

INSTITUTO MARISTA LA INMACULADA

Alumna:

Lohany Kassandra Márquez Miranda

Cta:

12

Catedrática:

Waleska Deyanira Gutiérrez Arteaga

Asignatura:

Programación

Tema:

Tutorial de Java

Curso:

12 BTP

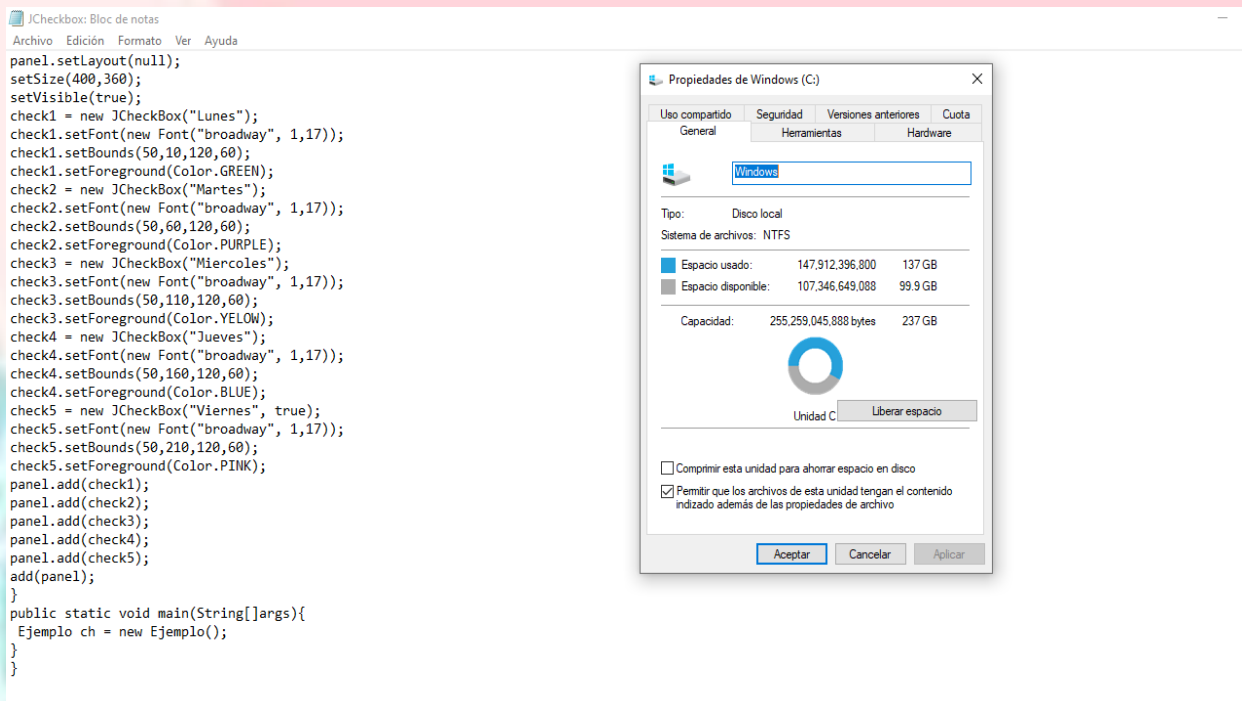
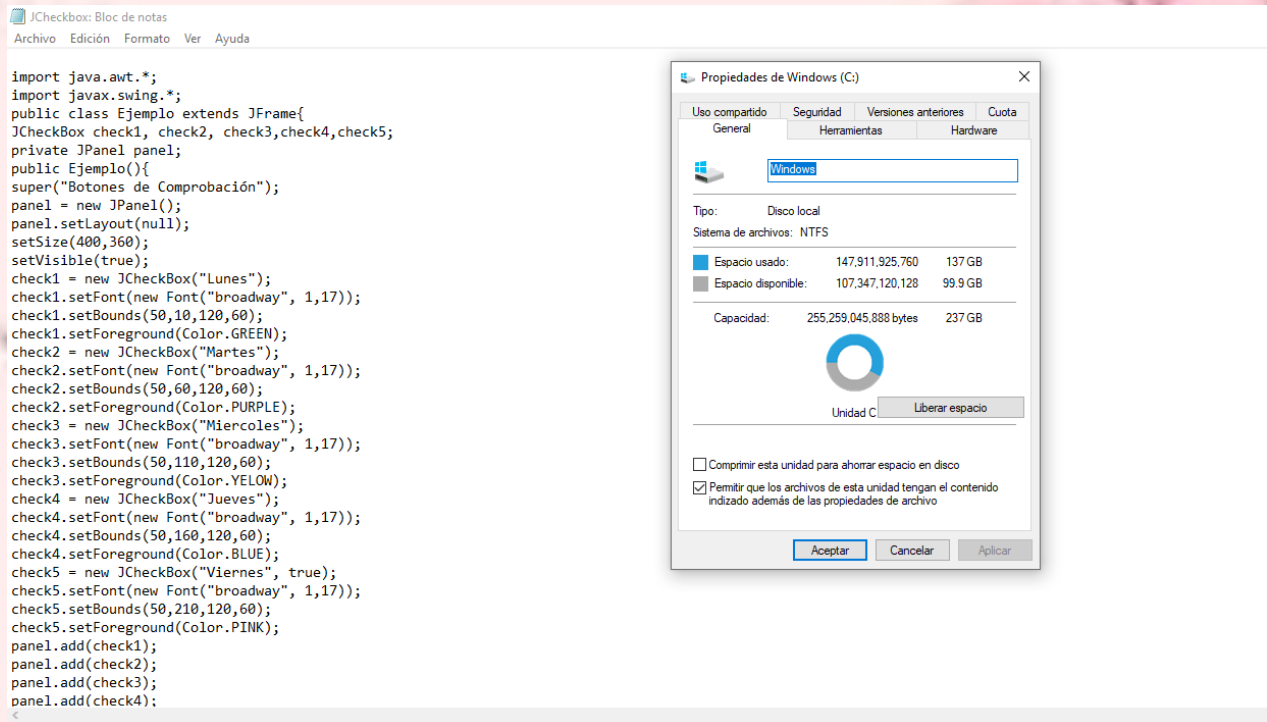
Sección:

