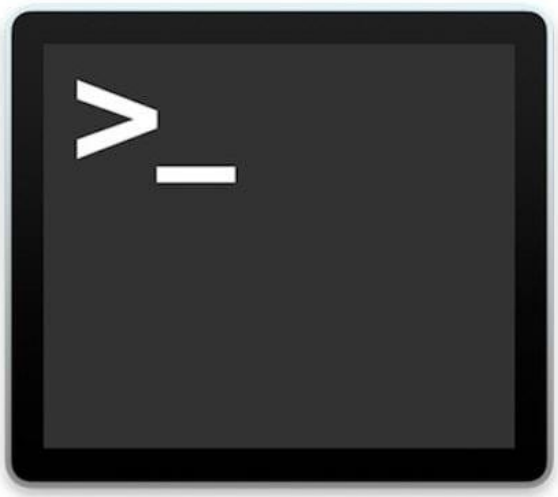
PROGRAMACIÓN DE SERVICIOS Y PROCESOS



PROGRAMACIÓN DE SERVICIOS Y PROCESOS

13/03/2019

DAVID BORREGO ASENCIO

ÍNDICE

ENUNCIADO

Deberá realizar un par de aplicaciones distribuidas desarrolladas en JAVA ambas con GUI para lo cual es preciso tener en cuenta lo siguiente:

• Una de las aplicaciones será el servidor que deberá "pensar" un número (calcula un número al azar entre 1 y 100). A esta aplicación servidor se le irán conectando aplicaciones clientes que tratarán de adivinar el número anterior.

• En las aplicaciones clientes lo primero que deberá hacer es preguntar por teclado el nombre del jugador. Lo siguiente que hará el cliente es conectarse con el servidor e irá mandando números para intentar acertar el que pensó el servidor.

• Cuando el cliente manda su número apuesta, además del número manda su nombre, para que el servidor sepa qué jugador ha enviado ese número.

• Cuando el servidor recibe un mensaje con el nombre del jugador y su apuesta devolverá un mensaje del tipo "Juan piensa que el número es el 54. Pero el número es menor a 54" o bien en el caso de acertar devolverá "Juan piensa que el número es el 50. Y HA ACERTADOOOO!!!!". En este último caso el juego finalizará.

• Todos los clientes deben ver los mensajes que se envían desde el servidor. Así irán teniendo pistas de cuál es el número con las apuestas de sus propios competidores.

• Entre apuesta y apuesta la aplicación cliente debe dejar pasar 3 segundos antes de volver a poder pedir por teclado otra apuesta.

