Instituto Politécnico Nacional.

Escuela Superior De Cómputo.

Materia:

Aplicaciones Para Comunicación En Red.

Examen #1: Sopa de Letras

(Reporte)

Profesor:

Axel Ernesto Moreno Cervantes.

Alumno:

Hernández Escobedo Fernando.

Villanueva Guzmán Randy.

Grupo:

3CM5

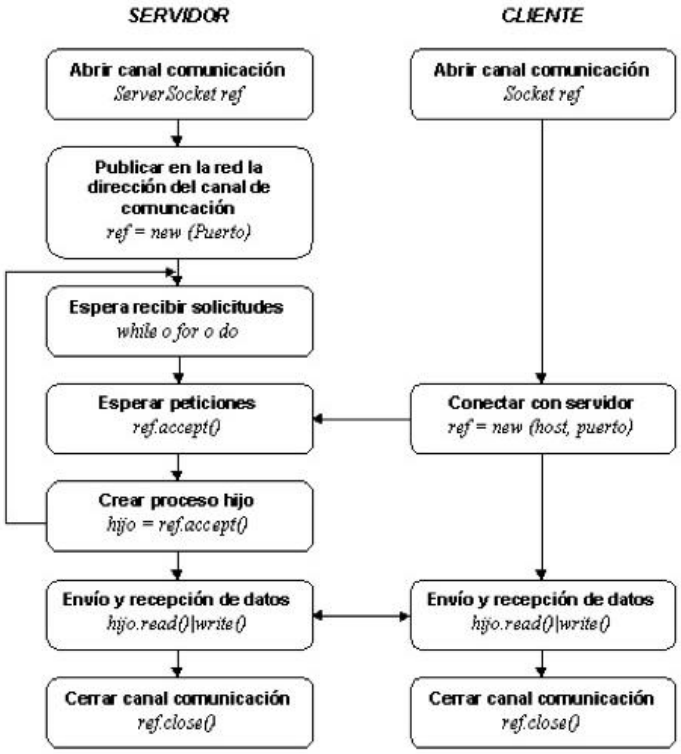
Introducción.

En la actualidad, muchos de los procesos que se ejecutan en una computadora requieren obtener o enviar información a otros procesos que se localizan en una computadora diferente. Para lograr esta comunicación se utilizan los protocolos de comunicación TCP y UDP.

El protocolo TCP (Transmission Control Protocol) establece un conducto de comunicación punto a punto entre dos computadoras, es decir, cuando se requiere la transmisión de un flujo de datos entre dos equipos, el protocolo TCP establece un conducto exclusivo entre dichos equipos por el cual los datos serán transmitidos y este perdurará hasta que la transmisión haya finalizado, gracias a esto TCP garantiza que los datos enviados de un extremo de la conexión lleguen al otro extremo y en el mismo orden en que fueron enviados. Las características que posee TCP hacen que el protocolo sea conocido como un protocolo orientado a conexión.

Los sockets son una forma de comunicación entre procesos que se encuentran en diferentes máquinas de una red, los sockets proporcionan un punto de comunicación por el cual se puede enviar o recibir información entre procesos.

Los sockets tienen un ciclo de vida dependiendo si son sockets de servidor, que esperan a un cliente para establecer una comunicación, o socket cliente que busca a un socket de servidor para establecer la comunicación.

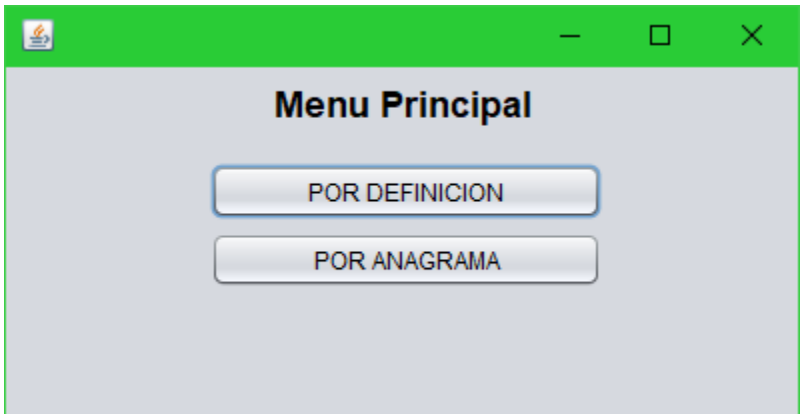


Desarrollo.

