

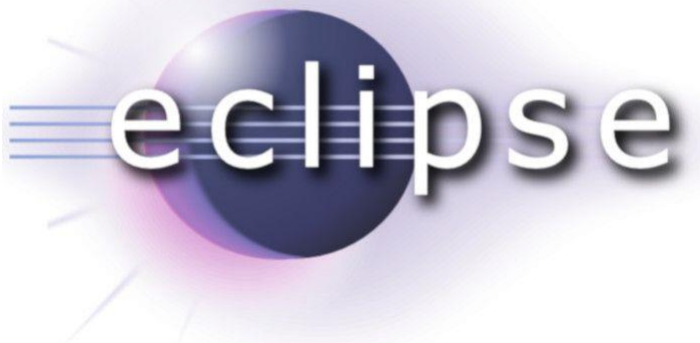


## CURSO INICIACIÓN JAVA V

Tutoriales de pildorasinformaticas

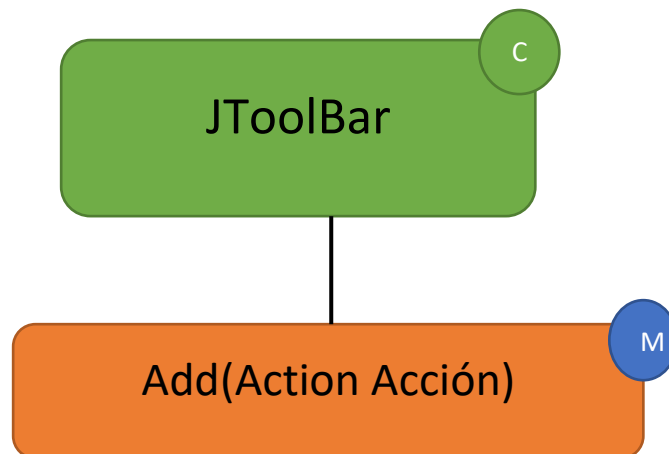
Descripción breve

Curso introductorio de Java por pildorasinformaticas.



Pere Manel Verdugo Zamora  
pereverdugo@gmail.com

## Aplicaciones gráficas. Barra de herramientas I (Vídeo 111)



Vamos a escribir el siguiente código:

```
package graficos;

import java.awt.*;
import java.awt.event.ActionEvent;

import javax.swing.*;

public class Barra_Herramientas {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Marco_Barra mimarco=new Marco_Barra();
        mimarco.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        mimarco.setVisible(true);
    }

}

class Marco_Barra extends JFrame{
    public Marco_Barra() {
        setTitle("Marco con Barra");
        setBounds(500,300,600,450);
        lamina = new JPanel();
        add(lamina);
        Action accionAzul=new AccionColor("Azul", new
        ImageIcon("src/graficos/bolaAzul.png"), Color.BLUE);
        Action accionAmarillo=new AccionColor("Amarillo", new
        ImageIcon("src/graficos/bolaAmarilla.png"), Color.YELLOW);
        Action accionRojo=new AccionColor("Rojo", new
        ImageIcon("src/graficos/bolaRoja.png"), Color.RED);
        Action accionSalir=new AbstractAction("Salir", new
        ImageIcon("src/graficos/salir.png")){

            @Override
            public void actionPerformed(ActionEvent e) {
                // TODO Auto-generated method stub
            }
        }
    }
}
```

```

        System.exit(0);
    }

};

JMenu menu=new JMenu("Color");
menu.add(accionAzul);
menu.add(accionAmarillo);
menu.add(accionRojo);
JMenuBar barra_menus=new JMenuBar();
barra_menus.add(menu);
setJMenuBar(barra_menus);

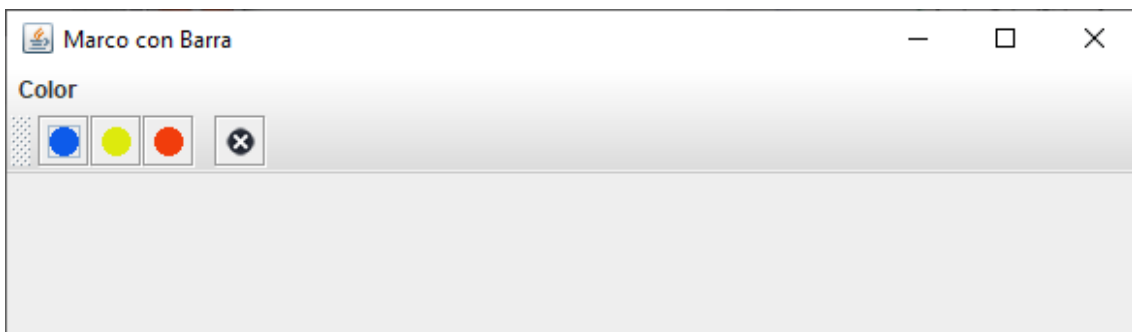
//Contrucción de la barra de herramientas
JToolBar barra=new JToolBar();
barra.add(accionAzul);
barra.add(accionAmarillo);
barra.add(accionRojo);
barra.addSeparator();
barra.add(accionSalir);
add(barra, BorderLayout.NORTH);
}

private class AccionColor extends AbstractAction{
    public AccionColor(String nombre, Icon icono, Color c) {
        putValue(Action.NAME, nombre);
        putValue(Action.SMALL_ICON, icono);
        putValue(Action.SHORT_DESCRIPTION, "Color de fondo..." +
nombre);
        putValue("Color",c);
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        // TODO Auto-generated method stub
        Color c=(Color) getValue("Color");
        lamina.setBackground(c);
    }
}
private JPanel lamina;
}
}

```

Este será el resultado:



Desde esta barra puedes cambiar de color la lámina, así como salir de la aplicación.



## Aplicaciones gráficas. Barra de herramientas II. (Vídeo 112)

Partiendo del proyecto procesador de textos vamos a realizar una barra de herramientas.

