LABORATORIO #1

Grupo 13

Integrantes:

Añez Vladimirovna Leonardo

Caricari Torrejon Pedro Luis

Oliva Rojas Gerson

Mercado Oudalova Danilo Anatoli

Mollinedo Franco Milena

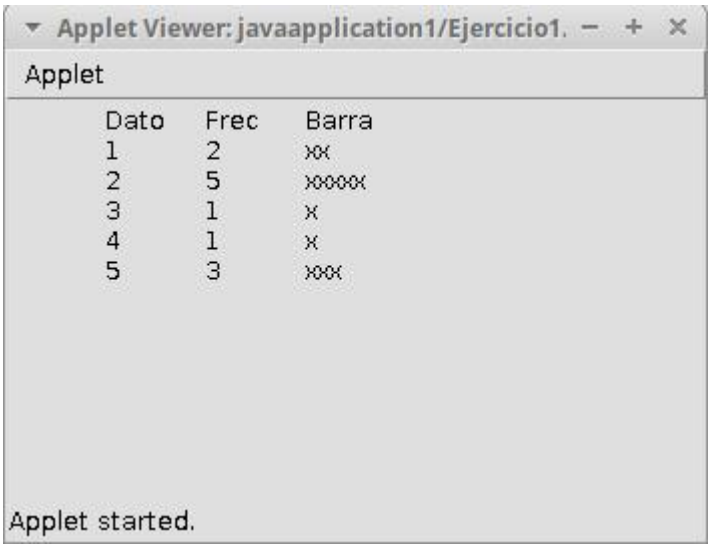
Materia: Interacción Hombre Computador

Fecha: 20/01/2020

Comentario:

Con esta práctica aprendimos la jerarquía que utiliza el lenguaje Java y la estructura que se encuentran en esas clases , pero en especial analizamos la clase Applet y cómo es el manejo de esta ya que se encuentra dividida por pixeles y tiene un eje X y un eje Y que manejamos conceptualmente , esta clase Applet deriva de la clase container que a su vez deriva de la super clase madre llamada “OBJECT”, esta clase Applet maneja componentes gráficos en los cuales se dibujan cadenas de caracteres con el método “drawString” donde los argumentos que recibe son 3 , el String que será dibujado , en que pixel del eje X se va dibujar y en que pixel del eje Y se dibujará , con esta practica aprendimos el manejo de este objeto para graficar distintas tablas.

Ejercicio 1:

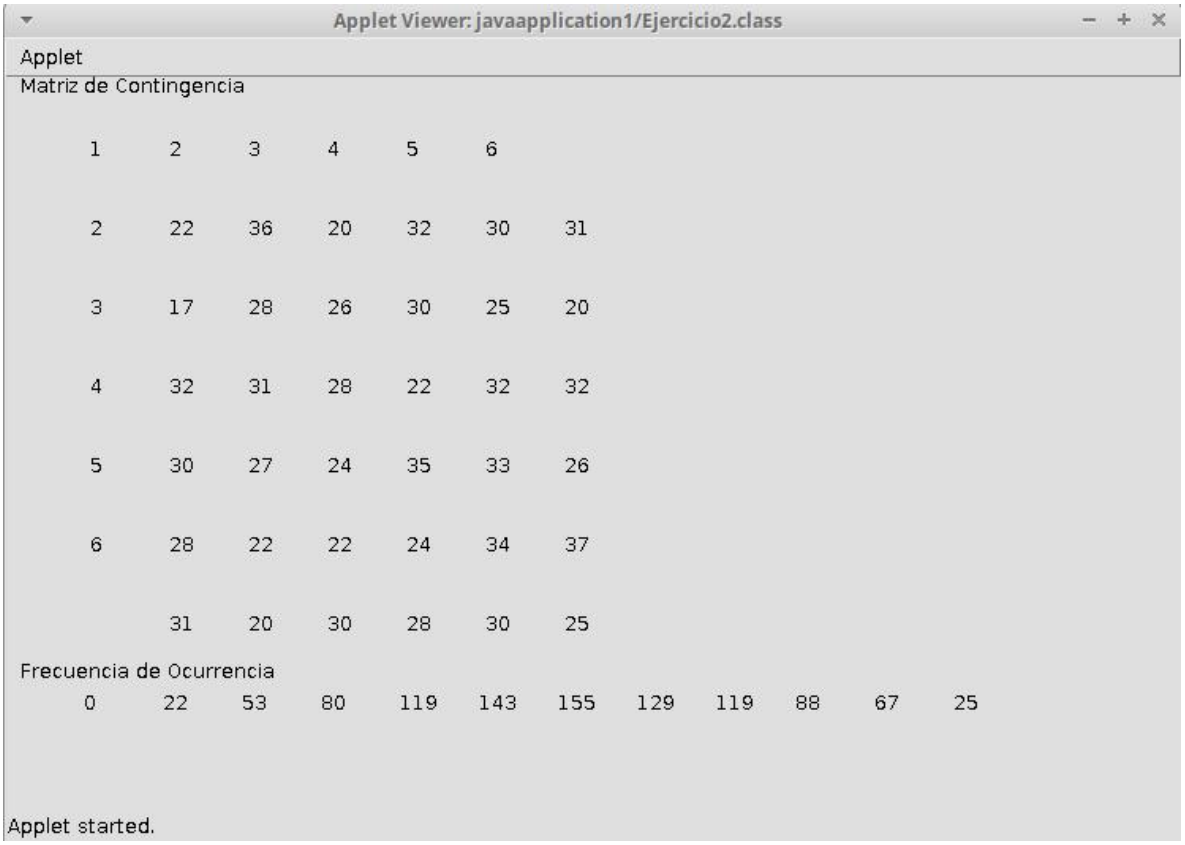



```
/*
 * To change this license header, choose License Headers in Project
Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package javaapplication1;


import java.awt.Graphics;


/**
 *
 * @author blackcat
 */
public class Ejercicio1 extends java.applet.Applet {


    /**
     * Initializes the applet NewApplet
     */
     int a[]= {2,1,5,2,3,1,4,5,5,2,2,2};
      int f[]= {0,0,0,0,0,0};
```

```
public void init() {
    try {
        java.awt.EventQueue.invokeAndWait(new Runnable() {
            public void run() {
                initComponents();
            }
        });
    } catch (Exception ex) {
        ex.printStackTrace();
    }
}
 @Override
public void paint(Graphics g){
        g.drawString("Dato",50,15);
        g.drawString("Frec",100,15);
        g.drawString("Barra",150,15);
        encontrarFrec(a,f);
            for(int i = 1,fila = 30; i<f.length; i++,fila+=15){
              g.drawString(""+i,50,fila);
              g.drawString(""+f[i],100,fila);
              g.drawString(cadena('x',f[i]),150,fila);
        }
  }

  public void encontrarFrec(int a[], int b[]){
        for(int i = 0;i< a.length ; i ++){
              f[a[i]]++;
        }
  }

  public String cadena(char ch, int n){
        String s1 = "";
        for(int i = 1;i<=n; i ++)
              s1=s1+ch;
```

```
            return s1;
      }


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

        setLayout(new java.awt.BorderLayout());
    }// </editor-fold>



    // Variables declaration - do not modify
    // End of variables declaration
}
```

Ejercicio 2:



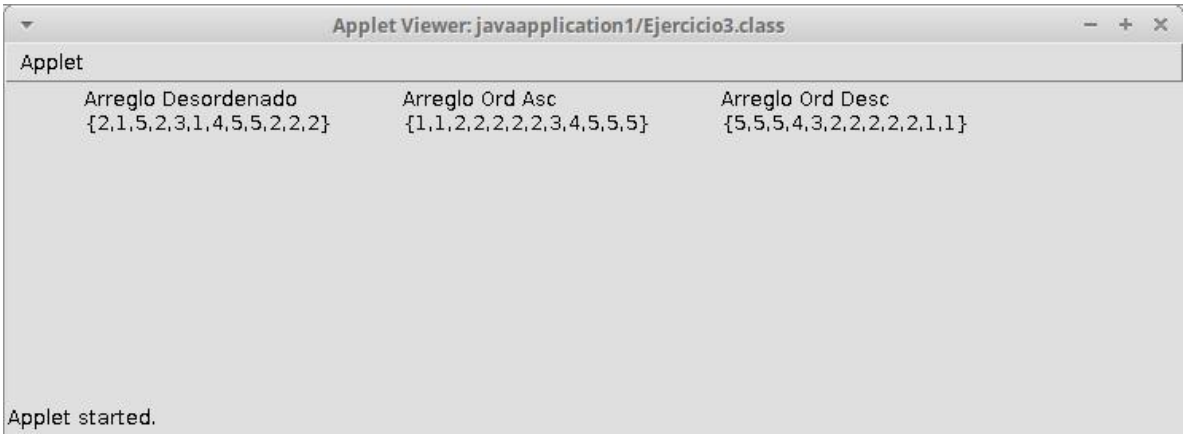

```
/*
 * To change this license header, choose License Headers in Project
Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package javaapplication1;


import java.awt.Graphics;
import java.util.Random;


/**
 *
 * @author blackcat
 */
```

```
public class Ejercicio2 extends java.applet.Applet {

    /**
     * Initializes the applet Ejercicio2
     */
    public void init() {
        randomPool();
        try {
            java.awt.EventQueue.invokeAndWait(new Runnable() {
                public void run() {
                    initComponents();
                }
            });
        } catch (Exception ex) {
            ex.printStackTrace();
        }
    }
     static int mContingencia[][] = new int[7][7];
    static int frecuencia[] = new int[13];

    public static void randomPool(){
            Random random = new Random();
        for(int i=0;i<1000;++i){
            int dice1 = random.nextInt(6)+1;
            int dice2 = random.nextInt(6)+1;

            frecuencia[dice1+dice2]++;

            mContingencia[dice1][dice2]++;

        }
        /*
        for(int i=1;i<=6;++i){
            for(int j=1;j<=6;++j){
```

```
                System.out.print(mContingencia[i][j]);
            }System.out.println("");
        }*/
    }

    static int espaciado = 50;
    public void paint(Graphics g){

        g.drawString("Matriz de Contingencia", 10, 10);

        for(int i=1;i<=6;++i){
            g.drawString(" "+i,i*espaciado,espaciado);
            g.drawString(" "+i,espaciado,i*espaciado);
        }

        for(int i=1;i<=6;++i){
            for(int j=1;j<=6;++j){
                g.drawString("
"+mContingencia[i][j],espaciado+i*espaciado,espaciado+j*espaciado);
            }
        }
        g.drawString("Frecuencia de Ocurrencia", 10, 380);
        for(int i=1;i<=12;++i){

                g.drawString(""+frecuencia[i], espaciado*i, 400);
        }




    }
    /**
     * This method is called from within the init() method to initialize
the
     * form. WARNING: Do NOT modify this code. The content of this method
is
```

```
     * always regenerated by the Form Editor.
     */
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        setLayout(new java.awt.BorderLayout());
    }// </editor-fold>


    // Variables declaration - do not modify
    // End of variables declaration
}
```

Ejercicio 3:

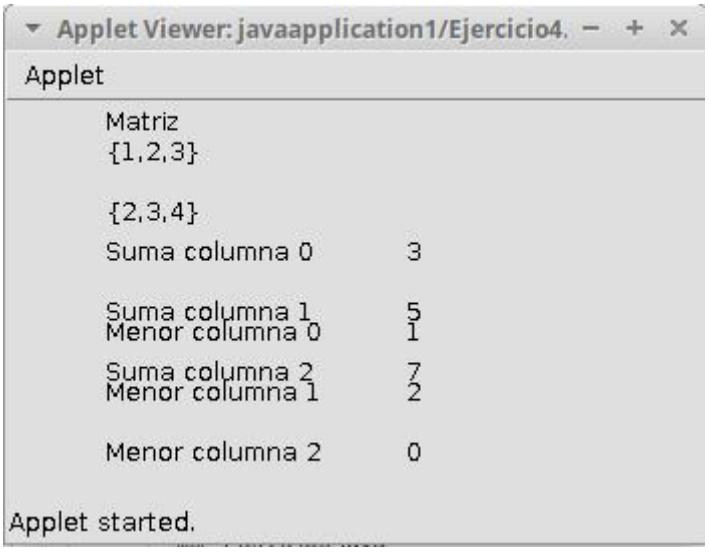



```
package javaapplication1;


import java.awt.Graphics;

import java.util.Arrays;


/*

 * 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 blackcat

 */

public class Ejercicio3 extends java.applet.Applet {


    /**

     * Initializes the applet Ejercicio2

     */

    int a[] = {2, 1, 5, 2, 3, 1, 4, 5, 5, 2, 2, 2};

    String desord = cadena(a);

    public void init() {
```

```
    try {
        java.awt.EventQueue.invokeAndWait(new Runnable() {
            public void run() {
                initComponents();
            }
        });
    } catch (Exception ex) {
        ex.printStackTrace();
    }
}

@Override
public void paint(Graphics g) {
    g.drawString("Arreglo Desordenado", 50, 15);
    g.drawString("Arreglo Ord Asc", 250, 15);
    g.drawString("Arreglo Ord Desc", 450, 15);

    g.drawString(desord, 50, 30);
    Arrays.sort(a);
    g.drawString(cadena(a), 250, 30);
    g.drawString(cadena(invertir(a)),450,30);



}

public int[] invertir(int a[]){
    int[] invertido = new int[a.length];
    for(int i=0;i<a.length;i++){
        invertido[i] = a[a.length-1-i];
    }
    return invertido;
}

public String cadena(int a[]) {
```

```
        String s1 = "{";

        for (int i = 0; i < a.length-1; i++) {
            s1 = s1 + a[i]+",";
        }
        s1= s1 + a[a.length-1] + "}";
        return s1;
    }



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

        setLayout(new java.awt.BorderLayout());
    }// </editor-fold>


    // Variables declaration - do not modify
    // End of variables declaration
}
```