CÓDIGO JAVA

Servidor

```
import java.awt.*;
import java.awt.event.*;
import java.io.*;
import java.net.*;
import java.util.ArrayList;
import java.util.List;
import javax.swing.*;


public class Servidor extends JFrame implements ActionListener,
Runnable
{
	private static final long serialVersionUID = 1L;
	private JScrollPane scrollpane;
	static JTextArea textarea;
	private JButton boton = new JButton("Apagar Servidor");
	private static JLabel lblPuerto;
	private ServerSocket serverSocket;
	private int numero_aleatorio;
	private List<String> conexiones;
	private boolean llave_fin;
	private static boolean llave_bucle_hilo;
	private static String ip_servidor;
	private static int puerto = 7247;
	private JScrollPane scrollpane2;
	private JTextArea textArea2;
	private JLabel label;

	public Servidor()
	{
		super("SALA DE CHAT");
		getContentPane().setForeground(Color.GREEN);
		setBackground(Color.BLACK);
		getContentPane().setBackground(Color.DARK_GRAY);
		setTitle("N\u00FAmero Oculto");
		getContentPane().setLayout(null);
		setSize(571, 415);
		setLocationRelativeTo(null);
		textarea = new JTextArea();
		textarea.setFont(new Font("Arial", Font.BOLD, 13));
		textarea.setForeground(Color.GREEN);
		textarea.setBackground(Color.BLACK);
		scrollpane = new JScrollPane(textarea);
		scrollpane.setBounds(10, 10, 382, 340);
		getContentPane().add(scrollpane);
		boton.setIcon(null);
		boton.setForeground(Color.BLACK);
		boton.setBackground(Color.RED);
		boton.setBounds(402, 9, 153, 30);
		getContentPane().add(boton);
		textarea.setEditable(false);
		boton.addActionListener(this);
		this.getRootPane().setDefaultButton(boton);
		lblPuerto = new JLabel("PUERTO: " + puerto);
```

```
            lblPuerto.setFont(new Font("Arial", Font.BOLD |
Font.ITALIC, 13));
            lblPuerto.setForeground(new Color(0, 255, 0));
            lblPuerto.setHorizontalAlignment(SwingConstants.CENTER);
            lblPuerto.setBounds(412, 82, 143, 14);
            getContentPane().add(lblPuerto);


            JLabel lblIpServidor = new JLabel("IP: " + ip_servidor);

      lblIpServidor.setHorizontalAlignment(SwingConstants.CENTER);
            lblIpServidor.setForeground(new Color(0, 255, 0));
            lblIpServidor.setFont(new Font("Arial", Font.BOLD |
Font.ITALIC, 13));
            lblIpServidor.setBounds(412, 97, 143, 14);
            getContentPane().add(lblIpServidor);

            scrollpane2 = new JScrollPane((Component) null);
            scrollpane2.setBounds(417, 185, 117, 165);
            getContentPane().add(scrollpane2);

            textArea2 = new JTextArea();
            textArea2.setForeground(Color.GREEN);
            textArea2.setFont(new Font("Arial", Font.BOLD, 13));
            textArea2.setEditable(false);
            textArea2.setBackground(Color.BLACK);
            scrollpane2.setViewportView(textArea2);

            JLabel lblUsuariosConectados = new JLabel("USUARIOS
ONLINE");

      lblUsuariosConectados.setHorizontalAlignment(SwingConstants.CENT
ER);
            lblUsuariosConectados.setForeground(Color.GREEN);
            lblUsuariosConectados.setFont(new Font("Arial", Font.BOLD,
13));
            lblUsuariosConectados.setBounds(402, 160, 153, 14);
            getContentPane().add(lblUsuariosConectados);

            label = new JLabel("");
            label.setHorizontalAlignment(SwingConstants.CENTER);
            label.setForeground(Color.GREEN);
            label.setFont(new Font("Arial", Font.BOLD, 13));
            label.setBounds(10, 361, 382, 14);
            getContentPane().add(label);

            JLabel lblNmero = new JLabel("N\u00DAMERO:");
            lblNmero.setHorizontalAlignment(SwingConstants.CENTER);
            lblNmero.setForeground(Color.GREEN);
            lblNmero.setFont(new Font("Arial", Font.BOLD | Font.ITALIC,
13));
            lblNmero.setBounds(412, 115, 143, 14);
            getContentPane().add(lblNmero);
            setDefaultCloseOperation(JFrame.DO_NOTHING_ON_CLOSE);
            setVisible(true);
            setResizable(false);


            llave_bucle_hilo = true;
            llave_fin = false;
```

```
            conexiones = new ArrayList<String>();
            numero_aleatorio = (int) (Math.random() * 100) + 1;
            lblNmero.setText("NÚMERO: " + numero_aleatorio);
            Thread hilo = new Thread(this);
            hilo.start();


      }

      public static void main(String[] args)
      {
            ip_servidor = JOptionPane.showInputDialog("Introduce la IP
de tu servidor:");
            new Servidor();
      }

      @Override
      public void actionPerformed(ActionEvent e)
      {
            try
            {
                  llave_bucle_hilo=false;
                  serverSocket.close();


            } catch (IOException e1)
            {
                  e1.printStackTrace();
                  System.out.println("Hola");
            }
            System.exit(0);
      }

      @Override
      public void run()
      {
            try
            {
                  int puerto = 7247;
                  serverSocket = new ServerSocket(puerto);



                  while(llave_bucle_hilo)
                  {
                        Socket socket_recibo = serverSocket.accept();

                        synchronized (this) //LO HACEMOS SYNCRONIZED
PARA QUE NO GANEN DOS JUGADORES A LA VEZ
                        {

                              DataInputStream flujo_entrada = new
DataInputStream(socket_recibo.getInputStream());
                              String mensaje_entrada =
flujo_entrada.readUTF();


                              if
(!comprueba_si_es_ip(mensaje_entrada)==true)
                              {
```

```
                                    String [] mensaje_desagregado=
mensaje_entrada.split("--");

                                    try
                                    {
                                          int numero =
Integer.valueOf(mensaje_desagregado[1]); //OBTENEMOS EL NÚMERO DE LA
CADENA RECIBIDA(SEPARAMOS DEL NOMBRE DE USUARIO)

                                          if (numero ==
numero_aleatorio)
                                          {


                                                socket_recibo.close();
                                                serverSocket.close();

      textarea.append(mensaje_desagregado[0]+ ">>> " + numero +"
***¡¡¡HA ACERTADO!!!***" + "\n");

      scrollpane.getVerticalScrollBar().setValue(scrollpane.getVertica
lScrollBar().getMaximum());
                                                label.setText("--EL
SERVIDOR HA FINALIZADO CORRECTAMENTE--");
                                                System.out.println("--
EL SERVIDOR HA FINALIZADO CORRECTAMENTE--");
                                                textArea2.setText("");
//BORRAMOS TODAS LAS IP CONECTADAS