---

```
package graficos;
import java.awt.BorderLayout;
import java.awt.Font;
import java.awt.event.*;
import java.awt.font.TextAttribute;
import javax.swing.*;
import javax.swing.text.StyledEditorKit;

public class Procesador_II {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        MenuProcesador_II mimarco=new MenuProcesador_II();
        mimarco.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}

class MenuProcesador_II extends JFrame{
    public MenuProcesador_II() {
        setBounds(500,300,550,400);
        LaminaProcesador_II milamina=new LaminaProcesador_II();
        add(milamina);
        setVisible(true);
    }
}

class LaminaProcesador_II extends JPanel{
    public LaminaProcesador_II() {
        setLayout(new BorderLayout());
        JPanel laminamenu=new JPanel();
        JMenuBar mibarra=new JMenuBar();

        fuente=new JMenu("Fuente");
        estilo=new JMenu("Estilo");
        tamagno=new JMenu("Tamaño");

        configura_menu("Arial","fuente","Arial", 9,10,"");
        configura_menu("Courier","fuente","Courier", 9,10,"");
        configura_menu("Verdana","fuente","Verdana", 9,10,"");

        configura_menu("Negrita","estilo","",Font.BOLD,1,"src/graficos/negrita
.png");
        configura_menu("Cursiva","estilo","",Font.ITALIC,1,"src/graficos/cursi
va.png");

        ButtonGroup tamagno_letra=new ButtonGroup();
        JRadioButtonMenuItem doce=new JRadioButtonMenuItem("12");
        JRadioButtonMenuItem dieciseis=new JRadioButtonMenuItem("16");
        JRadioButtonMenuItem veinte=new JRadioButtonMenuItem("20");
        JRadioButtonMenuItem veinticuatro=new JRadioButtonMenuItem("24");

        veinticuatro.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_D,Input
Event.CTRL_DOWN_MASK));

        tamagno_letra.add(doce);
        tamagno_letra.add(dieciseis);
        tamagno_letra.add(veinte);
    }
}
```

```

        tamagno_letra.add(veinticuatro);
        doce.addActionListener(new
StyledEditorKit.FontSizeAction("cambia_tamaño", 12));
        dieciseis.addActionListener(new
StyledEditorKit.FontSizeAction("cambia_tamaño", 16));
        veinte.addActionListener(new
StyledEditorKit.FontSizeAction("cambia_tamaño", 20));
        veinticuatro.addActionListener(new
StyledEditorKit.FontSizeAction("cambia_tamaño", 24));
        tamagno.add(doce);
        tamagno.add(dieciseis);
        tamagno.add(veinte);
        tamagno.add(veinticuatro);

        mibarra.add(fuente);
        mibarra.add(estilo);
        mibarra.add(tamagno);

        laminamenu.add(mibarra);
        add(laminamenu, BorderLayout.NORTH);

        miarea=new JTextPane();
        add(miarea, BorderLayout.CENTER);

        JPopupMenu emergente=new JPopupMenu();
        JMenuItem negritaE=new JMenuItem("Negrita");
        JMenuItem cursivaE=new JMenuItem("Cursiva");

        emergente.add(negritaE);
        emergente.add(cursivaE);

        miarea.setComponentPopupMenu(emergente);

```

```

//-----
JToolBar barra = new JToolBar();
JButton negritaBarra =new JButton(new
ImageIcon("src/graficos/negrita.png"));
JButton cursivaBarra =new JButton(new
ImageIcon("src/graficos/cursiva.png"));
JButton subraBarra = new JButton(new
ImageIcon("src/graficos/subrayado.png"));
negritaBarra.addActionListener(new StyledEditorKit.BoldAction());
cursivaBarra.addActionListener(new StyledEditorKit.ItalicAction());
subraBarra.addActionListener(new StyledEditorKit.UnderlineAction() );
barra.add(negritaBarra);
barra.add(cursivaBarra);
barra.add(subraBarra);
barra.setOrientation(1);
add(barra, BorderLayout.WEST);
//-----

```

```

negritaE.addActionListener(new StyledEditorKit.BoldAction());
cursivaE.addActionListener(new StyledEditorKit.ItalicAction());

```

```

}

```

```

public void configura_menu(String rotulo, String menu, String
tipo_letra, int estilos, int tam, String ruta_icono) {

```

```

        JMenuItem elem_menu=new JMenuItem(rotulo, new
ImageIcon(ruta_icono));
        if(menu=="fuente") {
            fuente.add(elem_menu);
            if(tipo_letra=="Arial") {
                elem_menu.addActionListener(new
StyledEditorKit.FontFamilyAction("cambia_letra", "Arial"));
            }
            else if(tipo_letra=="Courier") {
                elem_menu.addActionListener(new
StyledEditorKit.FontFamilyAction("cambia_letra", "Courier"));
            }
            else if(tipo_letra=="Verdana") {
                elem_menu.addActionListener(new
StyledEditorKit.FontFamilyAction("cambia_letra", "Verdana"));
            }
        }else if(menu=="estilo"){
            estilo.add(elem_menu);
            if(estilos==Font.BOLD) {

                elem_menu.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_N,
InputEvent.CTRL_DOWN_MASK));

                elem_menu.addActionListener(new
StyledEditorKit.BoldAction());

                }else if(estilos==Font.ITALIC) {

                elem_menu.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_K,
InputEvent.CTRL_DOWN_MASK));

                elem_menu.addActionListener(new
StyledEditorKit.ItalicAction());

            }
        }
        else if(menu=="tamaño") {
            tamagno.add(elem_menu);

            elem_menu.addActionListener(new
StyledEditorKit.FontSizeAction("cambia_tamaño", tam));
        }
    }
    JTextPane miarea;
    JMenu fuente, estilo, tamagno;
    Font letras;
}

```

---

Agregamos el código que está en el recuadro, este será el resultado:





## Aplicaciones gráficas. Barras de herramientas III. (Vídeo 113)

```
//-----  
    JToolBar barra = new JToolBar();  
    JButton negritaBarra =new JButton(new  
ImageIcon("src/graficos/negrita.png"));  
    JButton cursivaBarra =new JButton(new  
ImageIcon("src/graficos/cursiva.png"));  
    JButton subraBarra = new JButton(new  
ImageIcon("src/graficos/subrayado.png"));  
    JButton azulBarra = new JButton(new  
ImageIcon("src/graficos/bolaAzul.png"));  
    JButton amarilloBarra = new JButton(new  
ImageIcon("src/graficos/bolaAmarilla.png"));  
    JButton rojoBarra = new JButton(new  
ImageIcon("src/graficos/bolaRoja.png"));  
    negritaBarra.addActionListener(new StyledEditorKit.BoldAction());  
    cursivaBarra.addActionListener(new StyledEditorKit.ItalicAction());  
    subraBarra.addActionListener(new StyledEditorKit.UnderlineAction());  
    azulBarra.addActionListener(new StyledEditorKit.ForegroundAction("Pone  
azul", Color.BLUE));  
    amarilloBarra.addActionListener(new  
StyledEditorKit.ForegroundAction("Pone amarillo", Color.YELLOW));  
    rojoBarra.addActionListener(new StyledEditorKit.ForegroundAction("Pone  
rojo", Color.RED));  
    barra.add(negritaBarra);  
    barra.add(cursivaBarra);  
    barra.add(subraBarra);  
    barra.add(azulBarra);  
    barra.add(amarilloBarra);  
    barra.add(rojoBarra);  
    barra.setOrientation(1);  
    add(barra, BorderLayout.WEST);  
//-----
```

Agrega el texto que está enmarcado.

Este será el resultado:



Ahora queremos agregar las alineaciones, izquierda, centro, derecha y justificado.