El desarrollo de esta práctica-examen consta de

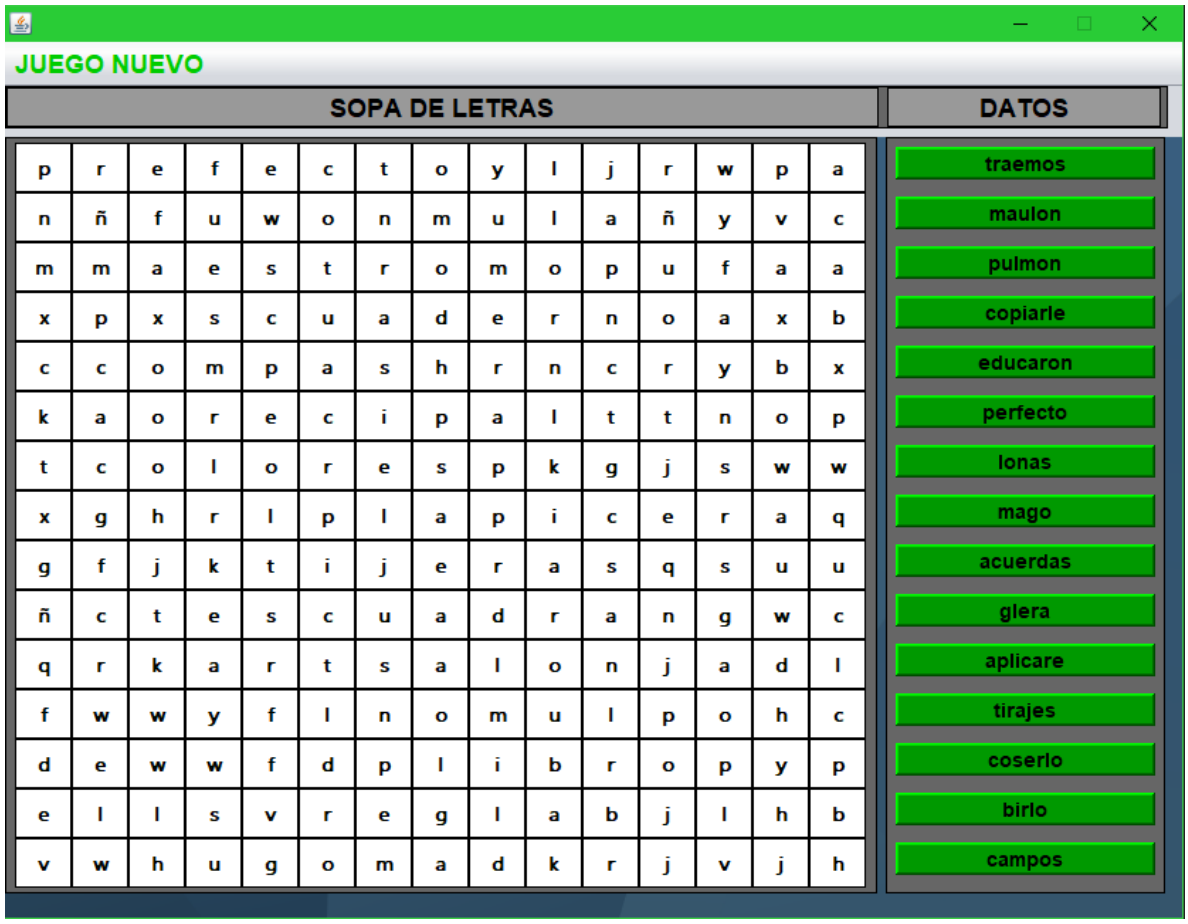
El desarrollo de esta práctica consta de, dos interfaces una de la Sopa de Letras y otra del menú de juego, así como de distintas clases que se encargan de gestionar los envíos y peticiones de información.

Cuando se corre la aplicación (SopaDeLetras) se hace una llamada con un parámetro, 1 o 0, al servidor para que se ponga a escuchar la petición del cliente que recibirá la información que requiere para la aplicación, dependiendo el modo de juego que se escogió será el parámetro 1 o 0 que se le mandará al servidor para que mandé información específica.



Interfaz del Menú de Juego.

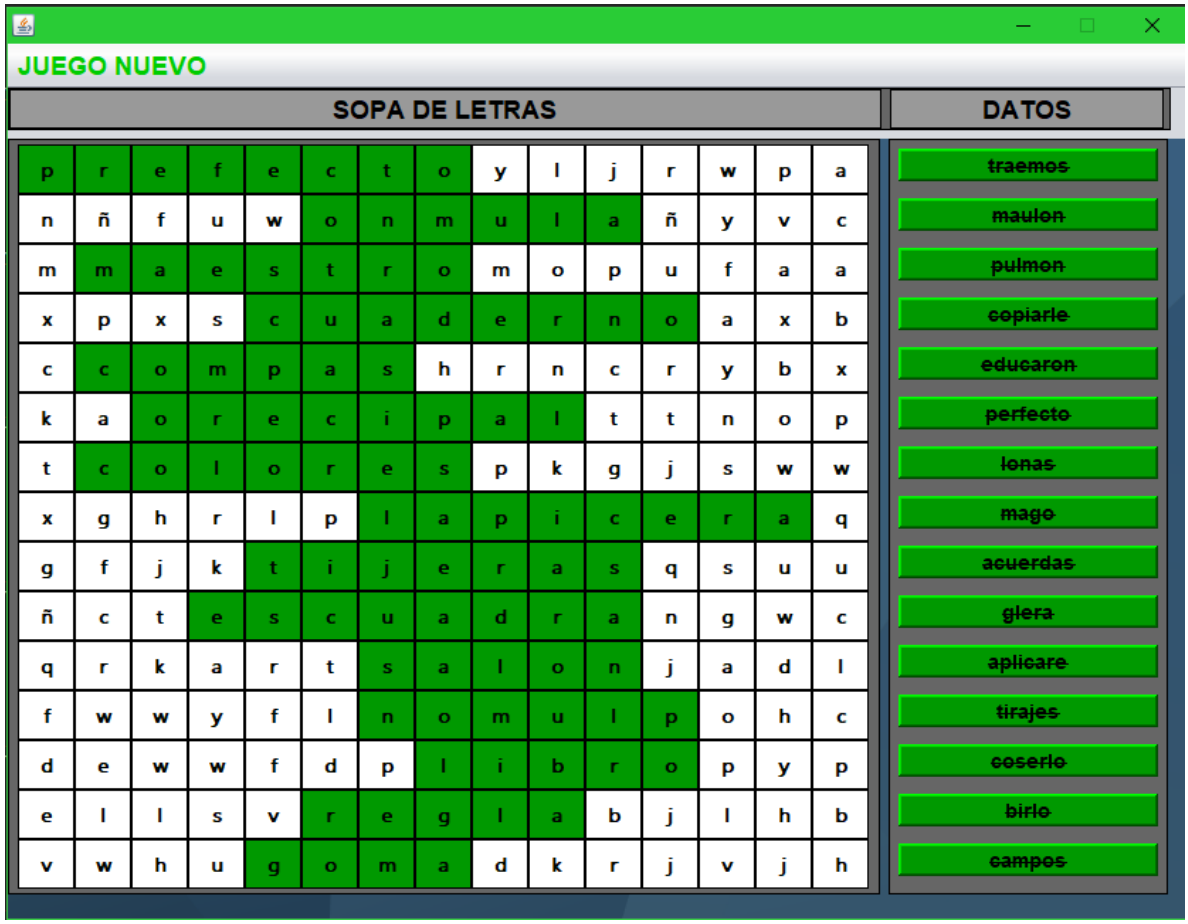
Interfaz de la Sopa de Letras en modo “POR DEFINICIÓN”.



Interfaz de la Sopa de Letras en modo “POR ANAGRAMA”.

Pruebas.

