



Instituto Tecnológico de Oaxaca



“Tecnología Propia e Independencia Económica”

Tarea: Ejercicios 5,6 y 7 Java

Carrera: Ingeniería en Sistemas Computacionales.

Estudiante: Juárez Lucas Edson Rubén

No. de Control: 18161150

Grupo y Hora: 9-10 am

Profesor: Luis Alberto Alonso Hernández

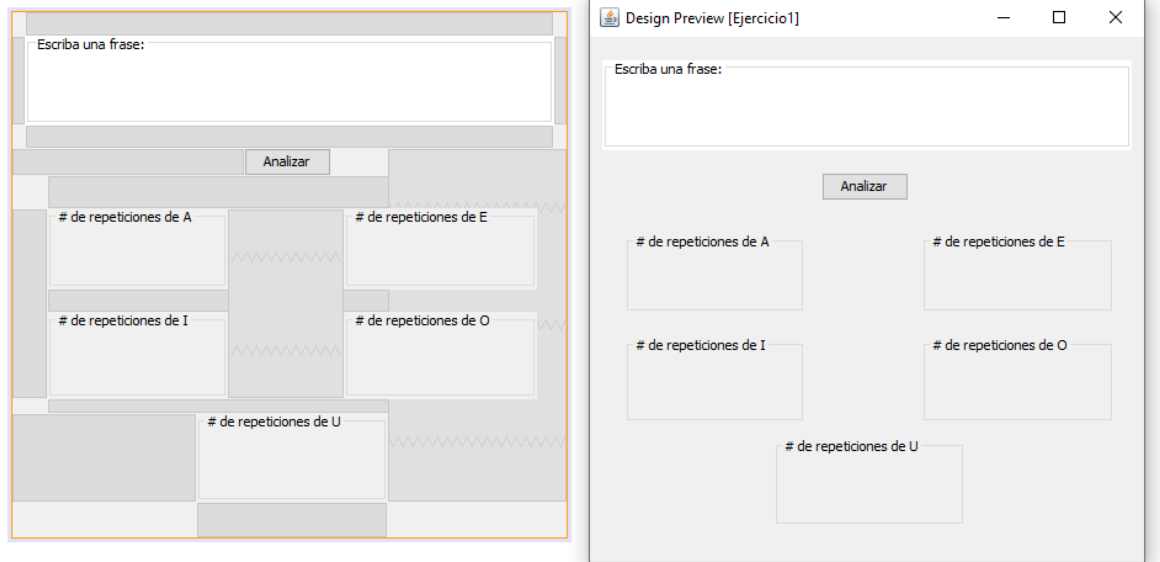
Oaxaca de Juárez, Oaxaca.

CADENAS (CONTINUACIÓN) PDF #5

Ejercicio 1

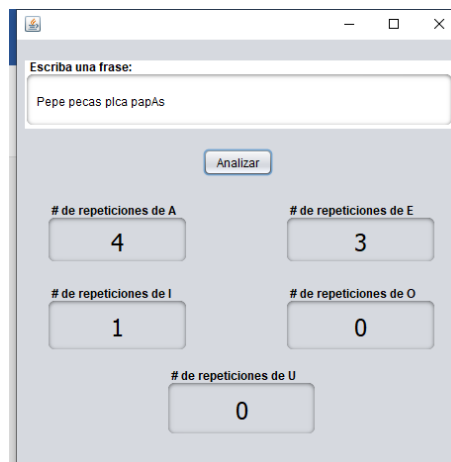
Realizar un programa cuya ventana tenga los siguientes elementos:

- Un cuadro de texto llamado txtFrase.
- Varias etiquetas.
- Un botón "Analizar" llamado btnAnalizar.

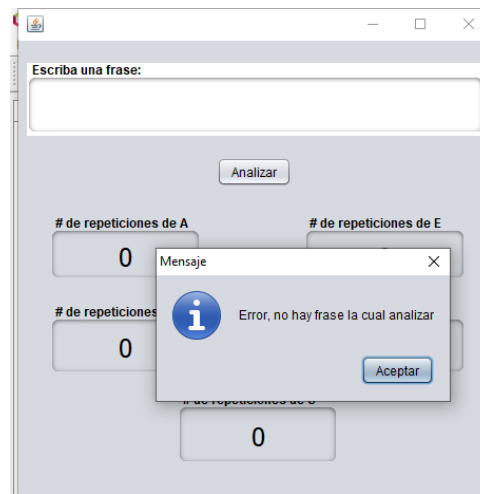


El programa debe contar cuantas vocales tiene la frase. El funcionamiento será el siguiente:

- El usuario escribirá una frase en el cuadro de texto txtFrase. Luego se pulsará el botón Analizar.
- El programa mostrará en las etiquetas el número de a, de e, de i de o y de u que se encuentran en la frase.
- Tenga en cuenta que puede haber vocales en mayúsculas y en minúsculas.