---

```
//-----  
    JToolBar barra = new JToolBar();  
    JButton negritaBarra = new JButton(new  
ImageIcon("src/graficos/negrita.png"));  
    JButton cursivaBarra = new JButton(new  
ImageIcon("src/graficos/cursiva.png"));  
    JButton subraBarra = new JButton(new  
ImageIcon("src/graficos/subrayado.png"));  
    JButton azulBarra = new JButton(new  
ImageIcon("src/graficos/bolaAzul.png"));  
    JButton amarilloBarra = new JButton(new  
ImageIcon("src/graficos/bolaAmarilla.png"));  
    JButton rojoBarra = new JButton(new  
ImageIcon("src/graficos/bolaRoja.png"));  
  
    JButton izquierdaBarra = new JButton(new  
ImageIcon("src/graficos/izquierda.png"));  
    JButton centroBarra = new JButton(new  
ImageIcon("src/graficos/centrado.png"));  
    JButton derechaBarra = new JButton(new  
ImageIcon("src/graficos/derecha.png"));  
    JButton justificadoBarra = new JButton(new  
ImageIcon("src/graficos/justificado.png"));  
  
    negritaBarra.addActionListener(new StyledEditorKit.BoldAction());  
    cursivaBarra.addActionListener(new StyledEditorKit.ItalicAction());  
    subraBarra.addActionListener(new StyledEditorKit.UnderlineAction());  
    azulBarra.addActionListener(new StyledEditorKit.ForegroundAction("Pone  
azul", Color.BLUE));  
    amarilloBarra.addActionListener(new  
StyledEditorKit.ForegroundAction("Pone amarillo", Color.YELLOW));  
    rojoBarra.addActionListener(new StyledEditorKit.ForegroundAction("Pone  
rojo", Color.RED));  
  
    izquierdaBarra.addActionListener(new  
StyledEditorKit.AlignmentAction("izquierda", 0));  
    centroBarra.addActionListener(new  
StyledEditorKit.AlignmentAction("centrado", 1));  
    derechaBarra.addActionListener(new  
StyledEditorKit.AlignmentAction("derecha", 2));  
    justificadoBarra.addActionListener(new  
StyledEditorKit.AlignmentAction("justificaco", 3));  
  
    barra.add(negritaBarra);  
    barra.add(cursivaBarra);  
    barra.add(subraBarra);  
    barra.add(azulBarra);  
    barra.add(amarilloBarra);  
    barra.add(rojoBarra);  
  
    barra.add(izquierdaBarra);  
    barra.add(centroBarra);  
    barra.add(derechaBarra);  
    barra.add(justificadoBarra);  
  
    barra.setOrientation(1);  
    add(barra, BorderLayout.WEST);
```

//-----

Agrega el correspondiente texto que está enmarcado en el código.

Este será el resultado:



Una vez escrito el texto prueba las respectivas alineaciones.



## Aplicaciones gráficas. Barras de herramientas IV. (Vídeo 114)

Vamos a modificar el código anterior para simplificarlo.

```
84  /*JToolBar barra = new JToolBar();
85  JButton negritaBarra = new JButton(new ImageIcon("src/graficos/negrita.png"));
86  JButton cursivaBarra = new JButton(new ImageIcon("src/graficos/cursiva.png"));
87  JButton subraBarra = new JButton(new ImageIcon("src/graficos/subrayado.png"));
88  JButton azulBarra = new JButton(new ImageIcon("src/graficos/bolaAzul.png"));
89  JButton amarilloBarra = new JButton(new ImageIcon("src/graficos/bolaAmarilla.png"));
90  JButton rojoBarra = new JButton(new ImageIcon("src/graficos/bolaRoja.png"));
91
92  JButton izquierdaBarra = new JButton(new ImageIcon("src/graficos/izquierda.png"));
93  JButton centroBarra = new JButton(new ImageIcon("src/graficos/centrado.png"));
94  JButton derechaBarra = new JButton(new ImageIcon("src/graficos/derecha.png"));
95  JButton justificadoBarra = new JButton(new ImageIcon("src/graficos/justificado.png"));
96
97  negritaBarra.addActionListener(new StyledEditorKit.BoldAction());
98  cursivaBarra.addActionListener(new StyledEditorKit.ItalicAction());
99  subraBarra.addActionListener(new StyledEditorKit.UnderlineAction());
100 azulBarra.addActionListener(new StyledEditorKit.ForegroundAction("Pone azul", Color.BLUE));
101 amarilloBarra.addActionListener(new StyledEditorKit.ForegroundAction("Pone amarillo", Color.YELLOW));
102 rojoBarra.addActionListener(new StyledEditorKit.ForegroundAction("Pone rojo", Color.RED));
103
104 izquierdaBarra.addActionListener(new StyledEditorKit.AlignmentAction("izquierda", 0));
105 centroBarra.addActionListener(new StyledEditorKit.AlignmentAction("centrado", 1));
106 derechaBarra.addActionListener(new StyledEditorKit.AlignmentAction("derecha", 2));
107 justificadoBarra.addActionListener(new StyledEditorKit.AlignmentAction("justificado", 3));
108
109 barra.add(negritaBarra);
110 barra.add(cursivaBarra);
111 barra.add(subraBarra);
112 barra.add(azulBarra);
113 barra.add(amarilloBarra);
114 barra.add(rojoBarra);
115
116 barra.add(izquierdaBarra);
117 barra.add(centroBarra);
118 barra.add(derechaBarra);
119 barra.add(justificadoBarra);*/
```

Comentamos todo este código que al final eliminaremos.

```
166  JTextPane miarea;
167  JMenu fuente, estilo, tamagno;
168  Font letras;
169  JButton negritaBarra, cursivaBarra;
170  JButton subraBarra, azulBarra, rojoBarra, amarilloBarra;
171  JButton a_izquierda, a_centrado, a_derecha, a_justificado;
172  JToolBar barra;
173 }
```

Al final del código sin salir de la última llave definimos estas variables.

Nos situamos a fuera del constructor de la clase

```
class LaminaProcesador_II extends JPanel{
```

Escribimos el siguiente método:

```
public JButton configura_barra(String ruta) {
    JButton boton=new JButton(new ImageIcon(ruta));
    barra.add(boton);
    return boton;
}
```