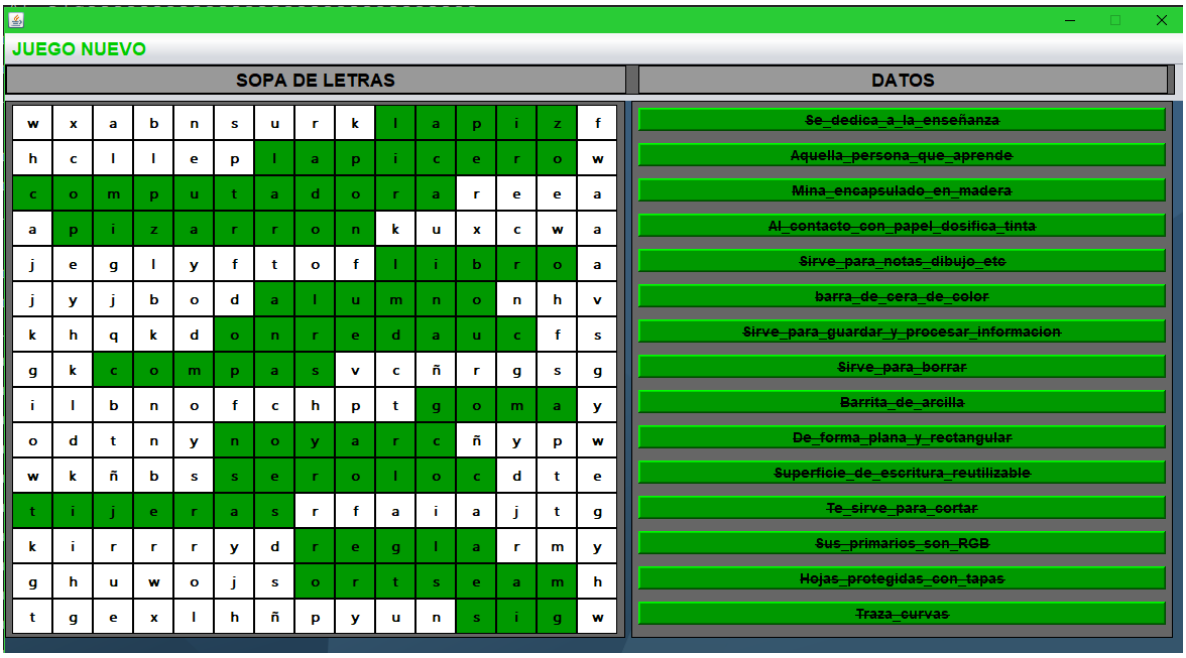
Jugando en modo “POR ANAGRAMA”:



Tablero con las palabras encontradas.

Tiempo que se tardó en resolver el tablero.

Jugando en modo “POR DEFINICIÓN”:



Tablero con las palabras encontradas.

Tiempo que se tardó en resolver el tablero.

Código.

MenuPrincipal.java

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 *
 * @author FernandoHE
 */
public class MenuPrincipal extends javax.swing.JFrame {
```

```
    /**
     * Creates new form MenuPrincipal
     */
    public MenuPrincipal() {
        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() {

        jLabel1 = new javax.swing.JLabel();
        jButton1 = new javax.swing.JButton();
        jButton2 = new javax.swing.JButton();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

        jLabel1.setFont(new java.awt.Font("Dialog", 1, 18)); // NOI18N
        jLabel1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        jLabel1.setText("Menu Principal");

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

        jButton2.setText("POR ANAGRAMA");
        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()
                .addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(layout.createSequentialGroup()
                        .addComponent(jLabel1,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
                        .addContainerGap())
                    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
                        .addGap(0, 96, Short.MAX_VALUE)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING,
false)
```

```
                            .addComponent(jButton2,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
                            .addComponent(jButton1,
javax.swing.GroupLayout.DEFAULT_SIZE, 196, Short.MAX_VALUE))
                        .addGap(98, 98, 98))))
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jLabel1)
                .addGap(18, 18, 18)
                .addComponent(jButton1)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jButton2)
                .addContainerGap(191, Short.MAX_VALUE))
        );

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

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        Cliente1 cl = new Cliente1();
        cl.cliente();
        this.setVisible(false);
    }

    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        Cliente2 c2 = new Cliente2();
        c2.cliente();
        this.setVisible(false);
    }

    /**
     * @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(MenuPrincipal.class.getName()).log(java.util.l
ogging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(MenuPrincipal.class.getName()).log(java.util.l
ogging.Level.SEVERE, null, ex);
```

```
        } catch (IllegalAccessException ex) {

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

java.util.logging.Logger.getLogger(MenuPrincipal.class.getName()).log(java.util.l
ogging.Level.SEVERE, null, ex);
        }
        //</editor-fold>
        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new MenuPrincipal().setVisible(true);
            }
        });
        if(Boolean.TRUE){
            //
        }
        Servidor1 s1 = new Servidor1();
        //Servidor2 s2 = new Servidor2();
        s1.servidor();
        //s2.servidor();

    }

    // Variables declaration - do not modify
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton2;
    private javax.swing.JLabel jLabel1;
    // End of variables declaration
}
```

Servidor1.java

```
import java.net.*;
import java.io.*;
/**
 *
 * @author FERNANDO
 */
public class Servidor1 {

    public void servidor(){
        try{
            int pto = 5678;
            ServerSocket s = new ServerSocket(pto);
            s.setReuseAddress( true );
            System.out.println("Servidor iniciado en el puerto " + pto + "
Esperando archivo");
            for(;;){
                Socket cl = s.accept();
                ObjectOutputStream oos = new ObjectOutputStream(
cl.getOutputStream() );
                DataInputStream dis = new DataInputStream(cl.getInputStream());
                int op = dis.readInt();
                switch(op){
                    case 1:
                                String[][] palabras1 = { {"maestro", "alumno",
"lapiz", "lapicero", "cuaderno", "crayon", "computadora", "goma",
                "gis", "regla", "compas", "libro", "colores", "tijeras",
"pizarron"},
```

```
                                {"Se_dedica_a_la_enseñanza",
"Aquella_persona_que_aprende", "Mina_encapsulado_en_madera",
                                "Al_contacto_con_papel_dosifica_tinta",
"Sirve_para_notas_dibujo_etc", "barra_de_cera_de_color",
                                "Sirve_para_guardar_y_procesar_informacion",
"Sirve_para_borrar", "Barrita_de_arcilla",
                                "De_forma_plana_y_rectangular", "Traza_curvas",
"Hojas_protegidas_con_tapas", "Sus_primarios_son_RGB",
                                "Te_sirve_para_cortar",
"Superficie_de_escritura_reutilizable"}};
                            oos.writeObject(palabras1);