"A"

Fecha:

12/10/2021

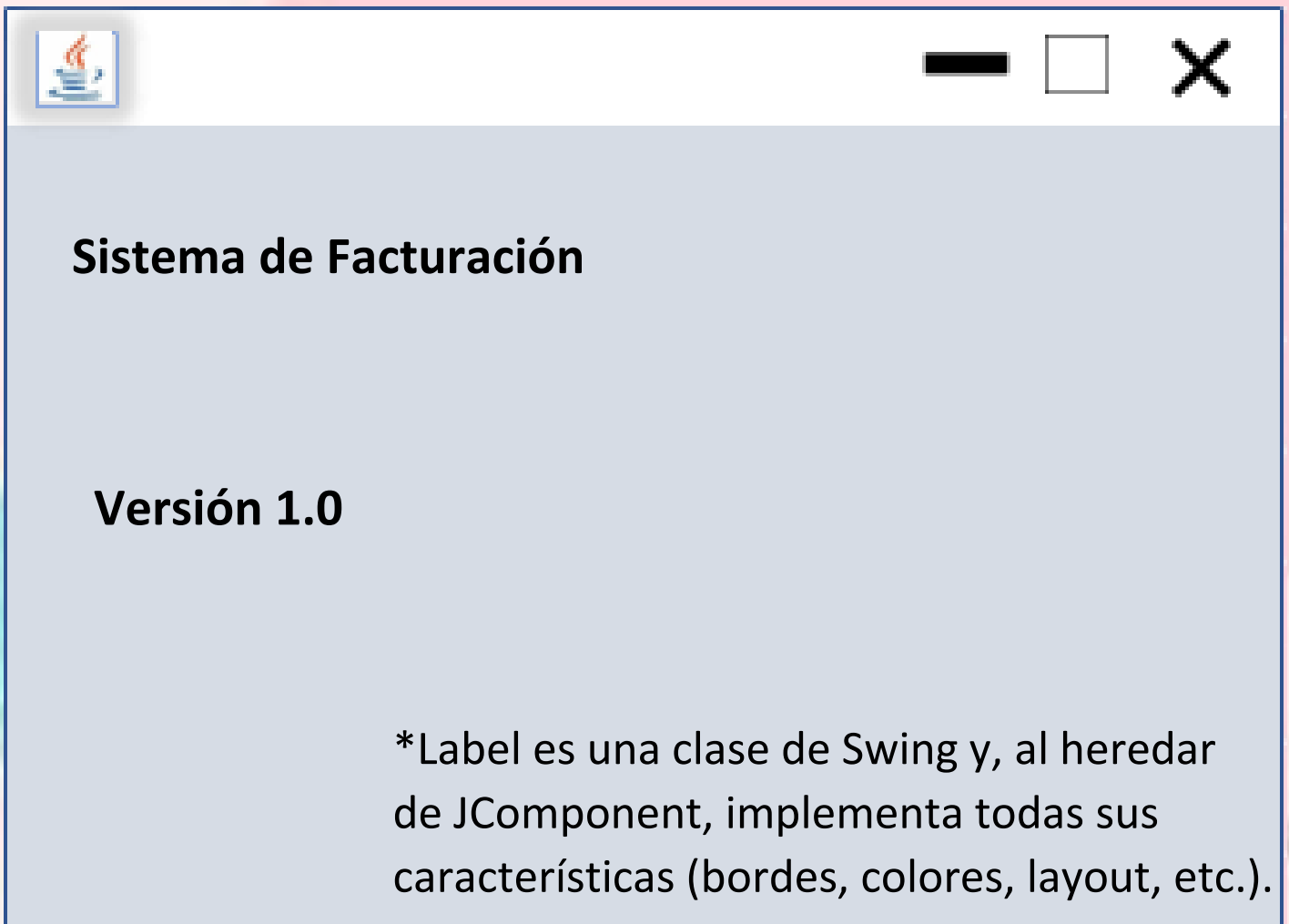
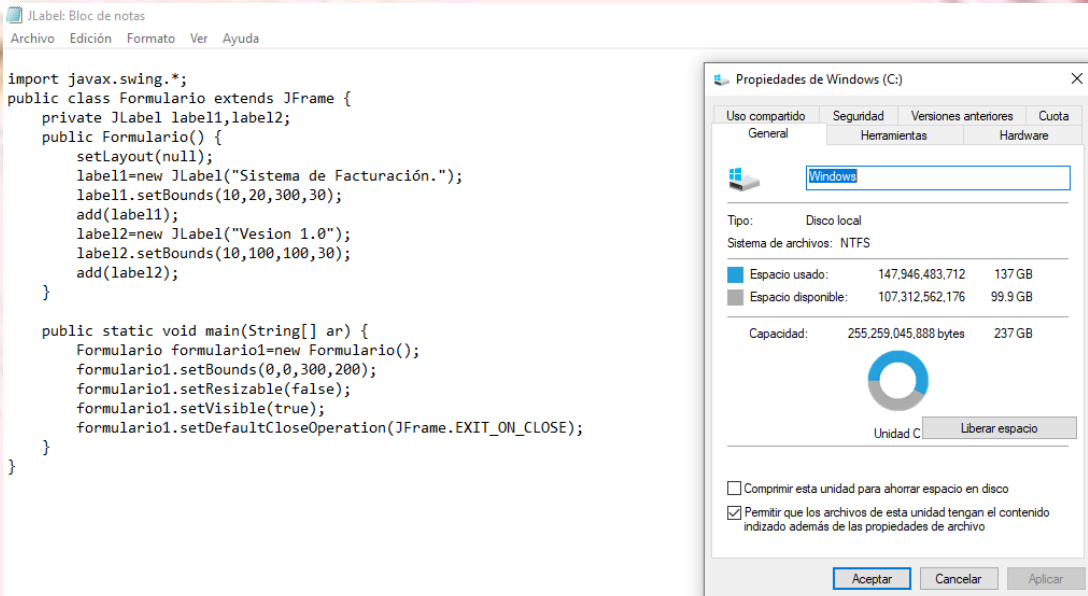
JCheckbox