                                                llave_fin = true;
                                                llave_bucle_hilo =
false;

                                          }
                                          else
                                          {
                                                if (numero <
numero_aleatorio)
                                                {

      textarea.append(mensaje_desagregado[0]+ "> " + numero +"    El
número a acertar es mayor." + "\n");

      scrollpane.getVerticalScrollBar().setValue(scrollpane.getVertica
lScrollBar().getMaximum());
                                                }
                                                else
                                                {

      textarea.append(mensaje_desagregado[0]+ "> " + numero +"    El
número a acertar es menor." + "\n");

      scrollpane.getVerticalScrollBar().setValue(scrollpane.getVertica
lScrollBar().getMaximum());
                                                }
                                          }


                                    } catch (Exception e)
                                    {
```

```
                                        System.out.println("No has
introducido un número válido.");
                                        flujo_entrada.close();
//CERRAMOS LOS FLUJOS DE ENTRADA
                                        socket_recibo.close();
//CERRAMOS LOS FLUJOS DE ENTRADA
                                }

                                flujo_entrada.close(); //CERRAMOS
LOS FLUJOS DE ENTRADA
                                socket_recibo.close(); //CERRAMOS
LOS FLUJOS DE ENTRADA



                                //-------------------------ENVIAMOS
TODO LO QUE HAY EN NUESTRO TEXTAREA PARA DUPLICAL LA INFORMACIÓN DEL
SERVIDOR EN LOS CLIENTES----------------------------


                                for (int i = 0; i <
conexiones.size(); i++)
                                {

                                        Socket socket_envio = new
Socket(conexiones.get(i), 8888);
                                        String mensaje_salida =
textarea.getText().toString();
                                        DataOutputStream flujo_salida
= new DataOutputStream(socket_envio.getOutputStream());

      flujo_salida.writeUTF(mensaje_salida);
                                        flujo_salida.close();
                                        socket_envio.close();
                                }


                                //---------------------------------
--------------------------------------------------------------------
--------------------------------------------------


                                if (llave_fin == true)
                                {
                                        for (int i = 0; i <
conexiones.size(); i++)
                                        {

                                                Socket socket_envio =
new Socket(conexiones.get(i), 8888);
                                                String mensaje_salida =
textarea.getText().toString() + "\n\n---------------------------------
----FIN DEL JUEGO-------------------------------------";
                                                DataOutputStream
flujo_salida = new DataOutputStream(socket_envio.getOutputStream());

      flujo_salida.writeUTF(mensaje_salida);
                                                flujo_salida.close();
                                                socket_envio.close();
                                        }
```

```
                                    }

                                }
                                else //EN CASO DE MANDAR EL CLIENTE LA IP
(SE COMPRUEBA MEDIANTE EL METODO COMPRUEBA_SI_ES_IP) SE ELIMINARÁ
DICHA IP DEL ARRAYLIST, PARA QUE NO HAY PROBLEMAS CUANDO SE DESCONECTE
Y CONECTEN CLIENTES NUEVOS
                                {

                                    if
(conexiones.contains(mensaje_entrada)==true)
                                    {

                                        System.out.println("-
"+mensaje_entrada + " se ha desconectado-");

                                        //ELIMINA LA IP DEL ARRAYLIST
                                        for (int i = 0; i <
conexiones.size(); i++)
                                        {


                                            if
((conexiones.get(i).equals(mensaje_entrada)))
                                            {

      conexiones.remove(i);
                                            }

                                        }


                                        textArea2.setText("");
//LIMPIAMOS TODAS LAS CONEXIONES
                                        actualiza_usuarios();
//RECARGAMOS EL ARRAY LIST CON TODAS LAS CONEXIONES

                                    }
                                    else
                                    {
                                        System.out.println("-
"+mensaje_entrada + " se ha conectado-");

      conexiones.add(mensaje_entrada);//ACUMULA LAS IP DE LAS
CONEXIONES QUE SE VAN AGREGANDO A NUESTRO SERVIDOR

                                        textArea2.setText("");
//LIMPIAMOS TODAS LAS CONEXIONES
                                        actualiza_usuarios();
                                    }
                                }


                                flujo_entrada.close(); //CERRAMOS LOS
FLUJOS DE ENTRADA
                                socket_recibo.close(); //CERRAMOS LOS
FLUJOS DE ENTRADA
                            }
                        }
                    }
```

```
            catch (IOException e)
            {
                  try
                  {
                        llave_bucle_hilo=false;
                        serverSocket.close();
                        System.out.println("FIN DEL PROGRAMA.");

                  } catch (IOException e1)
                  {
                        System.out.println("FIN DEL PROGRAMA.");
                  }
            }
      }
      public boolean comprueba_si_es_ip (String cadena)
      {
            String candena_ = cadena;
            boolean llave = false;
            int contador = 0;

            for (int i = 0; i < candena_.length(); i++)
            {
                  char c = candena_.charAt(i);
                  if (c == '.')
                  {
                        contador ++;
                  }
            }

            if (contador == 3)
            {
                  llave = true;
            }

            return llave;

      }

      public void actualiza_usuarios()
      {

            for (int i = 0; i < conexiones.size(); i++)
            {
                  textArea2.append("···
"+conexiones.get(i).toString()+" ···\n");
            }
      }
}
```