                            break;
                    case 2:
                            String [][] palabras2 = {
                            {"maestro", "alumno", "plumon", "lapicero",
"cuaderno", "prefecto", "salon", "goma","escuadra", "regla", "compas", "libro",
"colores", "tijeras", "lapicera"},
                            {"traemos", "maulon", "pulmon","copiarle",
"educaron", "perfecto", "lonas", "mago", "acuerdas", "glera", "campos", "birlo",
"coserlo", "tirajes", "aplicare"}};
                            oos.writeObject(palabras2);
                            break;
                }
                oos.flush();
                oos.close();
                cl.close();
            }
        }catch(Exception e){
                e.printStackTrace();
        }
    }
}
```

Sopa1.java

```
import java.awt.Color;
import java.util.Random;
import javax.swing.JInternalFrame;
import javax.swing.JLabel;
/**
 *
 * @author FernandoHE
 */
public class Sopa1 extends javax.swing.JFrame {
    String palabrasS[][];
    Final fin;
    long tiempo, tFinal;
    JLabel letra[][];//Letras en la sopa
    JLabel palabra[];//Palabras a la derecha
    String palabras[];//las palabras en un arreglo de string
    int iniciox[];//Coordenada X de donde inicia la palabra
    int inicioy[];//Coordenada Y de donde inicia la palabra
    boolean gano;
    boolean direccion[];//TRUE = Derecha  FALSE = Izquierda
    public Sopa1(String[][]pals) {
        palabrasS = pals;
        initComponents();
        palabra = new JLabel[]{p1, p2, p3, p4, p5, p6, p7, p8, p9, p10, p11, p12,
p13, p14, p15};
        for (int i = 0; i < 15; i++)
            palabra[i].setText(palabrasS[1][i]);
        this.setLocationRelativeTo(null);
```

```
        cargar();
        palabras = new String[15];
        for (int i = 0; i < 15; i++) {
            palabras[i] = palabrasS[0][i];//pasa la palabra del arreglo label al
al arreglo de string
        }
    }//Sopa


    private void jMenu1MouseClicked(java.awt.event.MouseEvent evt) {//GEN-
FIRST:event_jMenu1MouseClicked
        //esta reinicia el juego
        for (int i = 0; i < letra.length; i++) {
            palabra[i].setText(palabrasS[1][i]);//asigna a los label de la
derecha las palabras
        }
        for (int i = 0; i < 15; i++) {
            for (int j = 0; j < 15; j++) {
                Sopa_de_letra.remove(letra[i][j]);//quita el panel  Sopa_de_letra
            }
        }
        cargar();//carga el juego
    }//GEN-LAST:event_jMenu1MouseClicked
    public void cargar() {
        gano = false;
        iniciox = new int[15];//crea un arreglo de enteros para guadar las
posiciones de las palabras en x
        inicioy = new int[15];//crea un arreglo de enteros para guadar las
posiciones de las palabras en y
        direccion = new boolean[15];//crea un arreglo de enteros para guadar las
direcion de las palabras ya sea hacia la derecha o la izquierda
        celdasDeLetras();
        colocarPalabras();
        llenarEspaciosVacios();
    }//Cargar
    public void celdasDeLetras() {
        letra = new JLabel[15][15];//crea la matriz de celdas donde va cada letra
        for (int i = 0; i < 15; i++) {
            for (int j = 0; j < 15; j++) {
                letra[i][j] = new JLabel("",
javax.swing.SwingConstants.CENTER);//crea la casilla la vacia y con una
alineacion centrada
                letra[i][j].setName("");//le pone un nombre a la casilla en este
caso no le pone ninguno
                letra[i][j].setBackground(Color.WHITE);//coloca la casilla de
color blanco
                letra[i][j].setFont(new java.awt.Font("Segoe UI Symbol", 1, 14));
// asigna el tipo y el tamaño de la letra
                letra[i][j].setForeground(new java.awt.Color(0, 5, 2));

letra[i][j].setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.bor
der.BevelBorder.RAISED));
                letra[i][j].setOpaque(true);//esto es para que se pueda ver el
color de la casilla o cajonsito donde va la letra
                letra[i][j].setBorder(new
javax.swing.border.LineBorder(Color.BLACK, 1));//pone a la casiilla en borde con
una linea negra
                letra[i][j].addMouseListener(new java.awt.event.MouseAdapter()
{//pone a la casilla a la escucha del mouse para saber cuando se esta dando clic
                    @Override
                    public void mouseClicked(java.awt.event.MouseEvent evt) {
                        presioneClic(evt);//llama al metodo que debe ejecutarse
cuando se da clic
```

```
                    }
                });
                Sopa_de_letra.add(letra[i][j]);//coloca la casilla en el panel
Sopa_de_letra
            }
        }
    }//celdasDeLetras
    //este metodo se ejecuta cuando se presiona clic en una casilla
    public void presioneClic(java.awt.event.MouseEvent evt) {
        if (!gano) {//verifica si gano el juego
            if (evt.getComponent().getBackground().equals(Color.WHITE))//verifica
si la casilla esta de color blanco
            {
                evt.getComponent().setBackground(new java.awt.Color(0, 153,
0));//si esta de color blanco la pone de color verde
                tachar();
            }else if(evt.getComponent().getName().equals(""))//pregunta si la
casilla no tiene una letra de alguna palabra
            {
                evt.getComponent().setBackground(Color.WHITE);//pone la casilla
de color blanco
            }
        }
    }//presioneClic
    public void tachar() {
        for (int i = 0; i < 15; i++) {
            if (!palabra[i].getText().substring(0, 1).equals("<")) {
                //String auxString = palabras[i];
                //int tam = auxString.length();
                if
(tacharLetra(iniciox[i],inicioy[i],palabras[i].length(),direccion[i]))//pregunta
si hay una palabra encontrada
                {
                    System.out.println("Soy la del lado derecho" + i +
palabra[i].getText());

palabra[i].setText("<html><body><s>"+palabra[i].getText()+"</s></body></html>");/
/tacha la palabra
                    break;
                }
            }
        }
        boolean aux = true;//ayuda para saber si ya todas las palabras estan
tachadas
        for (int i = 0; i < letra.length; i++)
        {
            if (!palabra[i].getText().substring(0, 1).equals("<"))
            {
                aux = false;
                break;
            }
        }
        if (aux) {
            if (!(fin instanceof Final)) { //esto comprueba si la ventana no esta
en memoria, entonces la instancia
                tFinal = System.currentTimeMillis();
                double tiempoJuego = (double) ((tFinal - tiempo)/1000);
                System.out.println("Tiempo en segundos = " + tiempoJuego);
                fin = new Final(tiempoJuego);
                gano = true;
            }
            CentrarVentanaInterna(fin); //usamos el metodo generico para centrar
        }
```