*La clase JCheckbox proporciona los atributos y métodos para implementar una casilla de verificación que es básicamente un botón de dos estados.

JLabel

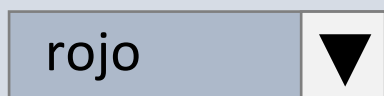
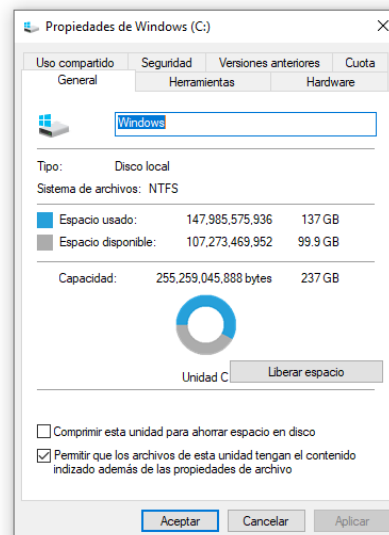


JComboBox

```
JComboBox: Bloc de notas
Archivo Edición Formato Ver Ayuda
import javax.swing.*;
import java.awt.event.*;
public class Formulario extends JFrame implements ItemListener{
    private JComboBox<String> combo1;
    public Formulario() {
        setLayout(null);
        combo1=new JComboBox<String>();
        combo1.setBounds(10,10,80,20);
        add(combo1);
        combo1.addItem("rojo");
        combo1.addItem("verde");
        combo1.addItem("azul");
        combo1.addItem("amarillo");
        combo1.addItem("negro");
        combo1.addItemListener(this);
    }

    public void itemStateChanged(ItemEvent e) {
        if (e.getSource()==combo1) {
            String seleccionado=(String)combo1.getSelectedItem();
            setTitle(seleccionado);
        }
    }

    public static void main(String[] ar) {
        Formulario formulario1=new Formulario();
        formulario1.setBounds(0,0,200,150);
        formulario1.setVisible(true);
        formulario1.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}
```



*El JComboBox nos permite mostrar una caja de selección en la que podemos seleccionar una sola opción.