Ejercicio 4:



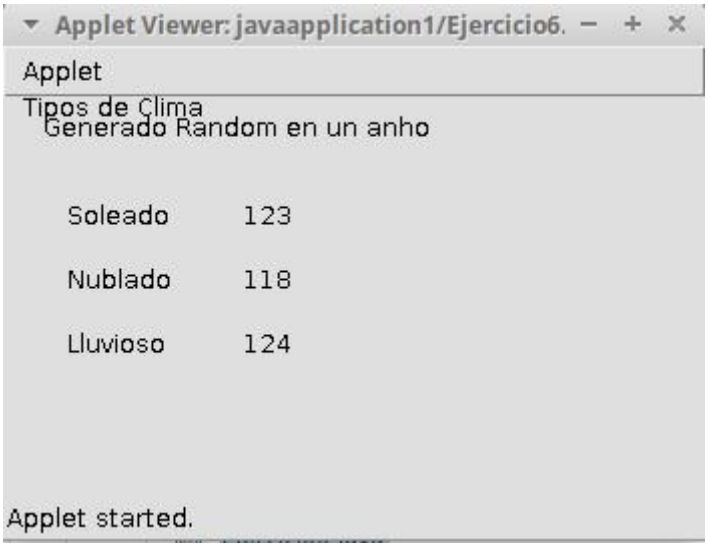

```
/*
 * To change this license header, choose License Headers in Project
Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package javaapplication1;

import java.awt.Graphics;
import java.util.Arrays;

/**
 *
 * @author blackcat
 */
public class Ejercicio4 extends java.applet.Applet {

    int a[][] = {
        {1, 2, 3}, {2, 3, 4}
    };
    int f = 2;
    int c = 3;
```

```
/**
 * Initializes the applet Ejercicio4
 */
public void init() {
    try {
        java.awt.EventQueue.invokeAndWait(new Runnable() {
            public void run() {
                initComponents();
            }
        });
    } catch (Exception ex) {
        ex.printStackTrace();
    }
}

public void paint(Graphics g) {
    g.drawString("Matriz", 50, 15);
    for (int i = 0, fila = 30; i < f; i++, fila += 30) {
        g.drawString(cadena(a[i]), 50, fila);
    }
    int ss[] = sum(a);
    for (int i = 0, fila = 80; i < c; i++, fila += 30) {
        g.drawString("Suma columna " + i, 50, fila);
        g.drawString(String.valueOf(ss[i]), 200, fila);
    }
    int mcc[] = minc(a);
    for (int i = 0, fila = 120; i < c; i++, fila += 30) {
        g.drawString("Menor columna " + i, 50, fila);
        g.drawString(String.valueOf(mcc[i]), 200, fila);
    }

}
```

```
    public String cadena(int a[]) {
        String s1 = "{";

        for (int i = 0; i < a.length - 1; i++) {
            s1 = s1 + a[i] + ",";
        }
        s1 = s1 + a[a.length - 1] + "}";
        return s1;
    }

    public static int[] sum(int[][] mat) {
        int[] myArray = new int[mat[0].length];

        for (int i = 0; i < mat[0].length; i++) {
            int suma = 0;
            for (int j = 0; j < mat.length; j++) {
                suma += mat[j][i];

            }
            myArray[i] = suma;

        }

        return myArray;
    }

    public  int[] minc(int[][] mat) {
        int[] menorColumna = new int[c];
        for (int i = 0; i < f; i++) {
            menorColumna[i] = mat[i][0];//valor minimo inicial para la
columna i
            for (int j = 0; j < c; j++) {
                if (mat[i][j] < menorColumna[i]) {
                    menorColumna[i] = mat[i][j];
```

```
                }
            }
        }
        return menorColumna;
    }

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

        setLayout(new java.awt.BorderLayout());
    }// </editor-fold>



    // Variables declaration - do not modify
    // End of variables declaration
}
```