```
    }//tachar
    public void CentrarVentanaInterna(JInternalFrame internalFrame) {
    //pasamos como parametro un objeto de tipo JinternalFrame
        int x = (escritorio.getWidth() / 2) - internalFrame.getWidth() / 2;
//caculas las posiciones x y y
        int y = (escritorio.getHeight() / 2) - internalFrame.getHeight() / 2;
        if (internalFrame.isShowing()) {// comprobamos si la ventana ya esta
ejecutada
            internalFrame.setLocation(x, y); // si es asi solo le colocamos en la
mitad
        }
        else
        {
            escritorio.add(internalFrame); // si no es asi le insertamos dentro
del desktoppane
            internalFrame.setLocation(x, y);
            internalFrame.show(); // y mostramos
        }
    }//CentrarVentanaInterna
//Verifica si se puede tachar la letra
    public boolean  tacharLetra(int x,int y,int tamanio,boolean direccion) {
        boolean respuesta = true;
        if (direccion) {//Si la palabra se lee de izquierda a derecha
            for (int i = y; i < tamanio+y; i++) {
                if (letra[x][i].getBackground().equals(Color.WHITE)) {//Si aun
tiene casillas blancas
                    respuesta = false;
                    break;
                }
            }
        }else {
            for (int j = y; j > y-tamanio; j--) {
                if (letra[x][j].getBackground().equals(Color.WHITE)) {
                    respuesta = false;
                    break;
                }
            }
        }
        return respuesta;//Retorna TRUE si ya se seleccionaron las letras que
conforman la palabra, FALSE de lo ocntrario
    }//tacharLetra
    public void colocarPalabras() {
        String palabra[] =
{"","","","","","","","","","","","","","",""};//palabra = palabrasS;
        for (int i = 0; i < 15; i++) {
            palabra[i] = palabrasS[0][i];//pasa la palabra del arreglo label al
al arreglo de string
        }

        Random random = new Random();//estemetodo ayuda a crear numeros
aleatorios
        int iniciax = 0;//posicion x donde inicia la palabra
        int iniciay;//posicion y donde inicia la palabra
        int unico[] = NumerosSinRepeticiones(15);//evita que en una fila se
generen mas de una vez
        for (int i = 0; i < 15; i++) {
            System.out.println("Soy unico[" + i + "] " + unico[i]);
        }
        iniciox = unico;//Contiene el orden en que las FILAS se llenaran
        for (int i = 0; i < palabra.length; i++) {
            if (palabra[i].length()<15) {
                iniciax = unico[i];
                iniciay = (int)(random.nextDouble()*15-1);
```

```
                int estrae = 0;//ayuda para estraer las letras de la palabra
                if (iniciay+palabra[i].length()<15) {
                    System.out.println("ENTRE AL if");
                    for (int j = iniciay; j < iniciay+palabra[i].length(); j++) {
                        letra[iniciax][j].setText(palabra[i].substring(estrae,
estrae+1));//estrae una letra de la palabra
                        letra[iniciax][j].setName("1");//pone el nombre a la
casilla para que se sepa que hay va una letra de una palabra
                        estrae++;//esto es para que se estraiga la siguiente
letra de la palabra
                        inicioy[i] = iniciay;
                        direccion[i] = true;
                    }
                } else if (iniciay-palabra[i].length()>0) {
                    System.out.println("ENTRE AL
elseeeeeeeeeeeeeeeeeeeeeeeeeeeeeee");
                    for (int j = iniciay; j >iniciay-palabra[i].length() ; j--) {
                        letra[iniciax][j].setText(palabra[i].substring(estrae,
estrae+1));
                        letra[iniciax][j].setName("1");
                        estrae++;
                        inicioy[i] = iniciay;
                        direccion[i] = false;
                    }
                }
                else {//Ponerla desde la primer columna
                    iniciay = 0;
                    for (int j = iniciay; j < iniciay+palabra[i].length(); j++) {
                        letra[iniciax][j].setText(palabra[i].substring(estrae,
estrae+1));//estrae una letra de la palabra
                        letra[iniciax][j].setName("1");//pone el nombre a la
casilla para que se sepa que hay va una letra de una palabra
                        estrae++;//esto es para que se estraiga la siguiente
letra de la palabra
                        inicioy[i] = iniciay;
                        direccion[i] = true;
                    }
                }
            }
            System.out.println("Soy iniciox[" + i + "] " + iniciox[i] + "\n" +
"Soy inicioy[" + i + "] " + inicioy[i]);
            System.out.println("Soy direccion[" + i + "] " + direccion[i]);
        }
    }//colocarPalabras
    public int[] NumerosSinRepeticiones(int repeticiones) {
        int numeros[] = new int[repeticiones];
        for (int i = 0; i < repeticiones; i++) {
            numeros[i] = -1;
            System.out.println("Soy numeros[" + i + "] " + numeros[i]);
        }
        Random random = new Random();
        boolean aux ;//informa si la fila esta o no repetida
        int numero = 0;
        for (int x = 0; x < repeticiones; x++)
        {
            aux = true;
            while (aux) {
                aux = false;
                numero = (int)(random.nextDouble()*16-1);
                for (int j = 0; j < numeros.length; j++) {
                    if (numeros[j] == numero) {
                        aux = true;
                        break;
```