Cliente

```
import java.awt.event.*;
import java.io.*;
import java.net.*;
import javax.swing.*;
import java.awt.Color;
import java.awt.Font;
```

```
public class Cliente extends JFrame implements ActionListener,
Runnable
{
	private static final long serialVersionUID = 1L;
	private JTextField mensaje = new JTextField();
	private JScrollPane scrollpane;
	private JTextArea textarea;
	private JButton boton_enviar = new JButton("Enviar");
	private JButton desconectar = new JButton("Salir");
	private ServerSocket serverSocket;
	private static String nombre;
	private boolean llave = false;
	private static boolean llave_bucle = true;

	private static String ip;
	private final JLabel lblIntroduceUnNmero = new JLabel("INTRODUCE
UN N\u00DAMERO ENTRE 0 Y 1000");

	public Cliente()
	{
		super("Usuario: " + nombre);
		getContentPane().setBackground(Color.DARK_GRAY);
		getContentPane().setLayout(null);
		setSize(550, 400);
		setLocationRelativeTo(null);
		mensaje.setFont(new Font("Arial", Font.BOLD, 13));
		mensaje.setBounds(10, 48, 400, 29);
		getContentPane().add(mensaje);
		textarea = new JTextArea();
		textarea.setFont(new Font("Arial", Font.ITALIC, 13));
		scrollpane = new JScrollPane(textarea);
		scrollpane.setBounds(10, 88, 400, 262);
		getContentPane().add(scrollpane);
		boton_enviar.setBackground(Color.YELLOW);
		boton_enviar.setBounds(420, 48, 100, 30);
		getContentPane().add(boton_enviar);
		desconectar.setBackground(Color.YELLOW);
		desconectar.setBounds(420, 11, 100, 30);
		getContentPane().add(desconectar);
		textarea.setEditable(false);
		boton_enviar.addActionListener(this);
		getRootPane().setDefaultButton(boton_enviar);

	lblIntroduceUnNmero.setHorizontalAlignment(SwingConstants.CENTER
);
		lblIntroduceUnNmero.setForeground(Color.GREEN);
		lblIntroduceUnNmero.setFont(new Font("Arial", Font.BOLD,
13));
		lblIntroduceUnNmero.setBounds(10, 18, 400, 14);

		getContentPane().add(lblIntroduceUnNmero);
		desconectar.addActionListener(this);
		setDefaultCloseOperation(JFrame.DO_NOTHING_ON_CLOSE);
		setVisible(true);

		Thread hilo = new Thread(this);
		hilo.start();

		//NOTIFICA QUE TE HAS CONECTADO CON TU IP
```

```
            try
            {

                  //OBTENEMOS LA IP LOCAL DE LA MÁQUINA
                  InetAddress address = InetAddress.getLocalHost();
                  String ip_local = address.getHostAddress();

                  Socket socket_envio = new Socket(ip, 7247);
                  DataOutputStream flujo_salida = new
DataOutputStream(socket_envio.getOutputStream());
                  flujo_salida.writeUTF(ip_local);
                  flujo_salida.close();
                  socket_envio.close();


            } catch (Exception e)
            {
                  System.out.println(e.getMessage());
                  textarea.append("\n \n ***-----FIN DEL JUEGO-----
***");
                  mensaje.setEditable(false);
                  boton_enviar.setEnabled(false);
            }



      }

      public static void main(String[] args)
      {
            ip = JOptionPane.showInputDialog("Introduce la IP a la que
te quieres conectar:");
            do
            {
                  nombre = JOptionPane.showInputDialog("Introduce tu
nombre o nick:");

            } while (nombre.length() > 7);


            new Cliente().setVisible(true);
      }


      @Override
      public void actionPerformed(ActionEvent ae)
      {
            if(ae.getSource().equals(boton_enviar))
            {
                  try
                  {
                        Socket socket_envio = new Socket(ip, 7247);
                        String texto = mensaje.getText().toString();
                        DataOutputStream flujo_salida = new
DataOutputStream(socket_envio.getOutputStream());
                        flujo_salida.writeUTF(nombre + "--" +texto);
                        flujo_salida.close();
                        socket_envio.close();
                        mensaje.setText("");
                        mensaje.requestFocus();
```

```
                        try
                        {
                              mensaje.setEditable(false);
                              Thread.sleep(3000);
                              mensaje.setEditable(true);
                        } catch (InterruptedException e) {
                              // TODO Auto-generated catch block
                              e.printStackTrace();
                        }

                  } catch (Exception e)
                  {
                        System.out.println(e.getMessage());
                        textarea.append("\n \n ***-----FIN DEL JUEGO---
--***");

      scrollpane.getVerticalScrollBar().setValue(scrollpane.getVertica
lScrollBar().getMaximum());
                        mensaje.setEditable(false);
                        boton_enviar.setEnabled(false);
                  }
            }
            if(ae.getSource().equals(desconectar))
            {
                  try
                  {

                        if (boton_enviar.isEnabled())      //SI EL BOTON
ENVIAR ESTÁ HABILITADO SIGNIFICA QUE NADIE HA GANADO POR LO QUE
ENVIAMOS NUESTRA IP PARA QUE SE ELIMINE DE LA LISTA DEL SERVIDOR, YA
QUE ESTE AÚN SIGUE ESCUCHANDO AL NO HABER FINALIZADO
                        {
                              //OBTENEMOS LA IP LOCAL DE LA MÁQUINA Y
LA ENVIAMOS AL SERVIDOR PARA QUE LA REGISTRE
                              InetAddress address =
InetAddress.getLocalHost();
                              String ip_local =
address.getHostAddress();

                              Socket socket_envio = new Socket(ip,
7247);
                              DataOutputStream flujo_salida = new
DataOutputStream(socket_envio.getOutputStream());
                              flujo_salida.writeUTF(ip_local);
                              flujo_salida.close();
                              socket_envio.close();
                        }


                        llave_bucle = false;
                        System.exit(0);


                  } catch (Exception e)
                  {
                        llave_bucle = false;
                        System.out.println(e.getMessage());
                        textarea.append("\n \n ***-----FIN DEL JUEGO---
--***");
```

```
	scrollpane.getVerticalScrollBar().setValue(scrollpane.getVertica
lScrollBar().getMaximum());
                        mensaje.setEditable(false);
                        boton_enviar.setEnabled(false);
                  }
            }




      }

      @Override
      public void run()
      {
            try
            {
                  int puerto = 8888;
                  serverSocket = new ServerSocket(puerto);

                  while(llave_bucle)
                  {
                        Socket socket_recibo = serverSocket.accept();
                        DataInputStream flujo_entrada = new
DataInputStream(socket_recibo.getInputStream());
                        String mensaje_entrada =
flujo_entrada.readUTF();

                        for (int i = 0; i < mensaje_entrada.length();
i++)
                        {
                              char caracterString = '-';

                              if (caracterString ==
mensaje_entrada.charAt(i))
                              {
                                    llave = true;
                                    break;
                              }
                        }


                        if (llave)
                        {
                              boton_enviar.setEnabled(false);
                              mensaje.setEnabled(false);
                        }
                        else
                        {
                              textarea.setText(mensaje_entrada + "\n");

	scrollpane.getVerticalScrollBar().setValue(scrollpane.getVertica
lScrollBar().getMaximum());
                              flujo_entrada.close();
                              socket_recibo.close();
                        }


                  }
```