Ahora el código que vamos a introducir que sustituirá el que está marcado como comentario será el siguiente:

```

barra=new JToolBar();
    configura_barra("src/graficos/negrita.png").addActionListener(new
StyledEditorKit.BoldAction());
    configura_barra("src/graficos/cursiva.png").addActionListener(new
StyledEditorKit.ItalicAction());
    configura_barra("src/graficos/subrayado.png").addActionListener(new
StyledEditorKit.UnderlineAction());
    barra.addSeparator(); ←
    configura_barra("src/graficos/bolaAzul.png").addActionListener(new
StyledEditorKit.ForegroundAction("Pone azul", Color.BLUE));
    configura_barra("src/graficos/bolaAmarilla.png").addActionListener(new
StyledEditorKit.ForegroundAction("Pone amarillo", Color.YELLOW));
    configura_barra("src/graficos/bolaRoja.png").addActionListener(new
StyledEditorKit.ForegroundAction("Pone rojo", Color.RED));
    barra.addSeparator(); ←
    configura_barra("src/graficos/izquierda.png").addActionListener(new
StyledEditorKit.AlignmentAction("izquierda", 0));
    configura_barra("src/graficos/centrado.png").addActionListener(new
StyledEditorKit.AlignmentAction("centrado", 1));
    configura_barra("src/graficos/derecha.png").addActionListener(new
StyledEditorKit.AlignmentAction("derecha", 2));
    configura_barra("src/graficos/justificado.png").addActionListener(new
StyledEditorKit.AlignmentAction("justificado", 3));

```

---

Como al incluir separadores necesitamos modificar el tamaño de la ventana:

```

18 class MenuProcesador_II extends JFrame{
19     public MenuProcesador_II() {
20         setBounds(500,300,550,420); ←
21         LaminaProcesador_II milamina=new LaminaProcesador_II();
22         add(milamina);
23         setVisible(true);
24     }
25 }

```

---

Eliminamos el texto que pasamos a comentarios, este es el texto final definitivo.

```

package graficos;
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Font;
import java.awt.event.*;
import java.awt.font.TextAttribute;
import javax.swing.*;
import javax.swing.text.StyledEditorKit;

public class Procesador_II {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        MenuProcesador_II mimarco=new MenuProcesador_II();
        mimarco.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}

class MenuProcesador_II extends JFrame{
    public MenuProcesador_II() {
        setBounds(500,300,550,420);

```

```

        LaminaProcesador_II milamina=new LaminaProcesador_II();
        add(milamina);
        setVisible(true);
    }
}
class LaminaProcesador_II extends JPanel{
    public LaminaProcesador_II() {
        setLayout(new BorderLayout());
        JPanel laminamenu=new JPanel();
        JMenuBar mibarra=new JMenuBar();

        fuente=new JMenu("Fuente");
        estilo=new JMenu("Estilo");
        tamagno=new JMenu("Tamaño");

        configura_menu("Arial","fuente","Arial", 9,10,"");
        configura_menu("Courier","fuente","Courier", 9,10,"");
        configura_menu("Verdana","fuente","Verdana", 9,10,"");

        configura_menu("Negrita","estilo","",Font.BOLD,1,"src/graficos/negrita
.png");
        configura_menu("Cursiva","estilo","",Font.ITALIC,1,"src/graficos/cursi
va.png");

        ButtonGroup tamagno_letra=new ButtonGroup();
        JRadioButtonMenuItem doce=new JRadioButtonMenuItem("12");
        JRadioButtonMenuItem dieciseis=new JRadioButtonMenuItem("16");
        JRadioButtonMenuItem veinte=new JRadioButtonMenuItem("20");
        JRadioButtonMenuItem veinticuatro=new JRadioButtonMenuItem("24");

        veinticuatro.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_D,Input
Event.CTRL_DOWN_MASK));

        tamagno_letra.add(doce);
        tamagno_letra.add(dieciseis);
        tamagno_letra.add(veinte);
        tamagno_letra.add(veinticuatro);
        doce.addActionListener(new
StyledEditorKit.FontSizeAction("cambia_tamaño", 12));
        dieciseis.addActionListener(new
StyledEditorKit.FontSizeAction("cambia_tamaño", 16));
        veinte.addActionListener(new
StyledEditorKit.FontSizeAction("cambia_tamaño", 20));
        veinticuatro.addActionListener(new
StyledEditorKit.FontSizeAction("cambia_tamaño", 24));
        tamagno.add(doce);
        tamagno.add(dieciseis);
        tamagno.add(veinte);
        tamagno.add(veinticuatro);

        mibarra.add(fuente);
        mibarra.add(estilo);
        mibarra.add(tamagno);

        laminamenu.add(mibarra);
        add(laminamenu, BorderLayout.NORTH);

        miarea=new JTextPane();
        add(miarea, BorderLayout.CENTER);

```

```

JPopupMenu emergente=new JPopupMenu();
JMenuItem negritaE=new JMenuItem("Negrita");
JMenuItem cursivaE=new JMenuItem("Cursiva");

emergente.add(negritaE);
emergente.add(cursivaE);

miarea.setComponentPopupMenu(emergente);

barra=new JToolBar();
configura_barra("src/graficos/negrita.png").addActionListener(new
StyledEditorKit.BoldAction());
configura_barra("src/graficos/cursiva.png").addActionListener(new
StyledEditorKit.ItalicAction());
configura_barra("src/graficos/subrayado.png").addActionListener(new
StyledEditorKit.UnderlineAction());
barra.addSeparator();
configura_barra("src/graficos/bolaAzul.png").addActionListener(new
StyledEditorKit.ForegroundAction("Pone azul", Color.BLUE));
configura_barra("src/graficos/bolaAmarilla.png").addActionListener(new
StyledEditorKit.ForegroundAction("Pone amarillo", Color.YELLOW));
configura_barra("src/graficos/bolaRoja.png").addActionListener(new
StyledEditorKit.ForegroundAction("Pone rojo", Color.RED));
barra.addSeparator();
configura_barra("src/graficos/izquierda.png").addActionListener(new
StyledEditorKit.AlignmentAction("izquierda", 0));
configura_barra("src/graficos/centrado.png").addActionListener(new
StyledEditorKit.AlignmentAction("centrado", 1));
configura_barra("src/graficos/derecha.png").addActionListener(new
StyledEditorKit.AlignmentAction("derecha", 2));
configura_barra("src/graficos/justificado.png").addActionListener(new
StyledEditorKit.AlignmentAction("justificado", 3));

barra.setOrientation(1);
add(barra, BorderLayout.WEST);
//-----
}

public JButton configura_barra(String ruta) {
    JButton boton=new JButton(new ImageIcon(ruta));
    barra.add(boton);
    return boton;
}

public void configura_menu(String rotulo, String menu, String
tipo_letra, int estilos, int tam, String ruta_icono) {
    JMenuItem elem_menu=new JMenuItem(rotulo, new
ImageIcon(ruta_icono));
    if(menu=="fuente") {
        fuente.add(elem_menu);
        if(tipo_letra=="Arial") {
            elem_menu.addActionListener(new
StyledEditorKit.FontFamilyAction("cambia_letra", "Arial"));
        }
        else if(tipo_letra=="Courier") {