```
                    }
                }
            }
            numeros[x] = numero;
            System.out.println("Soy numeros[" + x + "] " + numeros[x]);
        }
        return numeros;//Retorna arreglo con el orden en el que se pondran las
palabras
    }//NumerosSinRepeticiones
    public void llenarEspaciosVacios() {
        //este arreglo ayuda a poner las letras del avecedario
        //String abc[] =
{"A","B","C","D","E","F","G","H","I","J","K","L","M","N","Ñ","O","P","Q","R","S",
"T","U","V","W","X","Y","Z"};
        String abc[] =
{"a","b","c","d","e","f","g","h","i","j","k","l","m","n","ñ","o","p","q","r","s",
"t","u","v","w","x","y","z"};
        Random random = new Random();
        for (int i = 0; i < 15; i++) {
            for (int j = 0; j < 15; j++) {
                if (letra[i][j].getText().equals("")) {//si la casilla esta vacia
pongale una letra del arreglo abc
                    letra[i][j].setText(abc[(int)(random.nextDouble()*abc.length-
1)]);//aqui pone la letra
                }
            }
        }
        tiempo = System.currentTimeMillis();
    }//LlenarEspaciosVacios
    /**
     * @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(Sopa1.class.getName()).log(java.util.logging.L
evel.SEVERE, null, ex);
        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Sopa1.class.getName()).log(java.util.logging.L
evel.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Sopa1.class.getName()).log(java.util.logging.L
evel.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Sopa1.class.getName()).log(java.util.logging.L
evel.SEVERE, null, ex);
        }

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable()
        {
```

```
            public void run() {
                //new Sopa().setVisible(true);
            }
        });
    }//main

    // Variables declaration - do not modify//GEN-BEGIN:variables
    javax.swing.JPanel Sopa_de_letra;
    javax.swing.JDesktopPane escritorio;
    javax.swing.JLabel jLabel1;
    javax.swing.JLabel jLabel2;
    javax.swing.JMenu jMenu1;
    javax.swing.JMenuBar jMenuBar1;
    javax.swing.JPanel jPanel2;
    javax.swing.JPanel jPanel3;
    javax.swing.JLabel p1;
    javax.swing.JLabel p10;
    javax.swing.JLabel p11;
    javax.swing.JLabel p12;
    javax.swing.JLabel p13;
    javax.swing.JLabel p14;
    javax.swing.JLabel p15;
    javax.swing.JLabel p2;
    javax.swing.JLabel p3;
    javax.swing.JLabel p4;
    javax.swing.JLabel p5;
    javax.swing.JLabel p6;
    javax.swing.JLabel p7;
    javax.swing.JLabel p8;
    javax.swing.JLabel p9;
    // End of variables declaration//GEN-END:variables
}
```

Sopa2.java

```
import java.awt.Color;
import java.util.Random;
import javax.swing.JInternalFrame;
import javax.swing.JLabel;

/**
 *
 * @author FernandoHE
 */
public class Sopa2 extends javax.swing.JFrame {
    /**
     * Creates new form Sopa
     */
    String palabrasS[][];
    Final fin;
    long tiempo, tFinal;
    JLabel letra[][];//Letras en la sopa
    JLabel palabra[];//Palabras a la derecha
    String palabras[];//las palabras en un arreglo de string
    int iniciox[];//Coordenada X de donde inicia la palabra
    int inicioy[];//Coordenada Y de donde inicia la palabra
    boolean gano;
    boolean direccion[];//TRUE = Derecha  FALSE = Izquierda
    public Sopa2(String[][]pals) {
        palabrasS = pals;
        initComponents();
        palabra = new JLabel[]{p1, p2, p3, p4, p5, p6, p7, p8, p9, p10, p11, p12,
p13, p14, p15};
```

```
        for (int i = 0; i < 15; i++)
            palabra[i].setText(palabrasS[1][i]);
        this.setLocationRelativeTo(null);
        cargar();
        palabras = new String[15];
        for (int i = 0; i < 15; i++) {
            palabras[i] = palabrasS[0][i];//pasa la palabra del arreglo label al
al arreglo de string
        }
    }//Sopa


    private void jMenu1MouseClicked(java.awt.event.MouseEvent evt) {//GEN-
FIRST:event_jMenu1MouseClicked
        //esta reinicia el juego
        for (int i = 0; i < letra.length; i++) {
            palabra[i].setText(palabrasS[1][i]);//asigna a los label de la
derecha las palabras
        }
        for (int i = 0; i < 15; i++) {
            for (int j = 0; j < 15; j++) {
                Sopa_de_letra.remove(letra[i][j]);//quita el panel  Sopa_de_letra
            }
        }
        cargar();//carga el juego
    }//GEN-LAST:event_jMenu1MouseClicked
    public void cargar() {
        gano = false;
        iniciox = new int[15];//crea un arreglo de enteros para guadar las
posiciones de las palabras en x
        inicioy = new int[15];//crea un arreglo de enteros para guadar las
posiciones de las palabras en y
        direccion = new boolean[15];//crea un arreglo de enteros para guadar las
direcion de las palabras ya sea hacia alante o hacia tras
        celdasDeLetras();
        colocarPalabras();
        llenarEspaciosVacios();
    }//Cargar
    public void celdasDeLetras() {
        letra = new JLabel[15][15];//crea la matriz de celdas donde va cada letra
        for (int i = 0; i < 15; i++) {
            for (int j = 0; j < 15; j++) {
                letra[i][j] = new JLabel("",
javax.swing.SwingConstants.CENTER);//crea la casilla la vacia y con una
alineacion centrada
                letra[i][j].setName("");//le pone un nombre a la casilla en este
caso no le pone ninguno
                letra[i][j].setBackground(Color.WHITE);//coloca la casilla de
color blanco
                letra[i][j].setFont(new java.awt.Font("Segoe UI Symbol", 1, 14));
// asigna el tipo y el tamaño de la letra
                letra[i][j].setForeground(new java.awt.Color(0, 5, 2));

letra[i][j].setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.bor
der.BevelBorder.RAISED));
                letra[i][j].setOpaque(true);//esto es para que se pueda ver el
color de la casilla o cajonsito donde va la letra
                letra[i][j].setBorder(new
javax.swing.border.LineBorder(Color.BLACK, 1));//pone a la casiilla en borde con
una linea negra
                letra[i][j].addMouseListener(new java.awt.event.MouseAdapter()
{//pone a la casilla a la escucha del mouse para saber cuando se esta dando clic
                    @Override
```