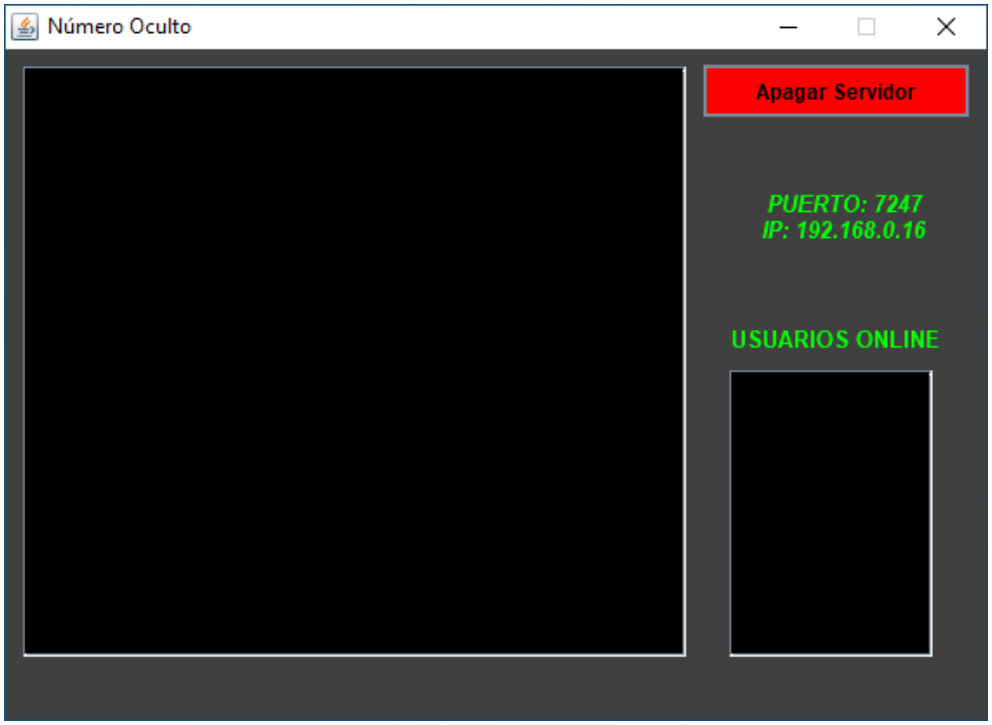
```
            } catch (IOException  e)
            {
                  try
                  {
                        llave_bucle = false;
                        serverSocket.close();

                  } catch (IOException e1)
                  {
                        System.out.println(e1.getMessage());
                  }
            }
      }
                                    }
```