Ejercicio 6:

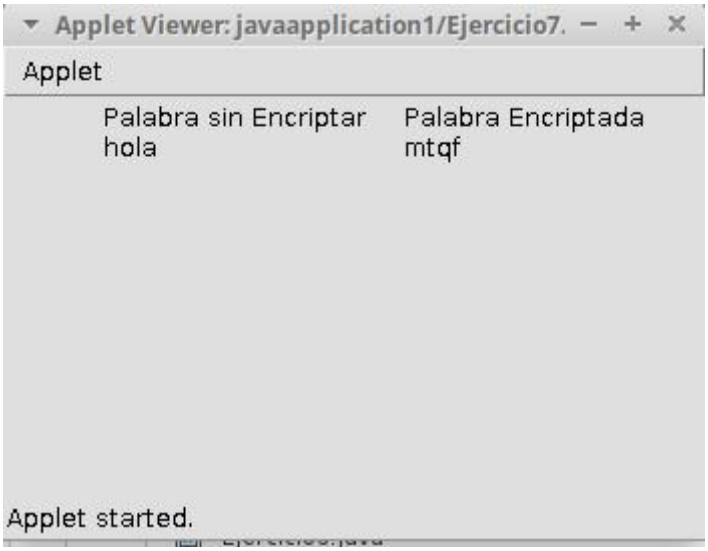



```
/*
 * To change this license header, choose License Headers in Project
Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package javaapplication1;

import java.awt.Graphics;
import java.util.Random;

/**
 *
 * @author blackcat
 */
public class Ejercicio6 extends java.applet.Applet {

    /**
     * Initializes the applet Ejercicio6
     */
    static int generacion[] = new int[3];
```

```
public void init() {
    randomGenerate();
    try {
        java.awt.EventQueue.invokeAndWait(new Runnable() {
            public void run() {
                initComponents();
            }
        });
    } catch (Exception ex) {
        ex.printStackTrace();
    }
}

static void randomGenerate() {
    Random random = new Random();
    for (int i = 0; i < 365; ++i) {
        int rand = random.nextInt(3);
        generacion[rand]++;
    }
}

/*
   public static void main(String[] args) {
    randomGenerate();
    }*/
@Override
public void paint(Graphics g) {

    g.drawString("Tipos de Clima ", 10, 10);
    g.drawString("Generado Random en un anho", 20, 20);

    g.drawString("Soleado  ", 32, 64);
    g.drawString("" + generacion[0], 120, 64);
```

```
        g.drawString("Nublado  ", 32, 96);
        g.drawString("" + generacion[1], 120, 96);
        g.drawString("Lluvioso ", 32, 128);
        g.drawString("" + generacion[2], 120, 128);
    }

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

        setLayout(new java.awt.BorderLayout());
    }// </editor-fold>


    // Variables declaration - do not modify
    // End of variables declaration
}
```

Ejercicio 7:




```
/*
 * 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.
 */
package javaapplication1;

import java.awt.Graphics;
import java.util.Random;

/**
 *
 * @author blackcat
 */
public class Ejercicio7 extends java.applet.Applet {

    /**
     * Initializes the applet Ejercicio7
     */
    public void init() {
```

```
    try {
        java.awt.EventQueue.invokeAndWait(new Runnable() {
            public void run() {
                initComponents();
            }
        });
    } catch (Exception ex) {
        ex.printStackTrace();
    }
}
@Override
public void paint(Graphics g) {
    g.drawString("Palabra sin Encriptar", 50, 15);
    g.drawString("Palabra Encriptada", 200, 15);

    g.drawString("mtqf",200,30);
    g.drawString(decrypt("mtqf", 5),50,30);




}
static char[] chars = {
    'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h',
    'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p',
    'q', 'r', 's', 't', 'u', 'v', 'w', 'x',
    'y', 'z', '0', '1', '2', '3', '4', '5',
    '6', '7', '8', '9', 'A', 'B', 'C', 'D',
    'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L',
    'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T',
    'U', 'V', 'W', 'X', 'Y', 'Z', '!', '@',
    '#', '$', '%', '^', '&', '(', ')', '+',
    '-', '*', '/', '[', ']', '{', '}', '=',
    '<', '>', '?', '_', '"', '.', ',', ' '
};
```