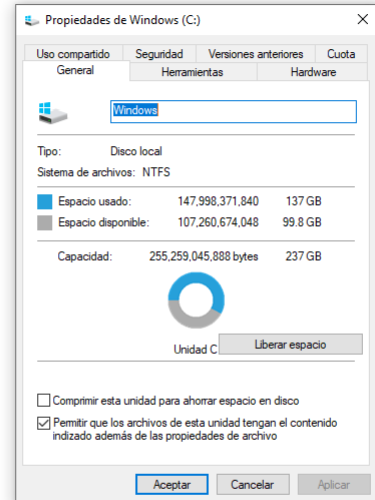
JRadioButton

JRadioButton: Bloc de notas

Archivo Edición Formato Ver Ayuda

```
import javax.swing.*;
import javax.swing.event.*;
public class Formulario extends JFrame implements ChangeListener{
    private JRadioButton radio1,radio2,radio3;
    private ButtonGroup bg;
    public Formulario() {
        setLayout(null);
        bg=new ButtonGroup();
        radio1=new JRadioButton("640*480");
        radio1.setBounds(10,20,100,30);
        radio1.addChangeListener(this);
        add(radio1);
        bg.add(radio1);
        radio2=new JRadioButton("800*600");
        radio2.setBounds(10,70,100,30);
        radio2.addChangeListener(this);
        add(radio2);
        bg.add(radio2);
        radio3=new JRadioButton("1024*768");
        radio3.setBounds(10,120,100,30);
        radio3.addChangeListener(this);
        add(radio3);
        bg.add(radio3);
    }

    public void stateChanged(ChangeEvent e) {
        if (radio1.isSelected()) {
            setSize(640,480);
        }
        if (radio2.isSelected()) {
            setSize(800,600);
        }
        if (radio3.isSelected()) {
            setSize(1024,768);
        }
    }
}
```