CAPTURAS DE PANTALLA

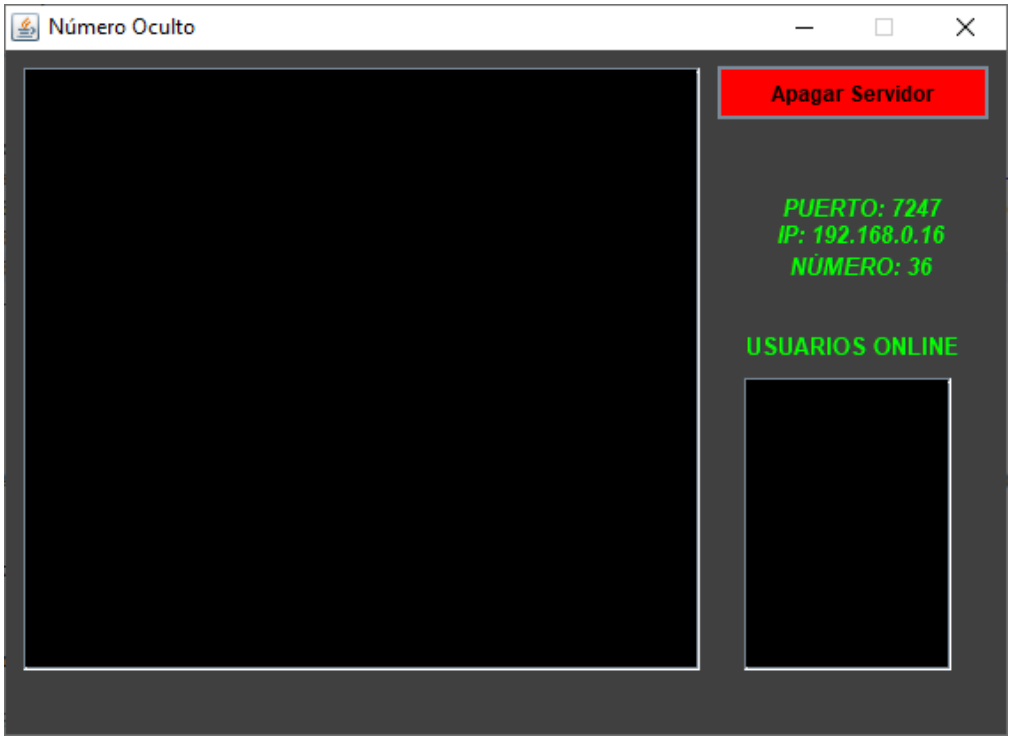
La aplicación servidor escucha por el puerto 7247. En cambio, el cliente actúa como cliente y cómo servidor, ya que cada cliente tiene que recibir las apuestas de los demás clientes a tiempo real, por lo que podemos decir que tiene ambas funciones. Los clientes escuchan la información que les llega del servidor por el puerto 8888.