- Si el usuario no introduce nada en el cuadro de texto txtFrase, entonces el programa debería mostrar un error.



```

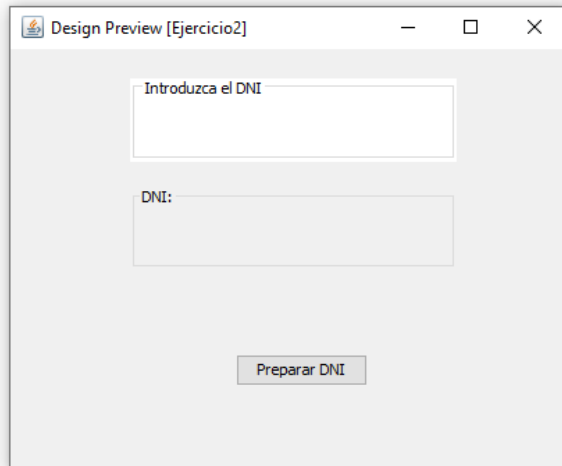
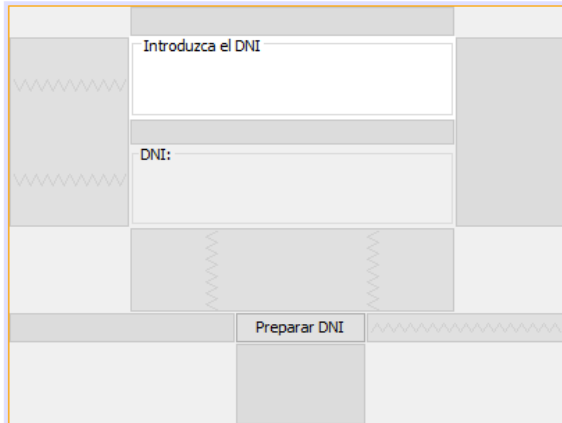
132
133 private void btnAnalizarActionPerformed(java.awt.event.ActionEvent evt) {
134     // TODO add your handling code here:
135     String frase = txtFrase.getText();
136     String frase2= frase.toLowerCase();
137     if (txtFrase.getText()!=null||frase!="||txtFrase.getText()!=||frase!=null||frase.equals(anObject:"")||frase.equals(anObject:null)) {
138         int a =cuentas(sPalabra:frase2, a:'a');
139         int e =cuentas(sPalabra:frase2, a:'e');
140         int i =cuentas(sPalabra:frase2, a:'i');
141         int o =cuentas(sPalabra:frase2, a:'o');
142         int u =cuentas(sPalabra:frase2, a:'u');
143         etiVocalA.setText(""+ a);
144         etiVocalE.setText(""+ e);
145         etiVocalI.setText(""+ i);
146         etiVocalO.setText(""+ o);
147         etiVocalU.setText(""+ u);
148
149     } else if (txtFrase.getText()==null||frase=="||txtFrase.getText()=="||frase==null||frase.equals(anObject:"")||frase.equals(anObject:null)){
150         JOptionPane.showMessageDialog(parentComponent:null, message:"Error, no hay frase la cual analizar");
151     }
152 }
153

```

Ejercicio 2

Realizar un programa cuya ventana tenga los siguientes elementos:

- Un cuadro de texto llamado txtDNI.
- Una etiqueta llamada etiDNI.
- Un botón "Preparar DNI" llamado btnPrepararDNI.



El programa funcionará de la siguiente forma:

- El usuario introducirá un DNI en el cuadro de texto llamado txtDNI y luego pulsará el botón Preparar DNI. El resultado será que el DNI introducido aparecerá "preparado" en la etiqueta etiDNI.
- A continuación, se explica cómo preparar el DNI:

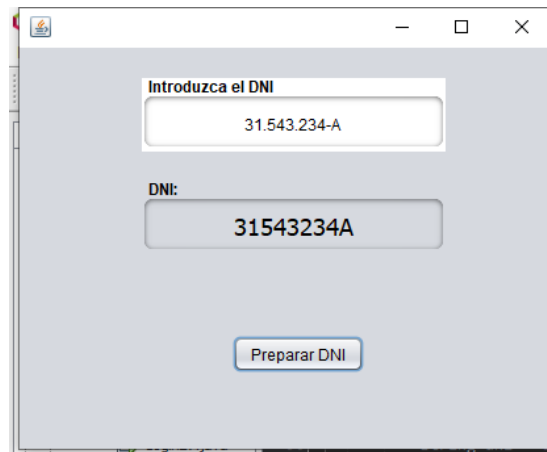
El usuario puede introducir un DNI con uno de estos formatos:

31.543.234-A
31.543.234 A
31.543.234A
31 543 234 A
etc.

Sin embargo, cuando el usuario pulse el botón Preparar DNI, en la etiqueta etiDNI debe aparecer el DNI con el siguiente formato:

31543234A

Es decir, sin ningún espacio y sin puntos ni guiones.



- Si el usuario no introduce nada en el cuadro de texto del DNI y pulsa el botón, entonces debe aparecer un error emergente (JOptionPane).



```
private void btnPrepararDNIActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    String dni = txtDNI.getText();  
    if (dni!="") {  
        String dni1 = dni.replace( target: ".", replacement: "");  
        String dni2 = dni1.replace( target: " ", replacement: "");  
        String dni3 = dni2.replace( target: "-", replacement: "");  
        String dni4 = dni3.replace( target: "-", replacement: "");  
        etiDNI.setText( text:dni4);  
    } else {  
        JOptionPane.showMessageDialog( parentComponent:null, message: "error");  
    }  
}
```

Ejercicio 3

Se pide hacer un programa que le permita al usuario introducir una palabra en latín de la primera declinación, y a continuación generar sus “casos” en plural y singular.