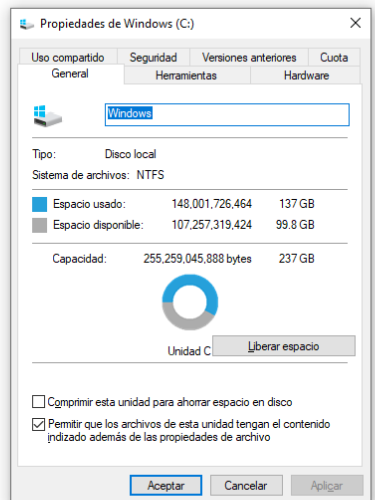
JRadioButton: Bloc de notas

Archivo Edición Formato Ver Ayuda

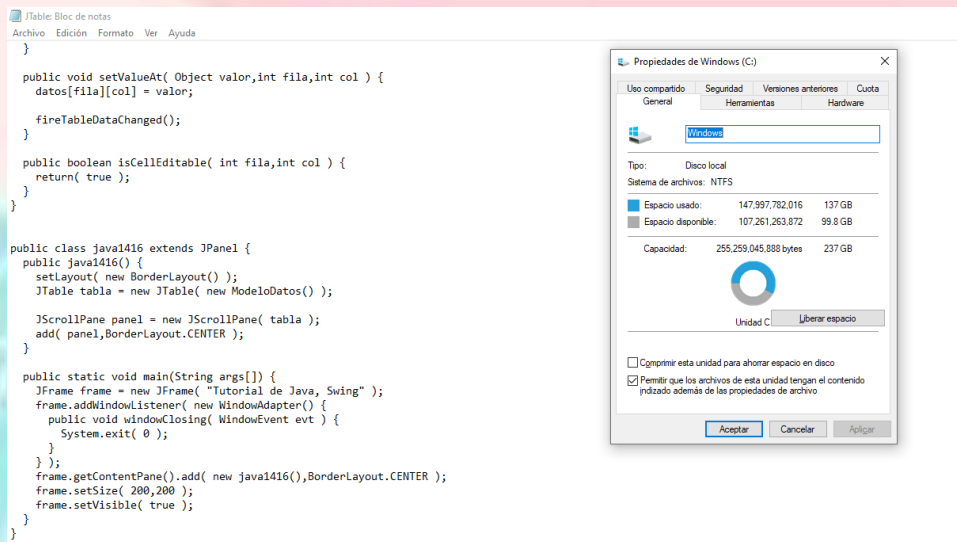
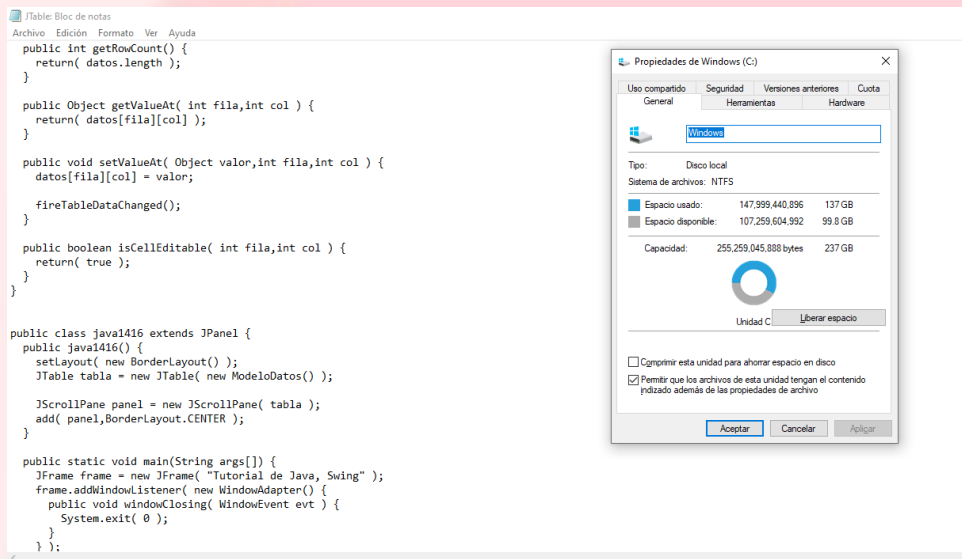
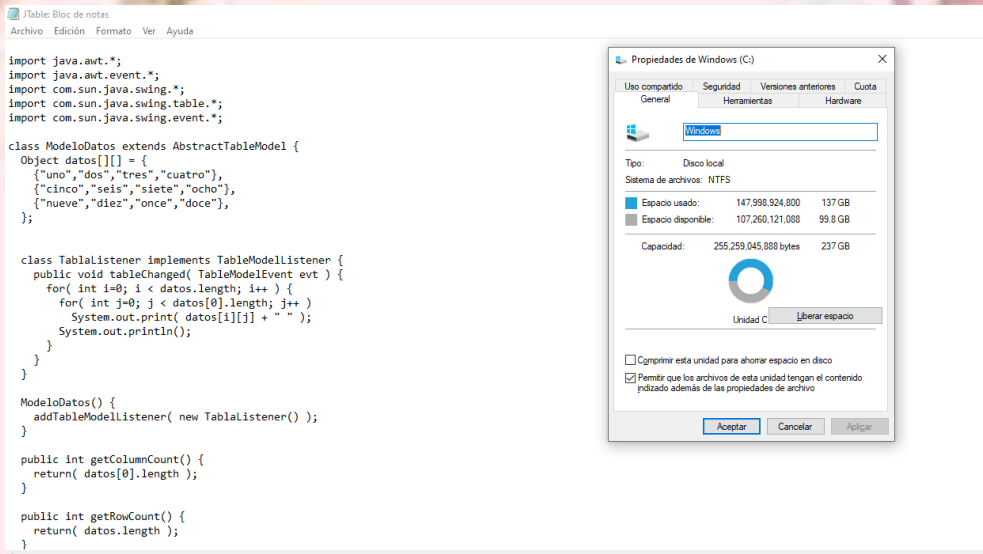
```
radio1.setBounds(10,20,100,30);
radio1.addChangeListener(this);
add(radio1);
bg.add(radio1);
radio2=new JRadioButton("800*600");
radio2.setBounds(10,70,100,30);
radio2.addChangeListener(this);
add(radio2);
bg.add(radio2);
radio3=new JRadioButton("1024*768");
radio3.setBounds(10,120,100,30);
radio3.addChangeListener(this);
add(radio3);
bg.add(radio3);
}

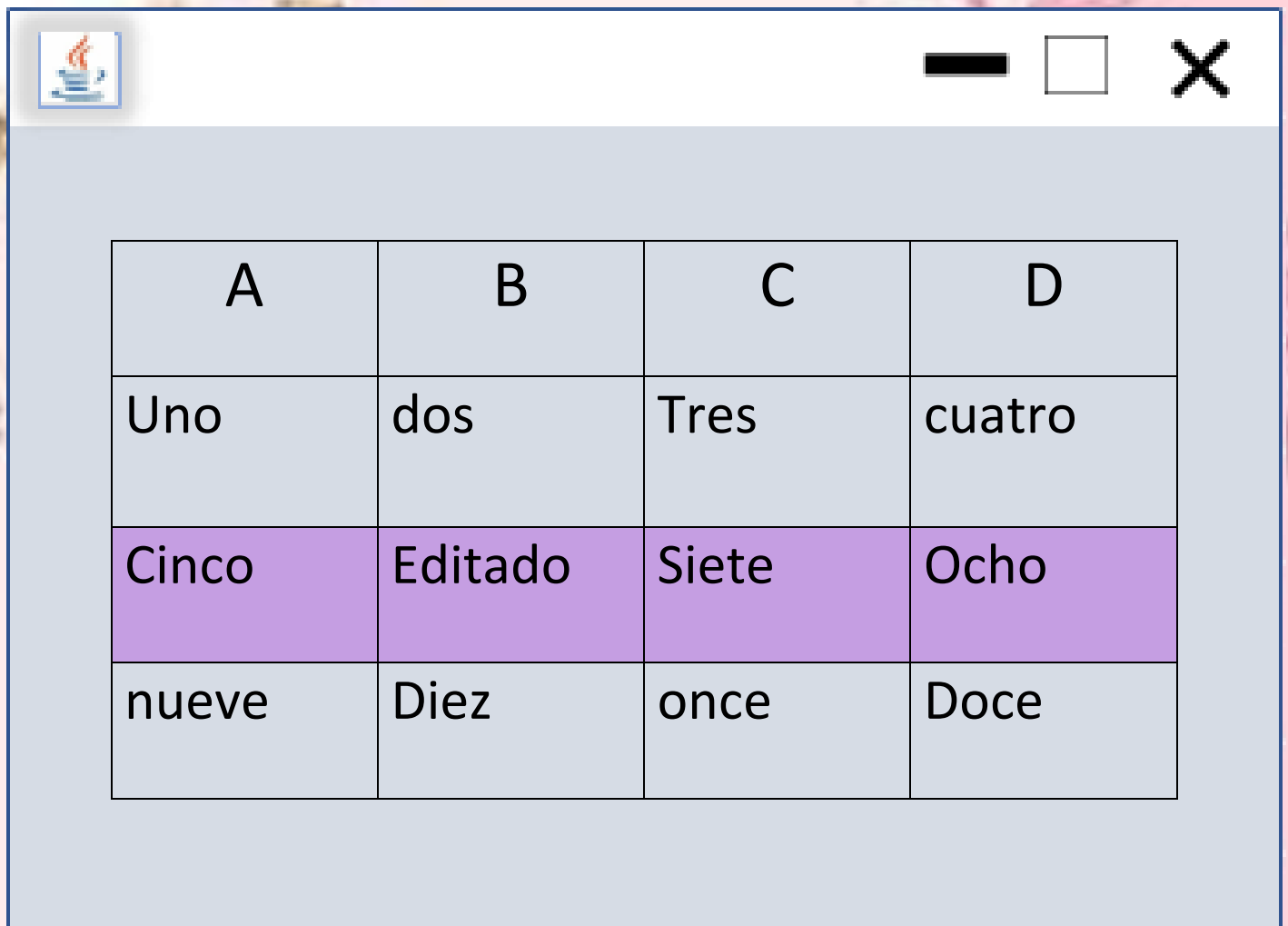
public void stateChanged(ChangeEvent e) {
    if (radio1.isSelected()) {
        setSize(640,480);
    }
    if (radio2.isSelected()) {
        setSize(800,600);
    }
    if (radio3.isSelected()) {
        setSize(1024,768);
    }
}

public static void main(String[] ar) {
    Formulario formulario1=new Formulario();
    formulario1.setBounds(0,0,350,230);
    formulario1.setVisible(true);
    formulario1.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}
}
```