A continuación, vamos a explica gráficamente la aplicación tanto del servidor como del cliente:

Cuando ejecutamos la aplicación servidor se nos pide una IP, este campo no tendrá ninguna función en nuestro programa ya que la IP dependerá de la máquina donde se ejecute nuestro programa. No obstante, se pide al usuario para que podamos ver los valores en nuestro programa, una vez arrancado.

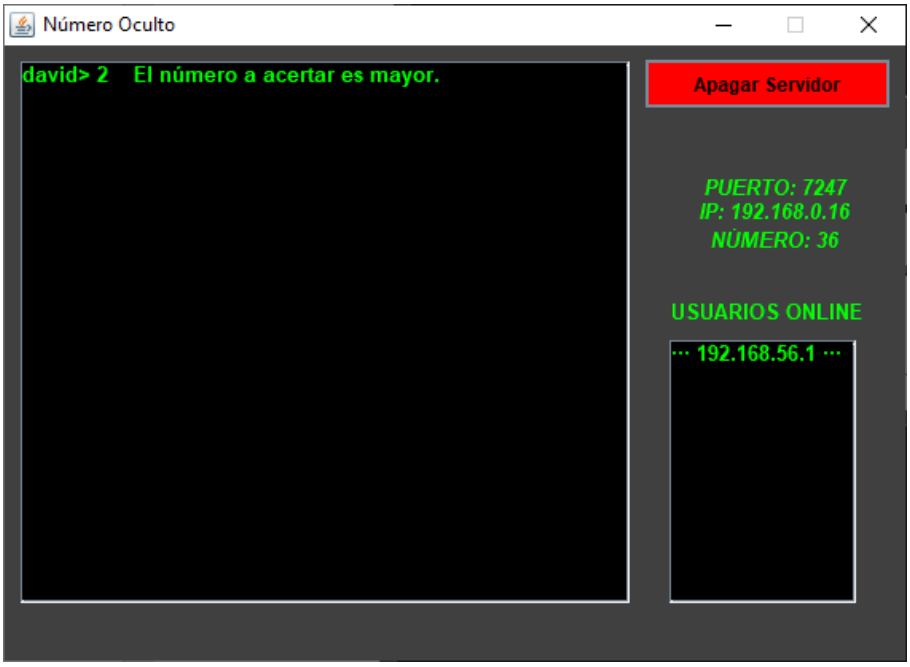


La aplicación servidor tiene un apartado que nos muestra los usuarios online, es decir, las direcciones IP que se conectan y desconectan de nuestro servidor a tiempo real. También podemos ver el puerto por el que escucha y la IP que introducimos al inicio del programa y el número que tenemos que acertar.



La aplicación cliente al ser inciada nos pedirá el número de IP en el que se encuentra nuestro servidor.

En ella podemos ir introduciendo número y el servidor nos responderá si el número que tenemos que acertar es mayor o menor. La información será duplicada en todos los clientes.


Número Oculto
david> 2 El número a acertar es mayor.
Apagar Servidor
PUERTO: 7247
IP: 192.168.0.16
NÚMERO: 36
USUARIOS ONLINE
··· 192.168.56.1 ···