;  
        if (kan>0 && kan <= saldo){  
            saldo -= kan;  
            resp = true;  
        }  
        return resp;  
    }  
  
    public String getNumcta(){  
        return numcta;  
    }  
  
    public double getSaldo(){  
        return saldo;  
    }  
}
```



```
}
```

```
import java.io.BufferedReader;  
import java.io.File;  
import java.io.FileReader;  
import java.io.IOException;  
import java.io.InputStream;  
import java.io.InputStreamReader;  
import java.io.OutputStream;  
import java.io.PrintWriter;  
import java.net.BindException;  
import java.net.ServerSocket;  
import java.net.Socket;  
import java.util.Scanner;
```

```
/**  
 * Clase que forma un hilo para atender clients remotos  
 * Mientras atiende la interfaz local  
 * Juan Manuel Fernández Peña. Curso 2011  
 */
```

```
public class Remoto extends Thread {  
  
    private int listenPort = 3000;  
    private boolean alto = false;  
    private OutputStream outputToSocket;  
    private InputStream inputFromSocket;  
    private BufferedReader streamReader;  
    private FileReader fileReader;  
    private BufferedReader bufferedFileReader;  
    private PrintWriter streamWriter;  
    private String line;  
    private Banco elbanco;  
  
    public Remoto(Banco bb){  
        elbanco = bb;  
  
    }  
}
```

```

public void run(){
    //conectarse
    acceptConnections();
}

public void acceptConnections() {
    //System.out.println("Servidor de sockets: inicia accept");
    try {
        ServerSocket server = new ServerSocket(listenPort);
        Socket incomingConnection = null;
        while (!alto) {
            // System.out.println("Servidor de sockets: ciclo accept");
            incomingConnection = server.accept();
            handleConnection(incomingConnection);
            sleep(250);
        }
    } catch (BindException e) {
        System.out.println("Unable to bind to port " + listenPort);
    } catch (IOException e) {
        System.out.println("Unable to instantiate a ServerSocket on port: " + listenPort);
    }
    catch (InterruptedException ie){}
}

public void handleConnection(Socket incomingConnection) {
    //System.out.println("Servidor de sockets: handle inicia");
    try {
        //System.out.println("Servidor de sockets: handle creará archivos");
        //outputToSocket = incomingConnection.getOutputStream();
        inputFromSocket = incomingConnection.getInputStream();
        streamReader =
            new BufferedReader(new InputStreamReader(inputFromSocket));
        //System.out.println("Conectamos entrada a socket");
        //fileReader = new FileReader(new File(streamReader.readLine()));
        //bufferedFileReader = new BufferedReader(fileReader);
        streamWriter =
            new PrintWriter(incomingConnection.getOutputStream());
        //System.out.println("Conectamos salida a socket");
    }
}

```

```

        line = null;
        //System.out.println("Servidor de sockets: handle maneja texto");
        //while ((line = streamReader.readLine()) != null) {
        if ((line = streamReader.readLine()) != null) {
            //System.out.println("Recibí: "+line);
            atiende();
        }
        //fileReader.close();
        streamWriter.close();
        streamReader.close();
    } catch (Exception e) {
        //System.out.println("Error handling a client: " + e);
    }
}

```

```

private void atiende(){
    String resp;
    Cuenta kue;
    Scanner sca = new Scanner(line);
    //System.out.println("Atiende: scanner listo");
    String orden = sca.next();
    String cta = sca.next();
    //System.out.println("Orden: "+orden+"; cuenta: "+cta);
    // piden saldo
    if (orden.equals("Saldo")){
        //System.out.println("Piden saldo");
        kue = elbanco.buscaCuenta(cta);
        if (kue != null)
            resp = ""+kue.getSaldo();
        else
            resp = "Cuenta inexistente";
        //System.out.println("Tenemos respuesta: "+resp);
        streamWriter.println(resp);
        streamWriter.flush();
        //System.out.println("Enviamos respuesta.");
    }
    else{
        //System.out.println("No se que piden");
    }
}

```

```
}  
protected void apaga(){  
    alto = false;  
}  
}
```

## Parte del negocio

```
import javax.swing.SwingUtilities;
import java.awt.BorderLayout;
import javax.swing.JPanel;
import javax.swing.JFrame;
import java.awt.Dimension;
import javax.swing.JLabel;
import java.awt.Rectangle;
import javax.swing.JTextField;
import javax.swing.JButton;