JTable





A	B	C	D
Uno	dos	Tres	cuatro
Cinco	Editado	Siete	Ocho
nueve	Diez	once	Doce

*JTable es una herramienta visual de Java que sirve para poder dibujar tablas, con sus respectivas filas y columnas en donde puedes ingresar el dato que tu desees.

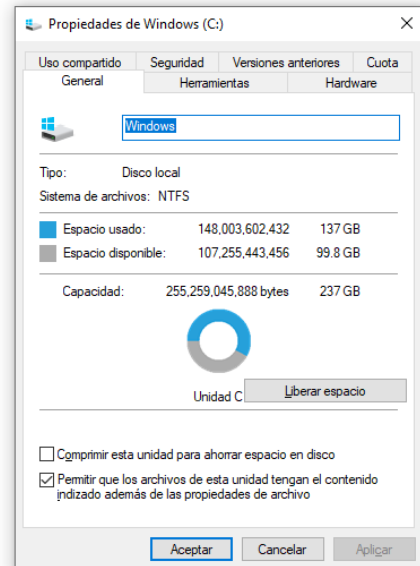
JTextArea

JTextArea: Bloc de notas

Archivo Edición Formato Ver Ayuda

```
import javax.swing.*;
public class Formulario extends JFrame{
    private JTextField textfield1;
    private JTextArea textarea1;
    public Formulario() {
        setLayout(null);
        textfield1=new JTextField();
        textfield1.setBounds(10,10,200,30);
        add(textfield1);
        textarea1=new JTextArea();
        textarea1.setBounds(10,50,400,300);
        add(textarea1);
    }

    public static void main(String[] ar) {
        Formulario formulario1=new Formulario();
        formulario1.setBounds(0,0,540,400);
        formulario1.setVisible(true);
        formulario1.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}
```



Lohamarquez@gmail.com

Linea1

Linea2

Linea3

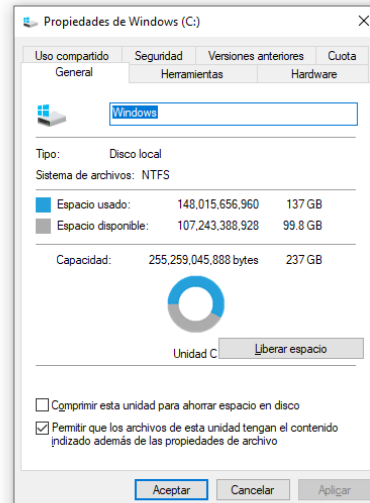
*JTextArea es un componente básico de la librería Swing de Java y su función principal es la de capturar y mostrar texto ingresado desde teclado por el usuario.