```
                    public void mouseClicked(java.awt.event.MouseEvent evt) {
                        presioneClic(evt);//llama al metodo que debe ejecutarse
cuando se da clic
                    }
                });
                Sopa_de_letra.add(letra[i][j]);//coloca la casilla en el panel
Sopa_de_letra
            }
        }
    }//celdasDeLetras
    //este metodo se ejecuta cuando se presiona clic en una casilla
    public void presioneClic(java.awt.event.MouseEvent evt) {
        if (!gano) {//verifica si gano el juego
            if (evt.getComponent().getBackground().equals(Color.WHITE))//verifica
si la casilla esta de color blanco
            {
                evt.getComponent().setBackground(new java.awt.Color(0, 153,
0));//si esta de color blanco la pone de color azul
                tachar();
            }else if(evt.getComponent().getName().equals(""))//pregunta si la
casilla no tiene una letra de alguna palabra
            {
                evt.getComponent().setBackground(Color.WHITE);//pone la casilla
de color blanco
            }
        }
    }//presioneClic
    public void tachar() {
        for (int i = 0; i < 15; i++) {
            if (!palabra[i].getText().substring(0, 1).equals("<")) {
                if
(tacharLetra(iniciox[i],inicioy[i],palabra[i].getText().length(),direccion[i]))//
pregunta si hay una palabra encontrada
                {

palabra[i].setText("<html><body><s>"+palabra[i].getText()+"</s></body></html>");/
/tacha la palabra
                    break;
                }
            }
        }
        boolean aux = true;//ayuda para saber si ya todas las palabras estan
tachadas
        for (int i = 0; i < letra.length; i++)
        {
            if (!palabra[i].getText().substring(0, 1).equals("<"))
            {
                aux = false;
                break;
            }
        }
        if (aux) {
            if (!(fin instanceof Final))
            { //esto comprueba si la ventana no esta en memoria, entonces la
instancia
                tFinal = System.currentTimeMillis();
                double tiempoJuego = (double) ((tFinal - tiempo)/1000);
                System.out.println("Tiempo en segundos = " + tiempoJuego);
                fin = new Final(tiempoJuego);
                gano = true;
            }
            CentrarVentanaInterna(fin); //usamos el metodo generico para centrar
        }
```

```
    }//tachar
    public void CentrarVentanaInterna(JInternalFrame internalFrame) {
    //pasamos como parametro un objeto de tipo JinternalFrame
        int x = (escritorio.getWidth() / 2) - internalFrame.getWidth() / 2;
//caculas las posiciones x y y
        int y = (escritorio.getHeight() / 2) - internalFrame.getHeight() / 2;
        if (internalFrame.isShowing()) {// comprobamos si la ventana ya esta
ejecutada
            internalFrame.setLocation(x, y); // si es asi solo le colocamos en la
mitad
        }
        else
        {
            escritorio.add(internalFrame); // si no es asi le insertamos dentro
del desktoppane
            internalFrame.setLocation(x, y);
            internalFrame.show(); // y mostramos
        }
    }//CentrarVentanaInterna
//Verifica si se puede tachar la letra
    public boolean  tacharLetra(int x,int y,int tamaño,boolean direccion) {
        boolean respuesta = true;
        if (direccion) {
            for (int i = y; i < tamaño+y; i++) {
                if (letra[x][i].getBackground().equals(Color.WHITE)) {
                    respuesta = false;
                    break;
                }
            }
        }else {
            for (int j = y; j > y-tamaño; j--) {
                if (letra[x][j].getBackground().equals(Color.WHITE)) {
                    respuesta = false;
                    break;
                }
            }
        }
        return respuesta;
    }//tacharLetra
    public void colocarPalabras() {
        String palabra[] =
{"","","","","","","","","","","","","","",""};//palabra = palabrasS;
        for (int i = 0; i < 15; i++) {
            palabra[i] = palabrasS[0][i];//pasa la palabra del arreglo label al
al arreglo de string
        }

        Random random = new Random();//estemetodo ayuda a crear numeros
aleatorios
        int iniciax = 0;//posicion x donde inicia la palabra
        int iniciay;//posicion y donde inicia la palabra
        int unico[] = NumerosSinRepeticiones(15);//evita que en una fila se
generen mas de una vez
        for (int i = 0; i < 15; i++) {
            System.out.println("Soy unico[" + i + "] " + unico[i]);
        }
        iniciox = unico;//Contiene el orden en que las FILAS se llenaran
        for (int i = 0; i < palabra.length; i++) {
            if (palabra[i].length()<15) {
                iniciax = unico[i];
                iniciay = (int)(random.nextDouble()*15-1);
                int estrae = 0;//ayuda para estraer las letras de la palabra
                if (iniciay+palabra[i].length()<15) {
```

```
                    System.out.println("ENTRE AL if");
                    for (int j = iniciay; j < iniciay+palabra[i].length(); j++) {
                        letra[iniciax][j].setText(palabra[i].substring(estrae,
estrae+1));//estrae una letra de la palabra
                        letra[iniciax][j].setName("1");//pone el nombre a la
casilla para que se sepa que hay va una letra de una palabra
                        estrae++;//esto es para que se estraiga la siguiente
letra de la palabra
                        inicioy[i] = iniciay;
                        direccion[i] = true;
                    }
                } else if (iniciay-palabra[i].length()>0) {
                    System.out.println("ENTRE AL
elseeeeeeeeeeeeeeeeeeeeeeeeeeeeeee");
                    for (int j = iniciay; j >iniciay-palabra[i].length() ; j--) {
                        letra[iniciax][j].setText(palabra[i].substring(estrae,
estrae+1));
                        letra[iniciax][j].setName("1");
                        estrae++;
                        inicioy[i] = iniciay;
                        direccion[i] = false;
                    }
                }
                else {//Ponerla desde la primer columna
                    iniciay = 0;
                    for (int j = iniciay; j < iniciay+palabra[i].length(); j++) {
                        letra[iniciax][j].setText(palabra[i].substring(estrae,
estrae+1));//estrae una letra de la palabra
                        letra[iniciax][j].setName("1");//pone el nombre a la
casilla para que se sepa que hay va una letra de una palabra
                        estrae++;//esto es para que se estraiga la siguiente
letra de la palabra
                        inicioy[i] = iniciay;
                        direccion[i] = true;
                    }
                }
            }
            System.out.println("Soy iniciox[" + i + "] " + iniciox[i] + "\n" +
"Soy inicioy[" + i + "] " + inicioy[i]);
            System.out.println("Soy direccion[" + i + "] " + direccion[i]);
        }
    }//colocarPalabras
    public int[] NumerosSinRepeticiones(int repeticiones) {
        int numeros[] = new int[repeticiones];
        for (int i = 0; i < repeticiones; i++) {
            numeros[i] = -1;
            System.out.println("Soy numeros[" + i + "] " + numeros[i]);
        }
        Random random = new Random();
        boolean aux ;//informa si la fila esta o no repetida
        int numero = 0;
        for (int x = 0; x < repeticiones; x++)
        {
            aux = true;
            while (aux) {
                aux = false;
                numero = (int)(random.nextDouble()*16-1);
                for (int j = 0; j < numeros.length; j++) {
                    if (numeros[j] == numero) {
                        aux = true;
                        break;
                    }
                }
```