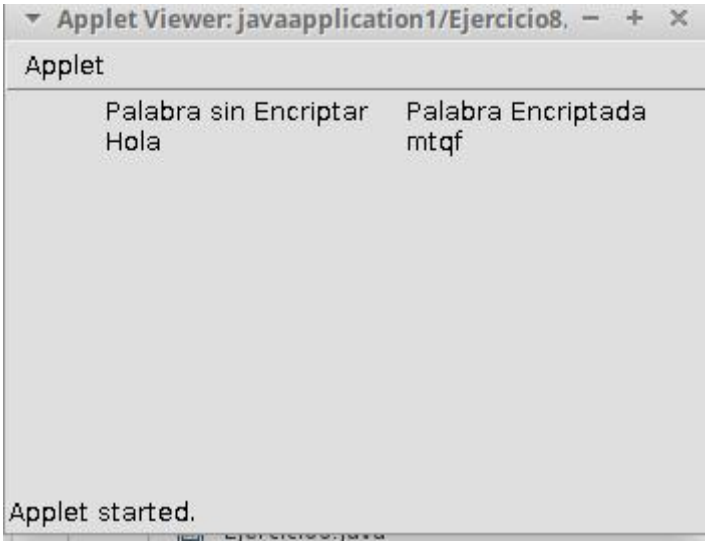
```
    static String decrypt(String cip, int offset)
    {
        char[] cipher = cip.toCharArray();
        for (int i = 0; i < cipher.length; i++) {
            for (int j = 0; j < chars.length; j++) {
                if (j >= offset && cipher[i] == chars[j]) {
                    cipher[i] = chars[j - offset];
                    break;
                }
                if (cipher[i] == chars[j] && j < offset) {
                    cipher[i] = chars[(chars.length - offset +1) + j];
                    break;
                }
            }
        }
        return String.valueOf(cipher);
    }
    /**
     * This method is called from within the init() method to initialize
the
     * form. WARNING: Do NOT modify this code. The content of this method
is
     * always regenerated by the Form Editor.
     */
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        setLayout(new java.awt.BorderLayout());
    }// </editor-fold>



    // Variables declaration - do not modify
    // End of variables declaration
}
```

```
    private void initComponents() {

        setLayout(new java.awt.BorderLayout());
    }// </editor-fold>


    // Variables declaration - do not modify
    // End of variables declaration
}
*
 * @author blackcat
 */
public class Ejercicio8 extends java.applet.Applet {

    /**
     * Initializes the applet Ejercicio8
     */
    public void init() {
        try {
            java.awt.EventQueue.invokeAndWait(new Runnable() {
                public void run() {
                    initComponents();
                }
            });
        } catch (Exception ex) {
            ex.printStackTrace();
        }
    }
    @Override
    public void paint(Graphics g) {
        g.drawString("Palabra sin Encriptar", 50, 15);
        g.drawString("Palabra Encriptada", 200, 15);
        g.drawString("Hola",50,30);
```

```
    g.drawString(encrypt("hola", 5),200,30);




}
static char[] chars = {
    'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h',
    'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p',
    'q', 'r', 's', 't', 'u', 'v', 'w', 'x',
    'y', 'z', '0', '1', '2', '3', '4', '5',
    '6', '7', '8', '9', 'A', 'B', 'C', 'D',
    'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L',
    'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T',
    'U', 'V', 'W', 'X', 'Y', 'Z', '!', '@',
    '#', '$', '%', '^', '&', '(', ')', '+',
    '-', '*', '/', '[', ']', '{', '}', '=',
    '<', '>', '?', '_', '"', '.', ',', ' '
};
static String encrypt(String text, int offset)
{
    char[] plain = text.toCharArray();

    for (int i = 0; i < plain.length; i++) {
        for (int j = 0; j < chars.length; j++) {
            if (j <= chars.length - offset) {
                if (plain[i] == chars[j]) {
                    plain[i] = chars[j + offset];
                    break;
                }
            }
            else if (plain[i] == chars[j]) {
                plain[i] = chars[j - (chars.length - offset + 1)];
            }
```

```
            }
        }
        return String.valueOf(plain);
    }



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

        setLayout(new java.awt.BorderLayout());
    }// </editor-fold>



    // Variables declaration - do not modify
    // End of variables declaration
}
```