```

```

        elem_menu.addActionListener(new
StyledEditorKit.FontFamilyAction("cambia_letra", "Courier"));
    }
    else if(tipo_letra=="Verdana") {
        elem_menu.addActionListener(new
StyledEditorKit.FontFamilyAction("cambia_letra", "Verdana"));
    }
}else if(menu=="estilo"){
    estilo.add(elem_menu);
    if(estilos==Font.BOLD) {

        elem_menu.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_N,
InputEvent.CTRL_DOWN_MASK));

        elem_menu.addActionListener(new
StyledEditorKit.BoldAction());

        }else if(estilos==Font.ITALIC) {

        elem_menu.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_K,
InputEvent.CTRL_DOWN_MASK));

        elem_menu.addActionListener(new
StyledEditorKit.ItalicAction());

        }
    }
    else if(menu=="tamaño") {
        tamagno.add(elem_menu);

        elem_menu.addActionListener(new
StyledEditorKit.FontSizeAction("cambia_tamaño", tam));
    }
}
JTextPane miarea;
JMenu fuente, estilo, tamagno;
Font letras;
JButton negritaBarra, cursivaBarra;
JButton subraBarra, azulBarra, rojoBarra, amarilloBarra;
JButton a_izquierda, a_centrado, a_derecha, a_justificado;
JToolBar barra;
}

```

---





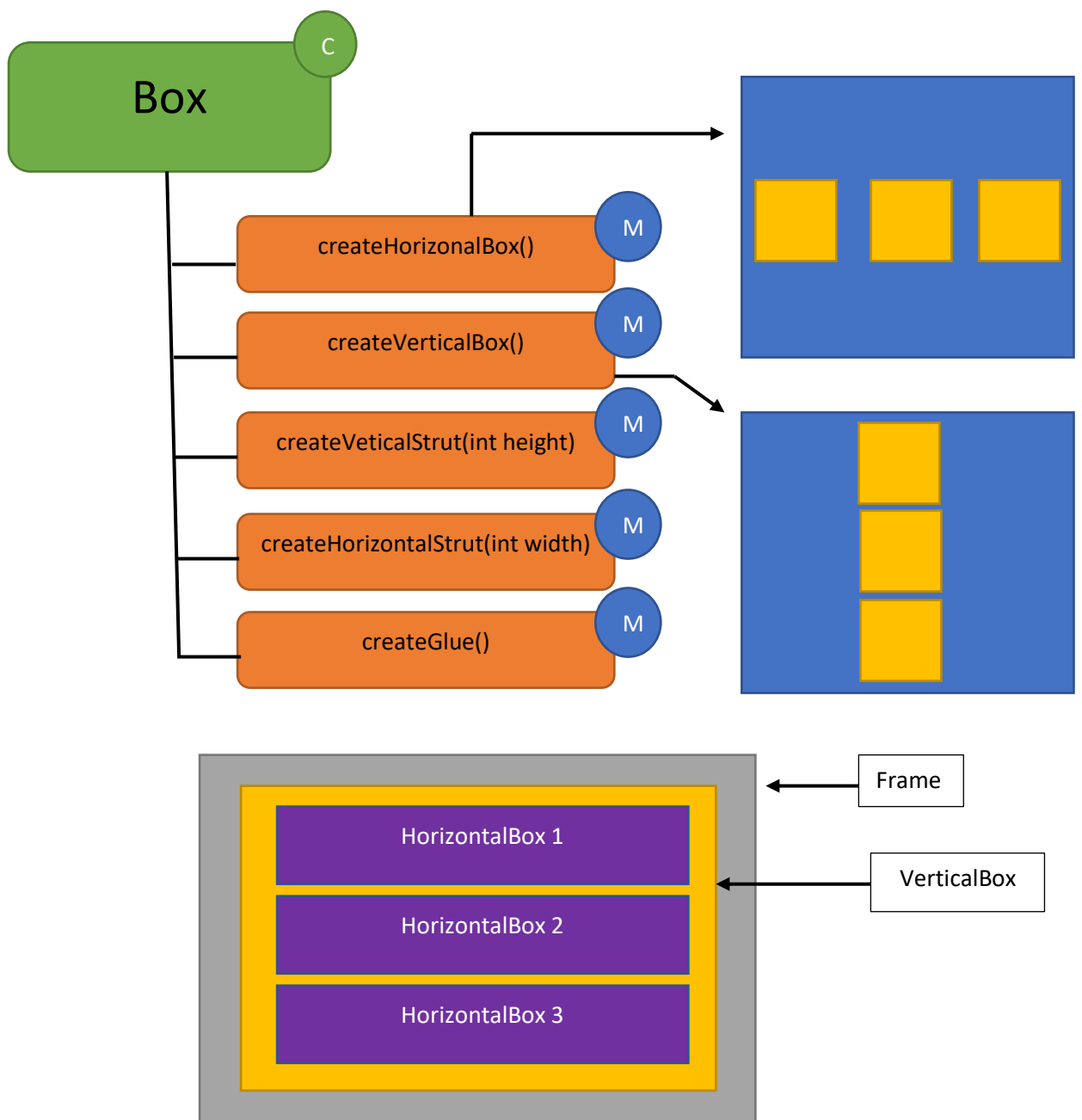


# Aplicaciones gráficas. Disposiciones avanzadas I. Box. (Vídeo 115)

Disposiciones en caja.



Disposición en caja.



Vamos a crear un código que cree esta estructura.

```
package graficos;
import java.awt.*;
import javax.swing.*;

public class PruebaDisposiciones {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        MarcoCaja mimarco=new MarcoCaja();
        mimarco.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        mimarco.setVisible(true);
    }
}

class MarcoCaja extends JFrame{
    public MarcoCaja() {
        setTitle("Marco en la caja");
        setBounds(600,350,200,200);
        JLabel rotulo1=new JLabel("Nombre");
        JTextField texto1=new JTextField(10);
        texto1.setMaximumSize(texto1.getPreferredSize());
        Box cajaH1=Box.createHorizontalBox();
        cajaH1.add(rotulo1);
        cajaH1.add(Box.createHorizontalStrut(10));
        cajaH1.add(texto1);
        JLabel rotulo2=new JLabel("Contraseña");
        JTextField texto2=new JTextField(10);
        texto2.setMaximumSize(texto2.getPreferredSize());
        Box cajaH2=Box.createHorizontalBox();
        cajaH2.add(rotulo2);
        cajaH2.add(Box.createHorizontalStrut(10));
        cajaH2.add(texto2);
        JButton boton1 =new JButton("OK");
        JButton boton2 =new JButton("Cancelar");
        Box cajaH3=Box.createHorizontalBox();
        cajaH3.add(boton1);
        cajaH3.add(Box.createGLue());
        cajaH3.add(boton2);
        Box cajaV1=Box.createVerticalBox();
        cajaV1.add(cajaH1);
        cajaV1.add(cajaH2);
        cajaV1.add(cajaH3);
        add(cajaV1, BorderLayout.CENTER);
    }
}
```