JSlider

JSlider: Bloc de notas

Archivo Edición Formato Ver Ayuda

```
import javax.swing.*.*;
import javax.swing.event.*;
import java.awt.*.*;
public class JavaExampleDynamicIcon
{
    public static void main(String[] as)
    {
        final JSlider Wdth = new JSlider(JSlider.HORIZONTAL, 1, 150, 75);
        final JSlider Hght = new JSlider(JSlider.VERTICAL, 1, 150, 75);
        class DynamicIcon implements Icon
        {
            public int getIconWidth()
            {
                return Wdth.getValue();
            }
            public int getIconHeight()
            {
                return Hght.getValue();
            }
            public void paintIcon(Component Cmpnt, Graphics gr, int a, int b)
            {
                gr.fillRect(a, b, getIconWidth(), getIconHeight(), true);
            }
        };
        Icon Icn = new DynamicIcon();
        final JLabel LblDymnc = new JLabel(Icn);
        class Updater implements ChangeListener
        {
            public void stateChanged(ChangeEvent Evnt)
            {
                LblDymnc.repaint();
            }
        };
        Updater Updtr = new Updater();
        Wdth.addChangeListener(Updtr);
        Hght.addChangeListener(Updtr);
    }
}
```



JSlider: Bloc de notas

Archivo Edición Formato Ver Ayuda

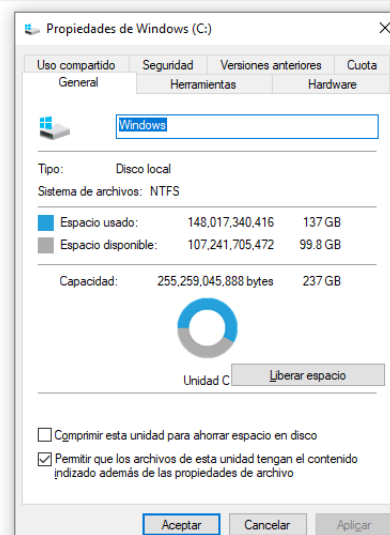
```
public int getIconWidth()
{
    return Wdth.getValue();
}

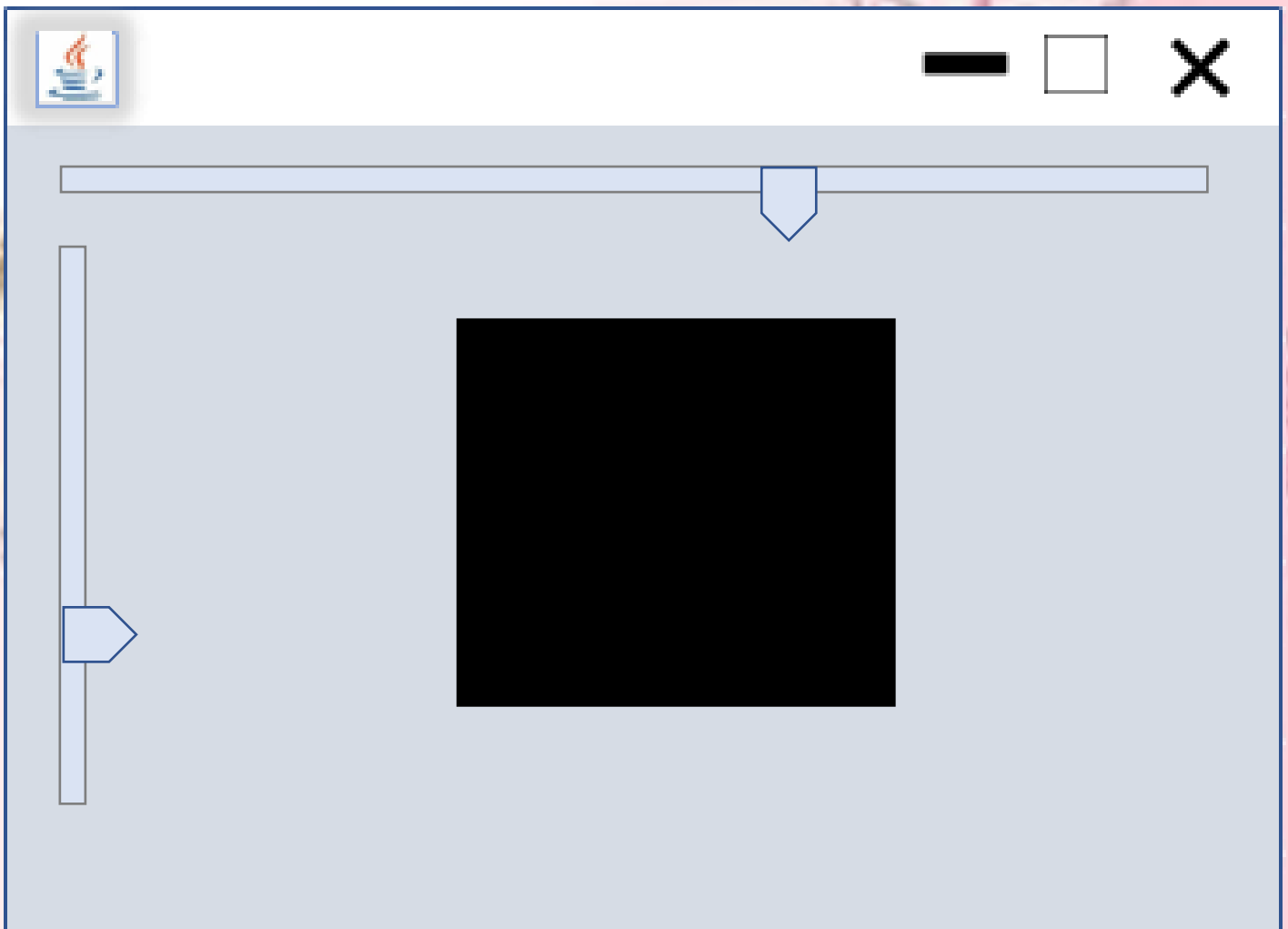
public int getIconHeight()
{
    return Hght.getValue();
}

public void paintIcon(Component Cmpnt, Graphics gr, int a, int b)
{
    gr.fillRect(a, b, getIconWidth(), getIconHeight(), true);
}

};
Icon Icn = new DynamicIcon();
final JLabel LblDymnc = new JLabel(Icn);
class Updater implements ChangeListener
{
    public void stateChanged(ChangeEvent Evnt)
    {
        LblDymnc.repaint();
    }
};

Updater Updtr = new Updater();
Wdth.addChangeListener(Updtr);
Hght.addChangeListener(Updtr);
JFrame Frm = new JFrame();
Container Cntnr = Frm.getContentPane();
Cntnr.setLayout(new BorderLayout());
Cntnr.add(Wdth, BorderLayout.NORTH);
Cntnr.add(Hght, BorderLayout.WEST);
Cntnr.add(LblDymnc, BorderLayout.CENTER);
Frm.setSize(210,210);
Frm.setVisible(true);
}
```