Las palabras de la primera declinación en latín son sustantivos femeninos (la mayoría), que terminan en a, como, por ejemplo: ROSA o ANIMA.

Estas palabras tienen las siguientes variantes o “casos”:

Por ejemplo, para la palabra ROSA, sus casos son los siguientes:

CASO	SINGULAR	PLURAL
NOMINATIVO	Rosa	Rosae
VOCATIVO	Rosa	Rosae
ACUSATIVO	Rosam	Rosas
GENITIVO	Rosae	Rosarum
DATIVO	Rosae	Rosis
ABLATIVO	Rosa	Rosis

Por ejemplo, para la palabra ANIMA, sus casos son los siguientes:

CASO	SINGULAR	PLURAL
NOMINATIVO	Anima	Animae
VOCATIVO	Anima	Animae
ACUSATIVO	Animam	Animas
GENITIVO	Animae	Animarum
DATIVO	Animae	Animis
ABLATIVO	Anima	Animis

Debes observar que algunos casos son exactamente iguales a la palabra inicial, como por ejemplo el Nominativo Singular.

Otros casos, en cambio, se construyen añadiendo algunas letras al final de la palabra inicial. Por ejemplo, el Acusativo singular se construye añadiendo una “m” a la palabra inicial.

Para construir el Dativo y Ablativo plural, es necesario concatenar la palabra inicial (sin la a final) con “is”. Por ejemplo, en el caso de la palabra Rosa, se concatenaría: Ros + is.

TENIENDO EN CUENTA LO ANTERIOR, SE PIDE REALIZAR EL SIGUIENTE PROGRAMA:
Realice un programa cuya ventana tenga los siguientes elementos:

- Un cuadro de texto txtPalabra.
- Doce etiquetas al menos correspondientes a los 6 casos en singular y plural.
- Un botón “Declinar” llamado btnDeclinar.

El diagrama muestra la estructura de la interfaz de usuario. A la izquierda, una columna vertical contiene las etiquetas de los casos: Nominativo, Vocativo, Acusativo, Genitivo, Dativo y Ablativo. A la derecha de cada etiqueta hay dos cuadros de texto para ingresar el singular y el plural. En la parte superior, hay una fila de encabezados: 'Caso', 'Singular' y 'Plural'. A la derecha del formulario, hay un cuadro de texto etiquetado 'Escriba la palabra a Declinar:' y un botón etiquetado 'Declinar'.

El programa funcionará de la siguiente forma:

- El usuario introducirá una palabra en latín de la primera declinación en el cuadro de texto, y luego pulsará el botón "Declinar". Entonces en las etiquetas aparecerán los casos declinados de la palabra.

Caso	Singular	Plural
Nominativo	anima	animae
Vocativo	anima	animae
Acusativo	animam	animas
Gentitivo	animae	animarum
Dativo	animae	animis
Ablativo	anima	animis

Escriba la palabra a Declinar:
Anima

Declinar

- Si el usuario introduce una palabra que no termine en "a", entonces el programa dará un error, ya que dicha palabra no es de la primera declinación.

Escriba la palabra a Declinar:
Perros

Declinar

Mensaje
Error, la palabra no termina con a
Aceptar

- Tenga en cuenta que la palabra introducida puede estar en mayúsculas o minúsculas.

```
private void btnDeclinarActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    String latin0 = txtPalabra.getText();  
    String latin = latin0.toLowerCase();  
    if (latin.endsWith(suffix:"a")) {  
        etiSNom.setText(text: latin);  
        etiSVoc.setText(text: latin);  
        etiSAcu.setText(latin + "m");  
        etiSGen.setText(latin + "e");  
        etiSDat.setText(latin + "e");  
        etiSAb.setText(text: latin );  
        etiPNom.setText(latin+ "e");  
        etiPVoc.setText(latin+ "e");  
        etiPAcu.setText(latin+"s");  
        etiPGen.setText(latin+"rum");  
        String a = latin.substring(beginIndex:0,latin.length()-1);  
        etiPDat.setText(a+ "is");  
        etiPAb.setText(a+"is");  
    } else {  
        JOptionPane.showMessageDialog( parentComponent:null, message:"Error, la palabra no termina con a");  
    }  
}
```

PROGRAMACIÓN

JAVA

EXCEPCIONES PDF #6

Ejercicio 4

Realizar un programa que le permita al usuario introducir una frase, una posición inicial y una posición final. (Tres cuadros de texto llamados *txtFrase*, *txtPosIni* y *txtPosFin*).

Diagrama de la interfaz de usuario para el ejercicio 4. La interfaz está organizada en un formulario con los siguientes elementos:

- Un cuadro de texto grande etiquetado "Frase".
- Dos cuadros de texto más pequeños etiquetados "Posición Inicial" y "Posición Final", situados uno a la izquierda y otro a la derecha.
- Un cuadro de texto grande etiquetado "Subcadena".
- Un botón etiquetado "Analizar" situado en la parte inferior derecha.

El programa debe mostrar la subcadena contenida entre la posición inicial y la posición final (use una etiqueta *etiSubcadena*)

Esto sucederá cuando se pulse un botón "Analizar" llamado *btnAnalizar*.

Captura de pantalla de la interfaz de usuario ejecutando el programa. La interfaz muestra los siguientes datos de entrada y salida:

- El cuadro de texto "Frase" contiene: "Los programadores del ITO son muy buenos".
- El cuadro de texto "Posición Inicial" contiene: "7".
- El cuadro de texto "Posición Final" contiene: "23".
- El cuadro de texto "Subcadena" muestra el resultado: "gramadores del I".
- El botón "Analizar" está visible en la parte inferior.