Ejercicio 8:




```
/*
 * 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.
 */
package javaapplication1;

import java.awt.Graphics;
import java.util.Arrays;

/**
 *
 * @author blackcat
 */
public class Ejercicio8 extends java.applet.Applet {

    /**
     * Initializes the applet Ejercicio8
     */
```

```
public void init() {
    try {
        java.awt.EventQueue.invokeAndWait(new Runnable() {
            public void run() {
                initComponents();
            }
        });
    } catch (Exception ex) {
        ex.printStackTrace();
    }
}
@Override
public void paint(Graphics g) {
    g.drawString("Palabra sin Encriptar", 50, 15);
    g.drawString("Palabra Encriptada", 200, 15);
    g.drawString("Hola",50,30);
    g.drawString(encrypt("hola", 5),200,30);




}
static char[] chars = {
    'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h',
    'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p',
    'q', 'r', 's', 't', 'u', 'v', 'w', 'x',
    'y', 'z', '0', '1', '2', '3', '4', '5',
    '6', '7', '8', '9', 'A', 'B', 'C', 'D',
    'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L',
    'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T',
    'U', 'V', 'W', 'X', 'Y', 'Z', '!', '@',
    '#', '$', '%', '^', '&', '(', ')', '+',
    '-', '*', '/', '[', ']', '{', '}', '=',
    '<', '>', '?', '_', '"', '.', ',', ' '
```

```
    };
    static String encrypt(String text, int offset)
    {
        char[] plain = text.toCharArray();

        for (int i = 0; i < plain.length; i++) {
            for (int j = 0; j < chars.length; j++) {
                if (j <= chars.length - offset) {
                    if (plain[i] == chars[j]) {
                        plain[i] = chars[j + offset];
                        break;
                    }
                }
                else if (plain[i] == chars[j]) {
                    plain[i] = chars[j - (chars.length - offset + 1)];
                }
            }
        }
        return String.valueOf(plain);
    }



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

        setLayout(new java.awt.BorderLayout());
    }// </editor-fold>
```

```
    // Variables declaration - do not modify
    // End of variables declaration
}
.length - offset) {
                    if (plain[i] == chars[j]) {
                        plain[i] = chars[j + offset];
                        break;
                    }
                }
                else if (plain[i] == chars[j]) {
                    plain[i] = chars[j - (chars.length - offset + 1)];
                }
            }
        }
        return String.valueOf(plain);
    }



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

        setLayout(new java.awt.BorderLayout());
    }// </editor-fold>
```

```
    // Variables declaration - do not modify
    // End of variables declaration
}
  static char[] chars = {
        'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h',
        'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p',
        'q', 'r', 's', 't', 'u', 'v', 'w', 'x',
        'y', 'z', '0', '1', '2', '3', '4', '5',
        '6', '7', '8', '9', 'A', 'B', 'C', 'D',
        'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L',
        'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T',
        'U', 'V', 'W', 'X', 'Y', 'Z', '!', '@',
        '#', '$', '%', '^', '&', '(', ')', '+',
        '-', '*', '/', '[', ']', '{', '}', '=',
        '<', '>', '?', '_', '"', '.', ',', ' '
    };
    static String encrypt(String text, int offset)
    {
        char[] plain = text.toCharArray();

        for (int i = 0; i < plain.length; i++) {
            for (int j = 0; j < chars.length; j++) {
                if (j <= chars.length - offset) {
                    if (plain[i] == chars[j]) {
                        plain[i] = chars[j + offset];
                        break;
                    }
                }
                else if (plain[i] == chars[j]) {
                    plain[i] = chars[j - (chars.length - offset + 1)];
                }
            }
        }
        return String.valueOf(plain);
```

```
    }


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

        setLayout(new java.awt.BorderLayout());
    }// </editor-fold>


    // Variables declaration - do not modify
    // End of variables declaration
}
```