HorizontalBox1

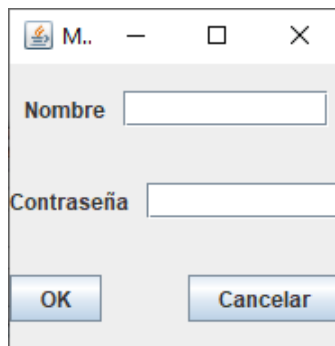
HorizontalBox2

HorizontalBox3

VerticalBox

Frame

Este será el resultado:

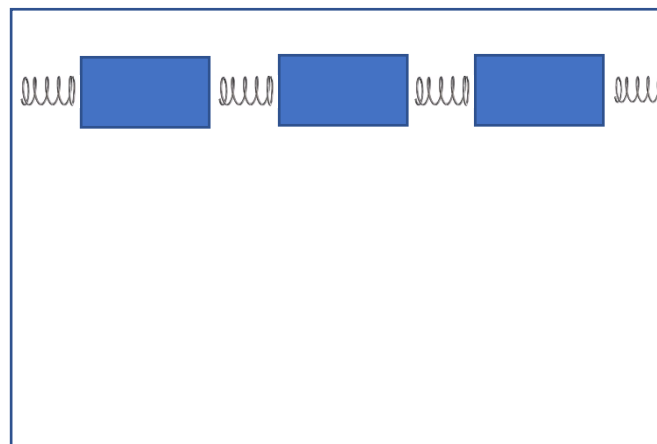
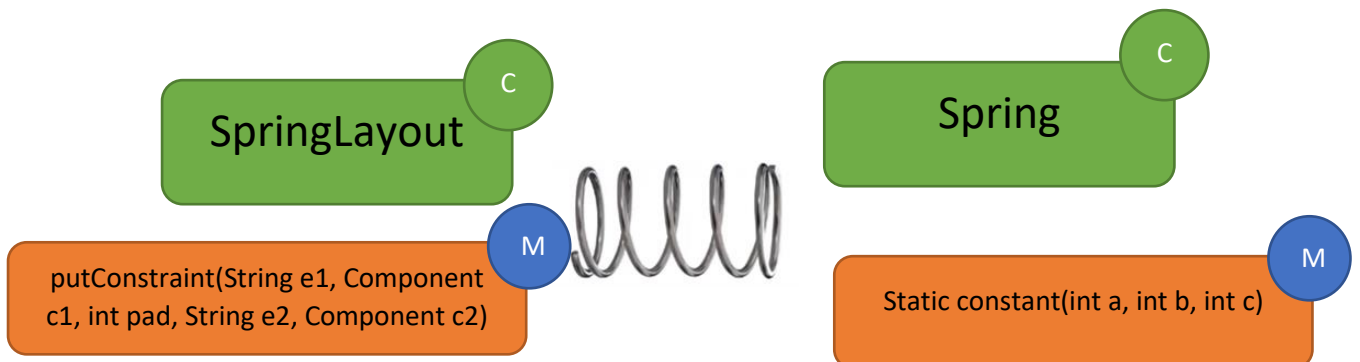




## Aplicaciones gráficas. Disposiciones avanzadas II. Spring.

(Vídeo 116)

Disposición en muelle.



```
package graficos;
import javax.swing.*;
public class DisposicionMuelle {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        MarcoMuelle mimarco=new MarcoMuelle();
        mimarco.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}
class MarcoMuelle extends JFrame{
    public MarcoMuelle() {
        setBounds(300,400,1000,350);
        LaminaMuelle milamina=new LaminaMuelle();
        add(milamina);
        setVisible(true);
    }
}
class LaminaMuelle extends JPanel{
    public LaminaMuelle() {
        JButton boton1=new JButton("boton 1");
        JButton boton2=new JButton("boton 2");
        JButton boton3=new JButton("boton 3");
```

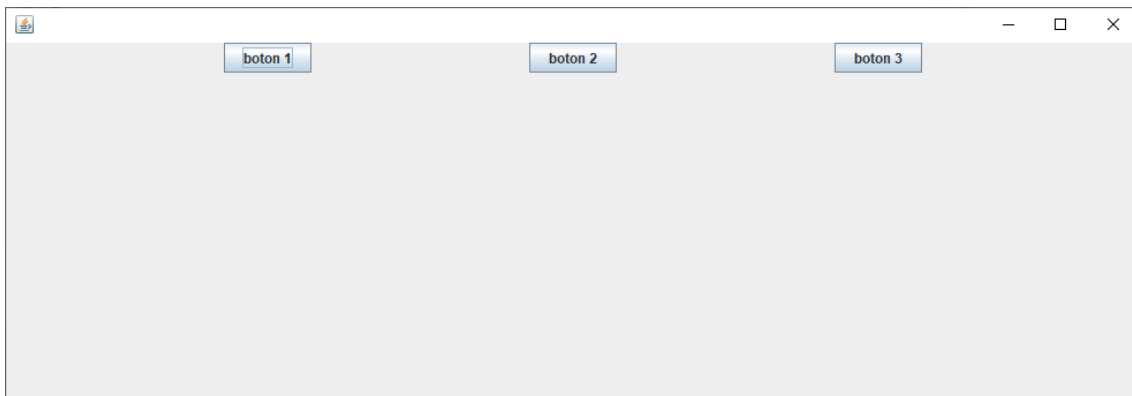
```

        SpringLayout milayout = new SpringLayout();
        setLayout(milayout);
        add(boton1);
        add(boton2);
        add(boton3);
        Spring mimuelle=Spring.constant(0,10,100);
        milayout.putConstraint(SpringLayout.WEST,boton1, mimuelle,
SpringLayout.WEST, this);
        milayout.putConstraint(SpringLayout.WEST,boton2, mimuelle,
SpringLayout.EAST, boton1);
        milayout.putConstraint(SpringLayout.WEST,boton3, mimuelle,
SpringLayout.EAST, boton2);
        milayout.putConstraint(SpringLayout.EAST,this, mimuelle,
SpringLayout.EAST, boton3);
    }
}

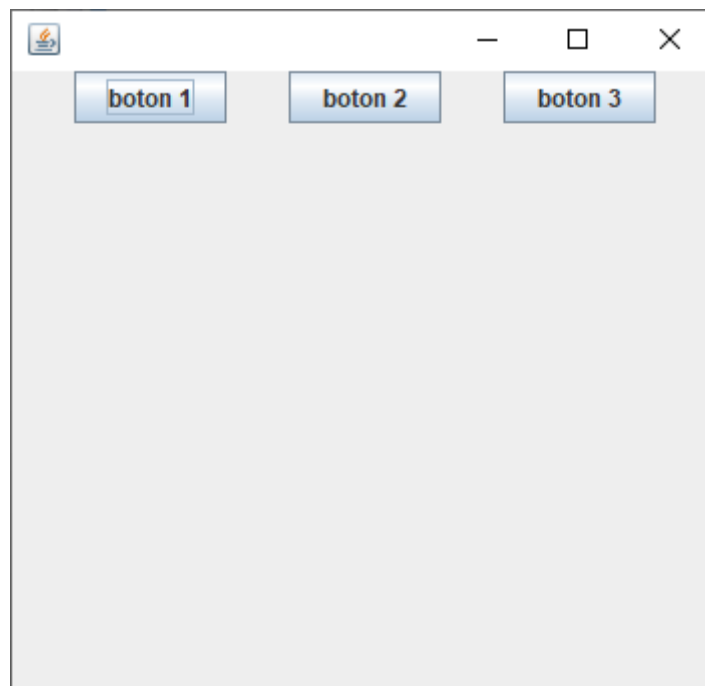
```

---

Cuando ejecutemos veremos la siguiente ventana:



Vamos a contraer la ventana horizontalmente.