PROGRAMACIÓN

JAVA

El código de este programa puede generar errores de ejecución (excepciones), en el caso de que el usuario no introduzca nada en el cuadro de texto de la frase, o en el caso de que el usuario introduzca un valor incorrecto de los cuadros de texto de las posiciones.



El programa debe ser capaz de capturar las excepciones producidas y mostrar un mensaje de error.

```
private void btnAnalizarActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    try {  
        String frase = txtFrase.getText();  
        int ini = Integer.parseInt(txtInicio.getText());  
        int end = Integer.parseInt(txtFinal.getText());  
        String subs = frase.substring(beginIndex:ini, endIndex:end);  
        etiSubcadena.setText(text:subs);  
    } catch (Exception e) {  
        JOptionPane.showMessageDialog(parentComponent:null, message:"Error, los datos no son los correctos");  
    }  
}
```

/**

PROGRAMACIÓN

JAVA

Ejercicio 5

Realizar un programa que pida las coordenadas de una recta, es decir, las coordenadas del punto p1 (x1, y1) y las coordenadas del punto p2 (x2, y2).

Así pues, el programa tendrá cuatro cuadros de texto: *txtX1*, *txtY1*, *txtX2* y *txtY2* donde el usuario introducirá las coordenadas.

El programa debe calcular la distancia de la recta, usando la siguiente fórmula:

$$\text{Raíz}((x2-x1)^2 + (y2-y1)^2)$$

```
private void btnCalcularActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    try {  
        double x1 = Double.parseDouble(s.txtX1.getText());  
        double y1 = Double.parseDouble(s.txtY1.getText());  
        double x2 = Double.parseDouble(s.txtX2.getText());  
        double y2 = Double.parseDouble(s.txtY2.getText());  
  
        double d = Math.sqrt(Math.pow((x2-x1), 2)+ Math.pow((y2-y1), 2));  
  
        etiResultado.setText(""+ d);  
    } catch (Exception e) {  
        JOptionPane.showMessageDialog(parentComponent:null, message:"Error, escriba bien los digitos");  
    }  
}
```

Añada un botón “Calcular” llamado *btnCalcular* que realice el cálculo. El código que introduzca en este botón debe ser capaz de capturar cualquier tipo de excepción producida. Interesa que aparezca un mensaje indicando el tipo de error producido, y el tipo de excepción.

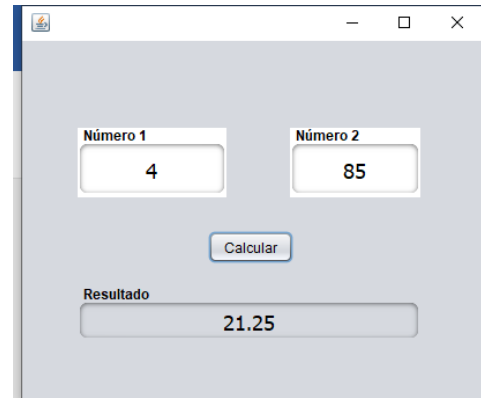
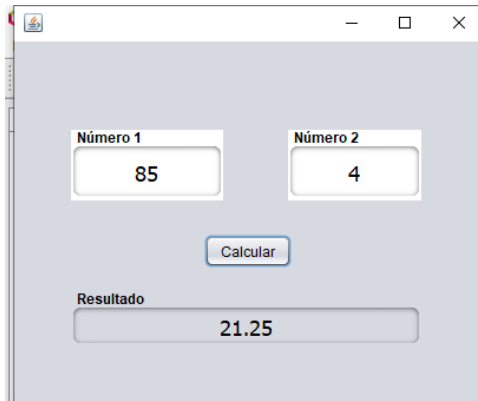
PROGRAMACIÓN

JAVA

Ejercicio 6

Realice un programa que le pida dos números al usuario. Ambos números deben estar comprendidos entre 0 y 100.

El programa debe calcular la división y el resto del mayor entre el menor.



El programa debe capturar todo tipo de excepciones producidas en el código.

```
private void btnCalcularActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    try {  
        double x = Double.parseDouble(s:txtDividendo.getText());  
        double y = Double.parseDouble(s:txtDivisor.getText());  
        if((y>100)|| (x<0)|| (y>100)&&(x<0)){  
            throw new ExcepcionIntervalo(msg:"Números fuera de rango");  
        }else if (x>y){  
            double r= x/y;  
            etiResultado.setText(""+r);  
        }else if (y>x){  
            double r= y/x;  
            etiResultado.setText(""+r);  
        }  
    } catch (NumberFormatException e) {  
        etiResultado.setText(text:"Ingrese digitos válidos");  
    } catch (ArithmeticException ex){  
        etiResultado.setText(text:"División entre cero");  
    } catch (ExcepcionIntervalo ex){  
        etiResultado.setText(text:"Números fuera de rango");  
    }  
}
```

Además, debe hacer que, si el usuario introduce un número no comprendido entre 0 y 100, el código genere una excepción propia.