*Se utiliza un JSlider para permitir que el usuario introduzca un valor numérico limitado por un valor máximo y un valor mínimo.

JToolBar

```
JToolBar: Bloc de notas
Archivo Edición Formato Ver Ayuda

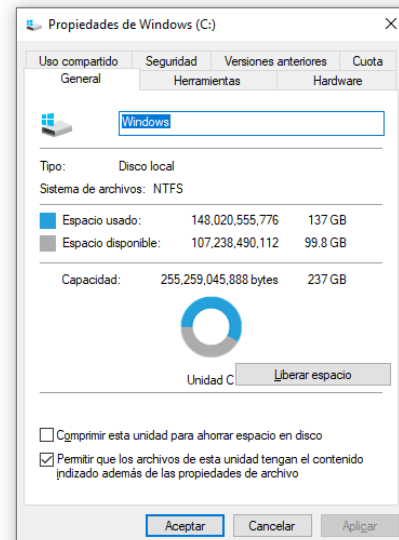
import java.awt.*;
import javax.swing.*;

public class Main
{
    public static void main(final String args[])
    {
        JFrame f = new JFrame("JToolBar Example");

        JToolBar tb = new JToolBar();
        tb.setRollover(true);

        tb.add(new JButton("Button 1"));
        tb.addSeparator();
        tb.add(new JButton("Button 2"));
        tb.add(
            new JComboBox(
                new String[] { "item1", "item2", "item3", "item4" }
            )
        );
        Container c = f.getContentPane();
        // add the toolbar to the container
        c.add(tb, BorderLayout.NORTH);
        JTextArea text = new JTextArea();
        JScrollPane p = new JScrollPane(text);
        c.add(p, BorderLayout.EAST);

        f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        f.setSize(400, 200);
        f.setVisible(true);
    }
}
```



Button1

Button2

Item1



*Esta clase implemente una barra de herramientas, formada normalmente por botones o controles que incluyen iconos y que aparecen organizados como una fila o una columna dependiendo de la zona de la pantalla donde se coloque.

JSeparator

```
JToolBar: Bloc de notas
Archivo  Edición  Formato  Ver  Ayuda

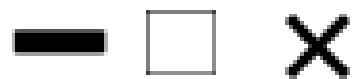
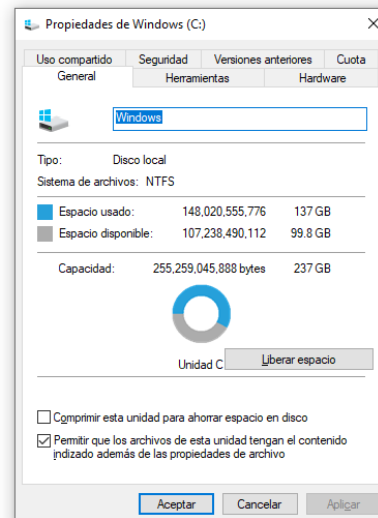
import java.awt.*;
import javax.swing.*;

public class Main
{
    public static void main(final String args[])
    {
        JFrame f = new JFrame("JToolBar Example");

        JToolBar tb = new JToolBar();
        tb.setRollover(true);

        tb.add(new JButton("Button 1"));
        tb.addSeparator();
        tb.add(new JButton("Button 2"));
        tb.add(
            new JComboBox(
                new String[] { "item1", "item2", "item3", "item4" }
            )
        );
        Container c = f.getContentPane();
        // add the toolbar to the container
        c.add(tb, BorderLayout.NORTH);
        JTextArea text = new JTextArea();
        JScrollPane p = new JScrollPane(text);
        c.add(p, BorderLayout.EAST);

        f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        f.setSize(400, 200);
        f.setVisible(true);
    }
}
```



File

New

Open

*El JSeparator clase proporciona una línea divisoria horizontal o vertical o espacio vacío. Se usa más comúnmente en los menús y barras de herramientas.