Vamos a expandirla horizontalmente.

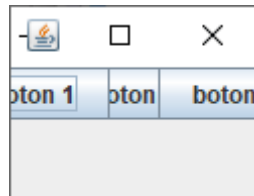


Los botones tienen un efecto muelle.



## Aplicaciones gráficas. Disposiciones avanzadas III. Spring II (Video 117).

Con respecto al capítulo anterior si contraemos la ventana al máximo, veremos que los botones se solapan.



```
19 class LaminaMuelle extends JPanel{
20     public LaminaMuelle() {
21         JButton boton1=new JButton("boton 1");
22         JButton boton2=new JButton("boton 2");
23         JButton boton3=new JButton("boton 3");
24         SpringLayout milayout = new SpringLayout();
25         setLayout(milayout);
26         add(boton1);
27         add(boton2);
28         add(boton3);
29         Spring mimuelle=Spring.constant(0,0,0);
30         milayout.putConstraint(SpringLayout.WEST,boton1, mimuelle, SpringLayout.WEST, this);
31         milayout.putConstraint(SpringLayout.WEST,boton2, mimuelle, SpringLayout.EAST, boton1);
32         milayout.putConstraint(SpringLayout.WEST,boton3, mimuelle, SpringLayout.EAST, boton2);
33         milayout.putConstraint(SpringLayout.EAST,this, mimuelle, SpringLayout.EAST, boton3);
34     }
35 }
```

Si pasamos los valores a 0,0,0 el resultado será el siguiente:



Ya no hay efecto muelle.

```
19 class LaminaMuelle extends JPanel{
20     public LaminaMuelle() {
21         JButton boton1=new JButton("boton 1");
22         JButton boton2=new JButton("boton 2");
23         JButton boton3=new JButton("boton 3");
24         SpringLayout milayout = new SpringLayout();
25         setLayout(milayout);
26         add(boton1);
27         add(boton2);
28         add(boton3);
29         Spring mimuelle=Spring.constant(0,0,10);
30         milayout.putConstraint(SpringLayout.WEST,boton1, mimuelle, SpringLayout.WEST, this);
31         milayout.putConstraint(SpringLayout.WEST,boton2, mimuelle, SpringLayout.EAST, boton1);
32         milayout.putConstraint(SpringLayout.WEST,boton3, mimuelle, SpringLayout.EAST, boton2);
33         milayout.putConstraint(SpringLayout.EAST,this, mimuelle, SpringLayout.EAST, boton3);
34     }
35 }
```

Si al tercer valor le ponemos un 10 verás que ya vuelve a tener el comportamiento de muelle, pero los botones no se solapan.

Volvemos a dejar los valores en 1,10,100.



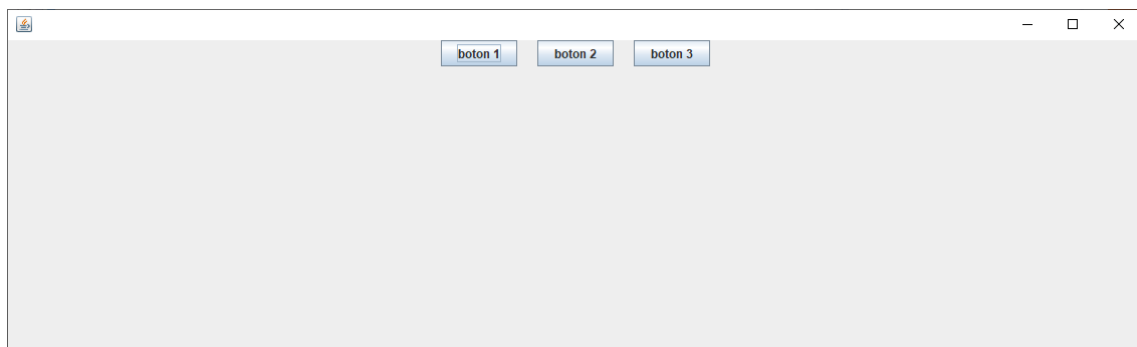
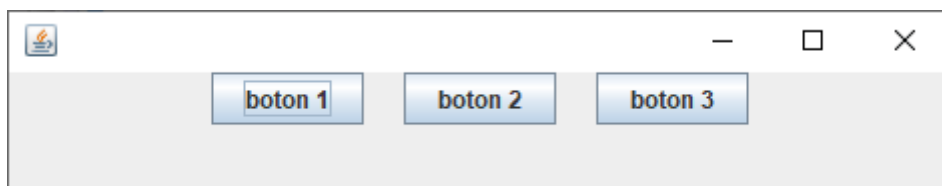
Ahora queremos que los muelles que separan los botones sean rígidos, para ello tendremos que construir otro muelle.

```
1 package graficos;
2 import javax.swing.*;
3 public class DisposicionMuelle {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         MarcoMuelle mimarco=new MarcoMuelle();
8         mimarco.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
9     }
10 }
11 class MarcoMuelle extends JFrame{
12     public MarcoMuelle() {
13         setBounds(300,400,1000,350);
14         LaminaMuelle milamina=new LaminaMuelle();
15         add(milamina);
16         setVisible(true);
17     }
18 }
19 class LaminaMuelle extends JPanel{
20     public LaminaMuelle() {
21         JButton boton1=new JButton("boton 1");
22         JButton boton2=new JButton("boton 2");
23         JButton boton3=new JButton("boton 3");
24         SpringLayout milayout = new SpringLayout();
25         setLayout(milayout);
26         add(boton1);
27         add(boton2);
28         add(boton3);
29         Spring mimuelle=Spring.constant(0,10,100);
30         Spring mimuelle_rigido=Spring.constant(20);
31         milayout.putConstraint(SpringLayout.WEST,boton1, mimuelle, SpringLayout.WEST, this);
32         milayout.putConstraint(SpringLayout.WEST,boton2, mimuelle_rigido, SpringLayout.EAST, boton1);
33         milayout.putConstraint(SpringLayout.WEST,boton3, mimuelle_rigido, SpringLayout.EAST, boton2);
34         milayout.putConstraint(SpringLayout.EAST,this, mimuelle, SpringLayout.EAST, boton3);
35     }
36 }
```

En la línea 30 creamos un botón rígido, ya que este constructor admite solo un parámetro.