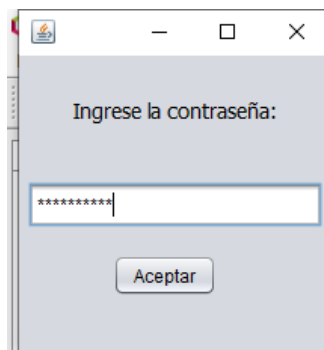
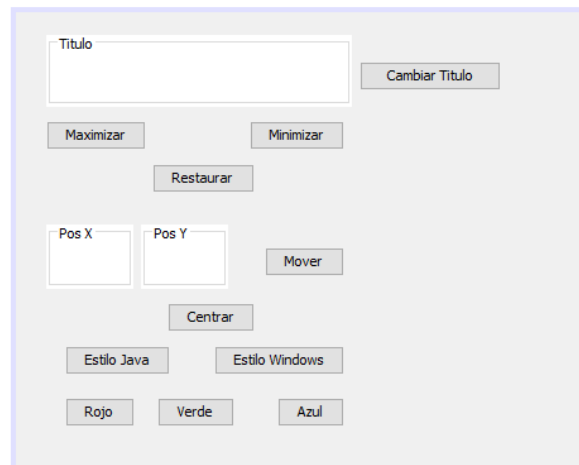
```
public class ExcepcionIntervalo extends Exception {  
    public ExcepcionIntervalo(String msg) {  
        super(string:msg);  
    }  
}
```

LA CLASE JFRAME PDF #7

Ejercicio 7

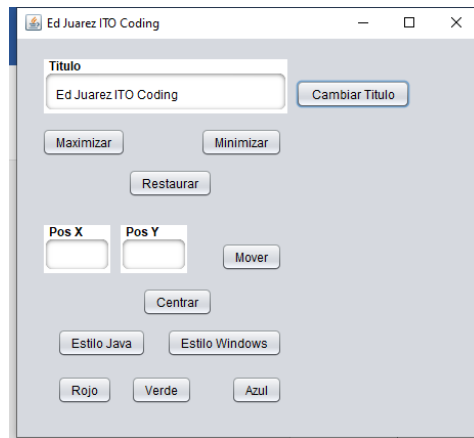
Realizar un programa cuya ventana contenga los siguientes elementos:

- Un cuadro de texto llamado txtTitulo.
- Un botón "Cambiar Título" llamado btnCambiarTitulo.
- Un botón "Maximizar" llamado btnMaximizar.
- Un botón "Minimizar" llamado btnMinimizar.
- Un botón "Restaurar" llamado btnRestaurar.
- Un cuadro de texto llamado txtPosicionX.
- Un cuadro de texto llamado txtPosicionY.
- Un botón "Mover" llamado btnMover.
- Un botón "Centrar" llamado btnCentrar.
- Un botón "Estilo Java" llamado btnEstiloJava.
- Un botón "Estilo Windows" llamado btnEstiloWindows.
- Un botón "Rojo" llamado btnRojo.
- Un botón "Verde" llamado btnVerde.
- Un botón "Azul" llamado btnAzul.

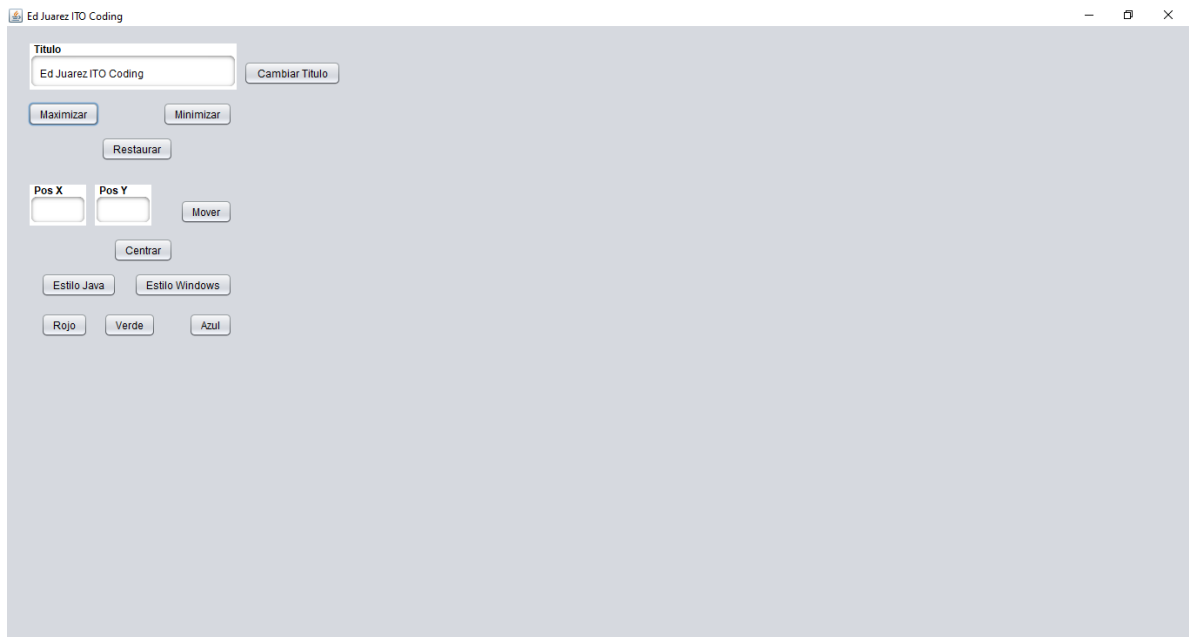


El funcionamiento del programa será el siguiente:

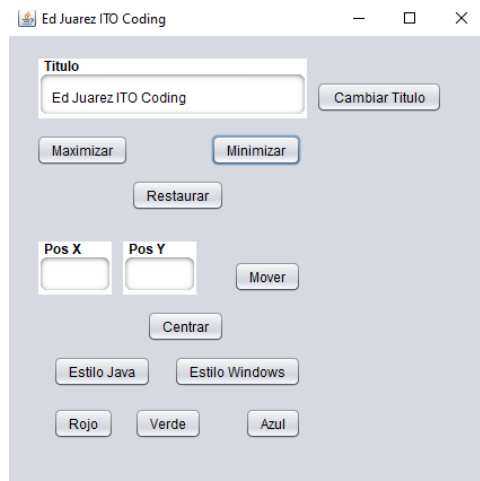
- Si el usuario pulsa el botón “Cambiar Titulo”, el título de la ventana cambiará, colocándose como título lo que el usuario haya escrito en el cuadro de texto txtTitulo.



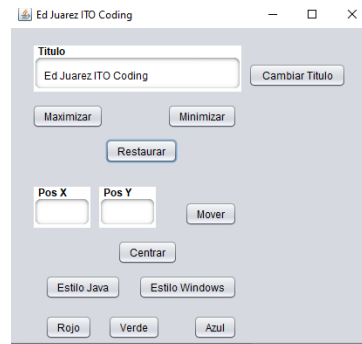
- Si el usuario pulsa el botón “Maximizar”, la ventana se maximizará.



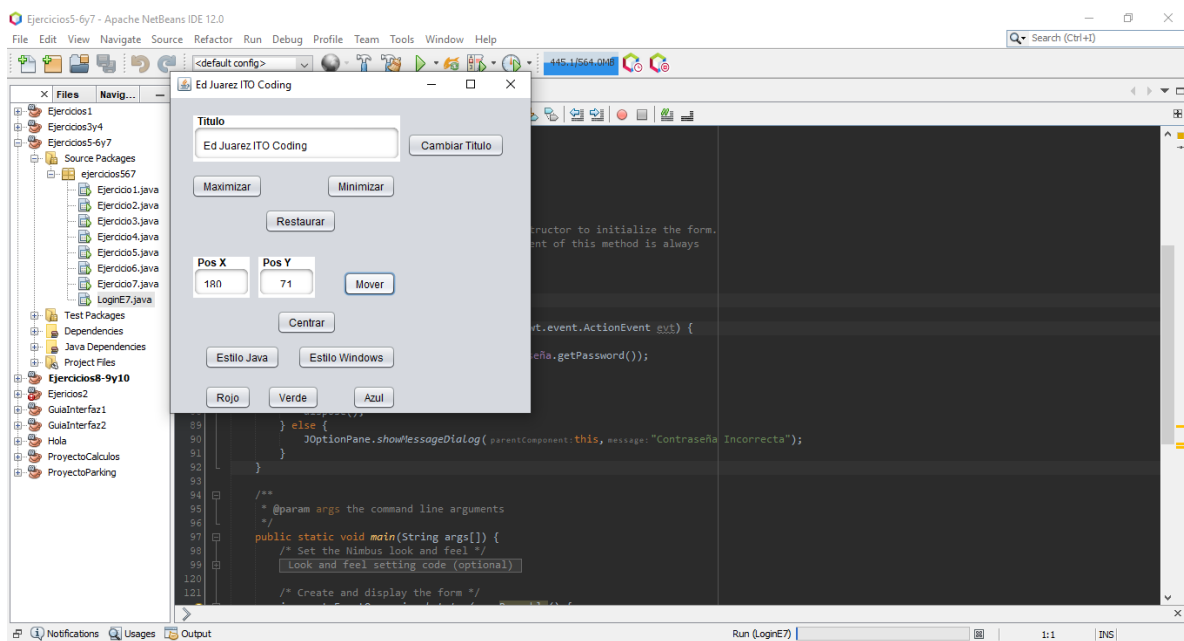
- Si el usuario pulsa el botón “Minimizar”, la ventana se minimizará.