```
            }
            numeros[x] = numero;
            System.out.println("Soy numeros[" + x + "] " + numeros[x]);
        }
        return numeros;
    }//NumerosSinRepeticiones
    public void llenarEspaciosVacios() {
        //este arreglo ayuda a poner las letras del avecedario
        //String abc[] =
{"A","B","C","D","E","F","G","H","I","J","K","L","M","N","Ñ","O","P","Q","R","S",
"T","U","V","W","X","Y","Z"};
        String abc[] =
{"a","b","c","d","e","f","g","h","i","j","k","l","m","n","ñ","o","p","q","r","s",
"t","u","v","w","x","y","z"};
        Random random = new Random();
        for (int i = 0; i < 15; i++) {
            for (int j = 0; j < 15; j++) {
                if (letra[i][j].getText().equals("")) {//si la casilla esta vacia
pongale una letra del arreglo abc
                    letra[i][j].setText(abc[(int)(random.nextDouble()*abc.length-
1)]);//aqui pone la letra
                }
            }
        }
        tiempo = System.currentTimeMillis();
    }//LlenarEspaciosVacios
    /**
     * @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(Sopa2.class.getName()).log(java.util.logging.L
evel.SEVERE, null, ex);
        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Sopa2.class.getName()).log(java.util.logging.L
evel.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Sopa2.class.getName()).log(java.util.logging.L
evel.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Sopa2.class.getName()).log(java.util.logging.L
evel.SEVERE, null, ex);
        }

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable()
        {
            public void run() {
                //new Sopa().setVisible(true);
            }
```

```
        });
    }//main

    // Variables declaration - do not modify//GEN-BEGIN:variables
    javax.swing.JPanel Sopa_de_letra;
    javax.swing.JDesktopPane escritorio;
    javax.swing.JLabel jLabel1;
    javax.swing.JLabel jLabel2;
    javax.swing.JMenu jMenu1;
    javax.swing.JMenuBar jMenuBar1;
    javax.swing.JPanel jPanel2;
    javax.swing.JPanel jPanel3;
    javax.swing.JLabel p1;
    javax.swing.JLabel p10;
    javax.swing.JLabel p11;
    javax.swing.JLabel p12;
    javax.swing.JLabel p13;
    javax.swing.JLabel p14;
    javax.swing.JLabel p15;
    javax.swing.JLabel p2;
    javax.swing.JLabel p3;
    javax.swing.JLabel p4;
    javax.swing.JLabel p5;
    javax.swing.JLabel p6;
    javax.swing.JLabel p7;
    javax.swing.JLabel p8;
    javax.swing.JLabel p9;
    // End of variables declaration//GEN-END:variables
}
```

Cliente1.java

```
import java.net.*;
import java.io.*;
/**
 *
 * @author FERNANDO
 */
public class Cliente1 {
    public void cliente(){
        try{
            int pto = 5678;
            String host = "localhost";
            Socket cl = new Socket(host,pto);
            System.out.println("Conexión al servidor establecida, comienza
intercambio de objetos... ");
            DataOutputStream dos = new DataOutputStream(cl.getOutputStream());
            ObjectInputStream ois = new ObjectInputStream( cl.getInputStream() );
            dos.writeInt(1);
            String palabras[][] = (String[][]) ois.readObject();
            for(int i = 0; i < 15; i++){
                System.out.println(palabras[0][i] + "\n");//Mostrando palabras
recibidas
                System.out.println(palabras[1][i] + "\n");//Mostrando
definiciones recibidas
            }

            System.out.println("Termina programa");
            ois.close();
            dos.close();
            cl.close();
            Sopa1 sopita = new Sopa1(palabras);
            sopita.setVisible(true);
```

```
        } catch(Exception e){
            e.printStackTrace();
        }
    }
}
```

Cliente2.java

```
import java.net.*;
import java.io.*;
/**
 *
 * @author FernandoHE
 */
public class Cliente2 {
    public void cliente(){
        try{
            int pto = 5678;
            String host = "localhost";
            Socket cl = new Socket(host,pto);
            System.out.println("Conexión al servidor establecida, comienza
intercambio de objetos... ");
            DataOutputStream dos = new DataOutputStream(cl.getOutputStream());
            ObjectInputStream ois = new ObjectInputStream( cl.getInputStream() );
            dos.writeInt(2);
            String palabras[][] = (String[][]) ois.readObject();
            for(int i = 0; i < palabras.length; i++){
                System.out.println(palabras[0][i] + "\n");//Mostrando palabras
recibidas
                System.out.println(palabras[1][i] + "\n");//Mostrando anagramas
de palabras recibidas
            }

            System.out.println("Termina programa");
            ois.close();
            cl.close();
            Sopa2 sopita = new Sopa2(palabras);
            sopita.setVisible(true);
        } catch(Exception e){
            e.printStackTrace();
        }
    }
}
```

Dificultades Encontradas.

La mayor dificultad fue realizar la interfaz y conectar esta con la lógica de la sopa, esto debido a que no había trabajado mucho anteriormente con interfaces gráficas.

Conclusiones.

En esta práctica fue un gran reto para los 2 ya que no teníamos mucha experiencia trabajando con interfaces gráficas, pero después de haber solucionado esa parte lo demás ya no fue tan complicado ya que entre los dos logramos desarrollar la lógica de la sopa de letras, por último, nos sirvió para repasar el uso de los Sockets y el flujo de información a través de ellos.