/**
 * Clase Interfaz de cliente remoto
 * Sólo pide saldo, por ahora
 * Juan Manuel Fernández Peña. Curso 2011
 */

public class Negociante extends JFrame {

    private static final long serialVersionUID = 1L;
    private JPanel jContentPane = null;
    private JLabel jLabel = null;
    private JTextField jTextField = null;
    private JLabel jLabel1 = null;
    private JTextField jTextField1 = null;
    private JButton jButton = null;

    private Negocio nego;
    private JTextField getJTextField() {
        if (jTextField == null) {
            jTextField = new JTextField();
            jTextField.setBounds(new Rectangle(136, 15, 117, 25));
        }
        return jTextField;
    }

    private JTextField getJTextField1() {
        if (jTextField1 == null) {
```

```

        jTextField1 = new JTextField();
        jTextField1.setBounds(new Rectangle(136, 58, 115, 27));
    }
    return jTextField1;
}

private JButton getJButton() {
    if (jButton == null) {
        jButton = new JButton();
        jButton.setBounds(new Rectangle(301, 16, 88, 27));
        jButton.setText("Saldo");
        jButton.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent e) {
                System.out.println("Negociante Saldo actionPerformed()"); // TODO Auto-generated Event stub
            }
        });
    }
    return jButton;
}

public static void main(String[] args) {
    // TODO Auto-generated method stub
    SwingUtilities.invokeLater(new Runnable() {
        public void run() {
            Negociante thisClass = new Negociante();
            thisClass.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
            thisClass.setVisible(true);
        }
    });
}

public Negociante() {
    super();
    initialize();
}

```

```
private void initialize() {
    this.setSize(430, 200);
    this.setContentPane(getJContentPane());
    this.setTitle("Negociante");
}

private JPanel getJContentPane() {
    if (jContentPane == null) {
        jLabel1 = new JLabel();
        jLabel1.setBounds(new Rectangle(15, 59, 93, 24));
        jLabel1.setText("Resultado");
        jLabel = new JLabel();
        jLabel.setBounds(new Rectangle(13, 16, 97, 22));
        jLabel.setText("Num Cuenta");
        jContentPane = new JPanel();
        jContentPane.setLayout(null);
        jContentPane.add(jLabel, null);
        jContentPane.add(getJTextField(), null);
        jContentPane.add(jLabel1, null);
        jContentPane.add(getJTextField1(), null);
        jContentPane.add(getJButton(), null);
    }
    return jContentPane;
}

} // @jve:decl-index=0:visual-constraint="10,10"
```

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

```
/**
 * Clase que maneja el socket para comunicarse
 * Juan Manuel Fernández Peña. Curso 2011
 */
```

```
public class Negocio {
    protected String hostIp = "localhost";
    protected int hostPort = 3000;
    protected BufferedReader socketReader;
```

```

protected PrintWriter socketWriter;
Socket client;
String elnumcta = "A2011003";
String line=" ";

public Negocio(String aHostIp, int aHostPort) {
    hostIp = aHostIp;
    hostPort = aHostPort;
}
public static void main(String[] args) {
    Negocio remoteFileClient = new Negocio("localhost", 3000); //"127.0.0.1"
    remoteFileClient.setUpConnection();
    remoteFileClient.accion();
    remoteFileClient.tearDownConnection();

}

public void accion(){
    line=" ";
    System.err.println("Negocio entranco a acción");
    try {
        socketWriter.println("Saldo "+elnumcta);
        socketWriter.flush();
        System.out.println("Esperamos respuesta");
        if ((line = socketReader.readLine()) != null)
            System.err.println("Recibido: "+line);
        else
            System.err.println("Recibimos mensaje vacío");
        System.err.println("Terminó espera de respuesta");
    }catch(IOException e){
        System.err.println("Error reading from banco: " + e);
    }
}

public void setUpConnection() {
    System.err.println("Negocio haciendo conexión");
    try {
        client = new Socket(hostIp, hostPort);
        System.err.println("Negocio Enchufe Conectado");
    }
}

```



```
socketReader = new BufferedReader(  
    new InputStreamReader(client.getInputStream()));  
socketWriter = new PrintWriter(client.getOutputStream());  
System.err.println("Negocio archivos conectados");  
} catch (UnknownHostException e) {  
    System.err.println("Error setting up socket connection: unknown host at " + hostIp); } catch (IOException e) {  
    System.err.println("Error setting up socket connection: " + e);  
}}
```

```
public void tearDownConnection() {  
    try {  
        socketWriter.close();  
        socketReader.close();  
    } catch (IOException e) {  
        System.err.println("Error tearing down socket connection: " + e);  
    }  
}
```

```
public String opera(String nc){  
    eInumcta = nc;  
    setUpConnection();  
    accion();  
    tearDownConnection();  
    return line;  
}  
}
```