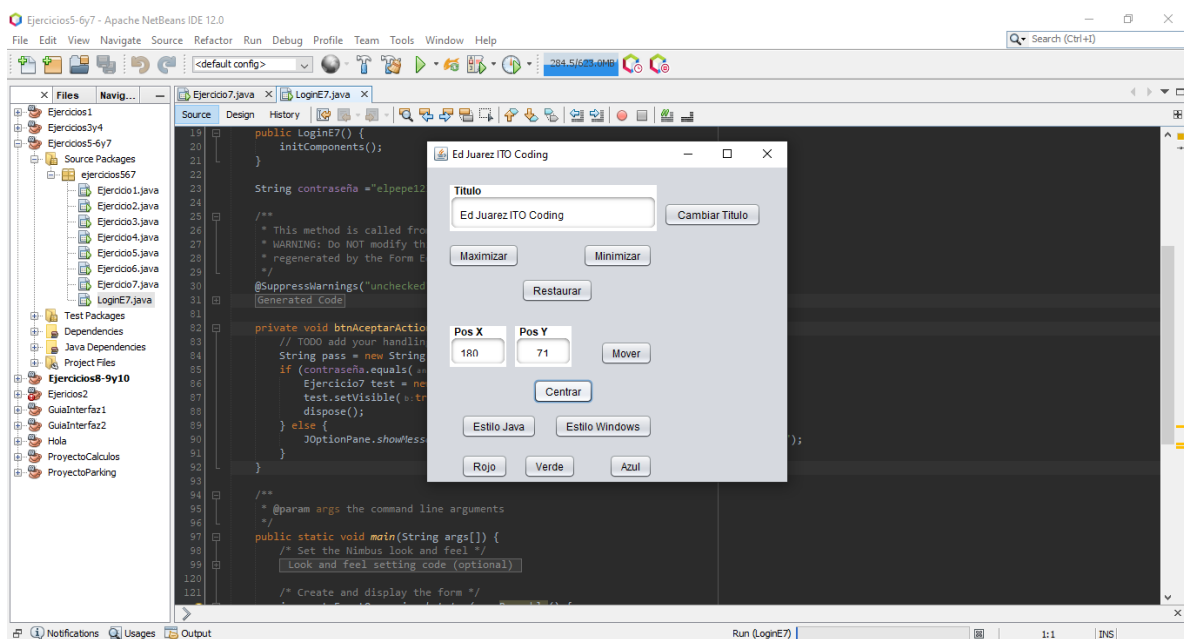
- Si el usuario pulsa el botón “Restaurar”, la ventana se restaurará.



- Si el usuario pulsa el botón “Mover”, la ventana se colocará en la posición de la pantalla que venga indicada por los valores x, y, introducidos respectivamente en los cuadros de texto txtPosicionX y txtPosicionY.



- Si el usuario pulsa el botón “Centrar”, la ventana se colocará en el centro de la pantalla.

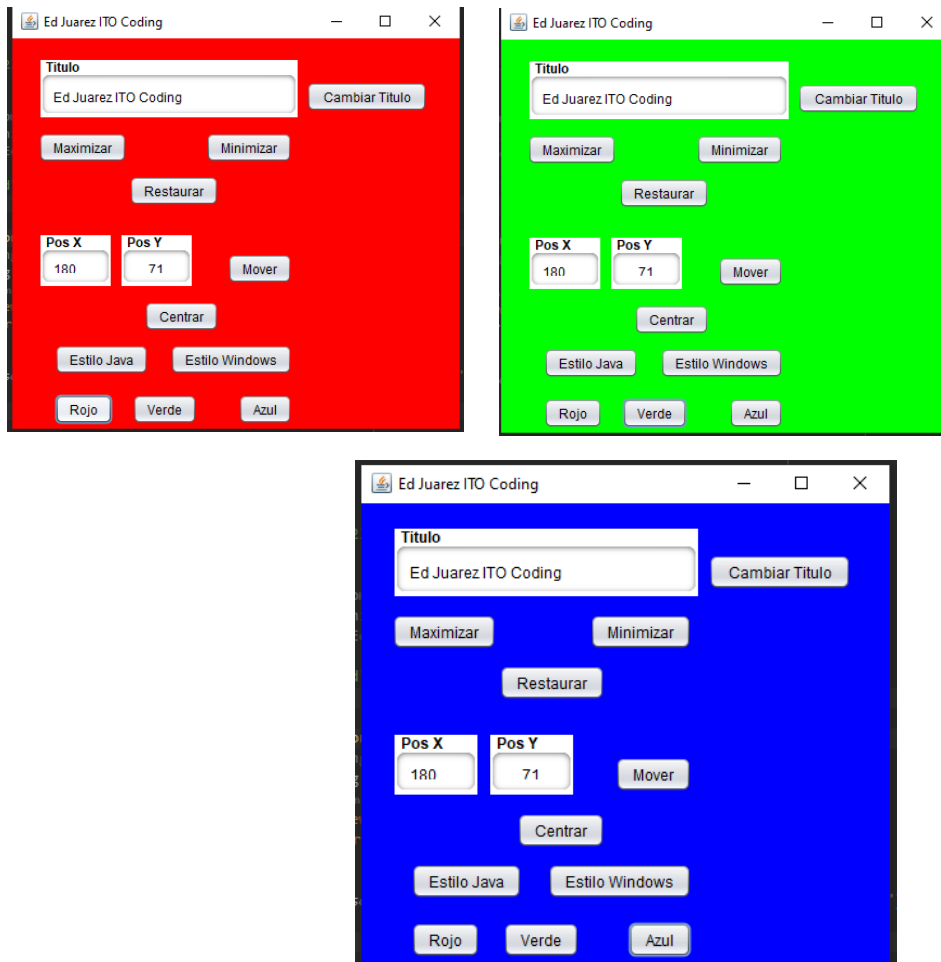


- Si el usuario pulsa el botón “Estilo Java”, entonces toda la ventana cambiará a la visualización Java.

- Si el usuario pulsa el botón “Estilo Windows”, entonces toda la ventana cambiará a la visualización Windows.

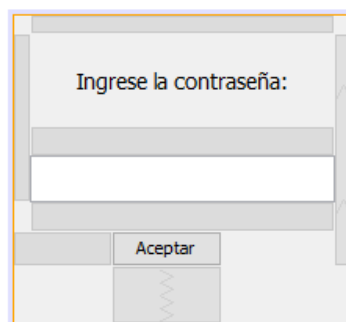
Este si no pude hacerlos con el LookandFeel lo intenté, pero no sé porque no cambiaba, de todas maneras ahí dejaré el código para revisarlo

- Si el usuario pulsa uno de los botones de color, entonces el fondo de la ventana se cambiará al color indicado.



Además, el programa contará con las siguientes características:

- Al iniciarse el programa, este debe pedir una contraseña al usuario. Si el usuario introduce la contraseña correcta, entonces el usuario entrará en el programa. En caso contrario el programa se cerrará.



```
String contraseña = "elpepe1234";

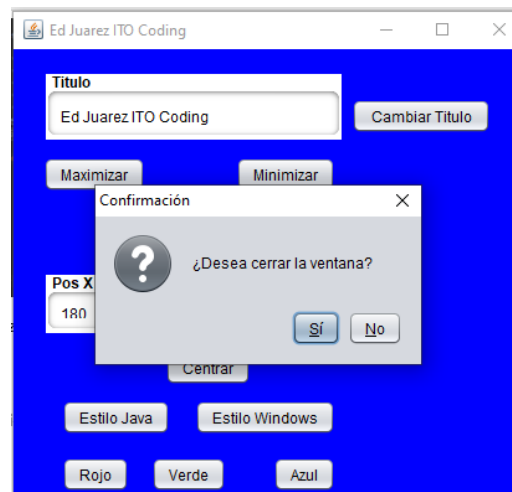
/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
Generated Code

private void btnAceptarActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    String pass = new String( chars: passContraseña.getPassword());
    if (contraseña.equals( anObject: pass)) {
        Ejercicio7 test = new Ejercicio7();
        test.setVisible( b: true);
        dispose();
    } else {
        JOptionPane.showMessageDialog( parentComponent: this, message: "Contraseña Incorrecta");
    }
}
}
```

- Al finalizar el programa, este debe pedir una confirmación al usuario:

¿Desea salir del programa?

Si el usuario responde Sí, entonces el programa finalizará. En caso contrario, el programa seguirá funcionando.



```
private void btnCambiarTituloActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    this.setTitle( title: txtTitulo.getText());
}

private void btnMaximizarActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    this.setExtendedState( state: JFrame.MAXIMIZED_BOTH);
}

private void btnMinimizarActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    this.setSize( width: 440, height: 440);
}

private void btnRestaurarActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    this.setSize( width: 430, height: 400);
}

private void btnMoverActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    int x = Integer.parseInt( s: txtPosicionX.getText());
    int y = Integer.parseInt( s: txtPosicionY.getText());
    this.setLocation(x, y);
}

private void btnCentrarActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    this.setLocation( x: 475, y: 150);
}
```



```

55
56 private void btnWindowsActionPerformed(java.awt.event.ActionEvent evt) {
57     try {
58         JFrame.setDefaultLookAndFeelDecorated( defaultLookAndFeelDecorated:true);
59         JDialog.setDefaultLookAndFeelDecorated( defaultLookAndFeelDecorated:true);
60         UIManager.setLookAndFeel( className:UIManager.getSystemLookAndFeelClassName ());
61     } catch (ClassNotFoundException ex) {
62         Logger.getLogger( name:Ejercicio7.class.getName()).log( level:Level.SEVERE, msg:null, thrown:ex);
63     } catch (InstantiationException ex) {
64         Logger.getLogger( name:Ejercicio7.class.getName()).log( level:Level.SEVERE, msg:null, thrown:ex);
65     } catch (IllegalAccessException ex) {
66         Logger.getLogger( name:Ejercicio7.class.getName()).log( level:Level.SEVERE, msg:null, thrown:ex);
67     } catch (UnsupportedLookAndFeelException ex) {
68         Logger.getLogger( name:Ejercicio7.class.getName()).log( level:Level.SEVERE, msg:null, thrown:ex);
69     }
70 }
71
72 private void btnJavaActionPerformed(java.awt.event.ActionEvent evt) {
73     // TODO add your handling code here:
74     this.setDefaultLookAndFeelDecorated( defaultLookAndFeelDecorated:true);
75 }
76
77 private void btnRojoActionPerformed(java.awt.event.ActionEvent evt) {
78     // TODO add your handling code here:
79     this.getContentPane().setBackground( c:Color.RED);
80 }
81
82 private void btnVerdeActionPerformed(java.awt.event.ActionEvent evt) {
83     // TODO add your handling code here:
84     this.getContentPane().setBackground( c:Color.GREEN);
85 }
86
87 private void btnAzulActionPerformed(java.awt.event.ActionEvent evt) {

```

```

87     // TODO add your handling code here:
88     this.getContentPane().setBackground( c:Color.BLUE);
89 }
90
91 private void btnAzulActionPerformed(java.awt.event.ActionEvent evt) {
92     // TODO add your handling code here:
93     this.getContentPane().setBackground( c:Color.BLUE);
94 }
95
96 private void formWindowClosing(java.awt.event.WindowEvent evt) {
97     // TODO add your handling code here:
98     int option = JOptionPane.showConfirmDialog( parentComponent:null,
99         message:"¿Desea cerrar la ventana?", title:"Confirmación",
100         optionType:JOptionPane.YES_NO_O
101     );
102     if (option == JOptionPane.YES_OPTION) {
103         System.exit( status:0);
104     }
105 }

```