En las líneas 32 y 33 cambiamos a muelle\_rigido que hace que los botones entre sí no sean elásticos.

Cuando ejecutes verás que estos se centrarán a lo ancho del marco pero entre ellos mantendrán la misma distancia.



Ahora puedes jugar con los parámetros de los muelles fijos y los muelles elásticos.



## Aplicaciones gráficas. Disposiciones avanzadas IV. Disposiciones libres I. (Vídeo 118)

Disposición libre.

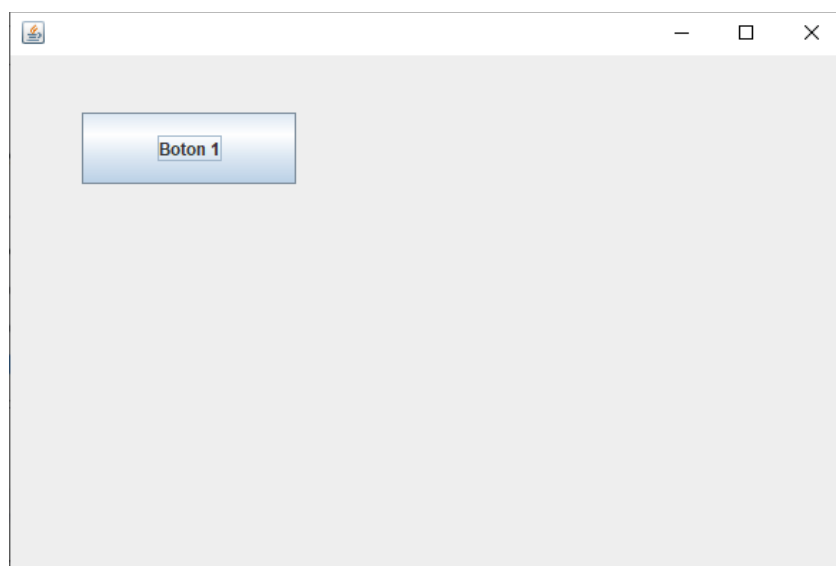
```
1 package graficos;
2 import javax.swing.*;
3
4 public class LibreDisposicion {
5
6     public static void main(String[] args) {
7         // TODO Auto-generated method stub
8         MarcoLibre mimarco=new MarcoLibre();
9         mimarco.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
10    }
11 }
12
13 class MarcoLibre extends JFrame{
14     public MarcoLibre() {
15         setBounds(450,350,600,400);
16         LaminaLibre milamina=new LaminaLibre();
17         add(milamina);
18         setVisible(true);
19     }
20 }
21
22 class LaminaLibre extends JPanel{
23     public LaminaLibre() {
24         setLayout(null);
25         JButton boton1=new JButton("Boton 1");
26         boton1.setBounds(50, 40, 150, 50);
27         add(boton1);
28     }
29 }
```

Vamos a trabajar en disposición libre.

Definimos un botón con el nombre boton1.

Le damos las coordenadas Horizontal, Vertical, ancho y alto.

Agregamos el botón a la lámina.



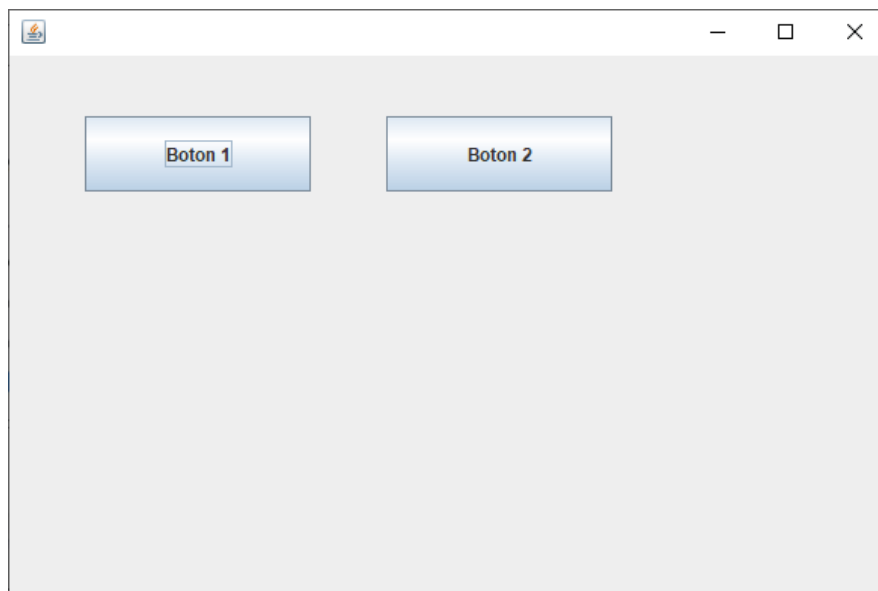
Vamos a colocar un segundo botón.

```

1 package graficos;
2 import javax.swing.*;
3
4 public class LibreDisposicion {
5
6     public static void main(String[] args) {
7         // TODO Auto-generated method stub
8         MarcoLibre mimarco=new MarcoLibre();
9         mimarco.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
10    }
11 }
12
13 class MarcoLibre extends JFrame{
14     public MarcoLibre() {
15         setBounds(450,350,600,400);
16         LaminaLibre milamina=new LaminaLibre();
17         add(milamina);
18         setVisible(true);
19     }
20 }
21
22 class LaminaLibre extends JPanel{
23     public LaminaLibre() {
24         setLayout(null);
25         JButton boton1=new JButton("Boton 1");
26         boton1.setBounds(50, 40, 150, 50);
27         JButton boton2=new JButton("Boton 2");
28         boton2.setBounds(250, 40, 150, 50);
29         add(boton1);
30         add(boton2);
31     }
32 }

```

Este será el resultado:



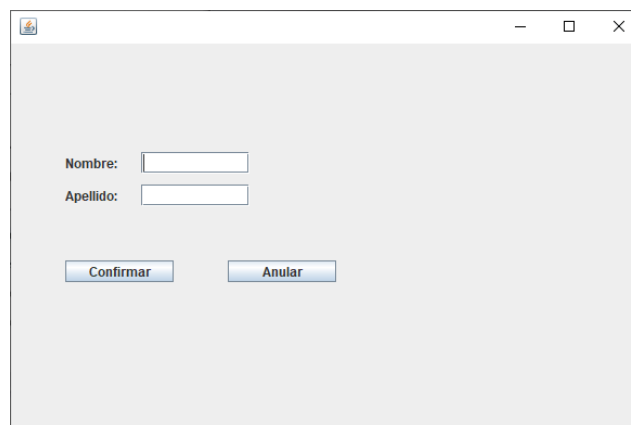
Vamos a realizar un mini formulario para practicar lo aprendido.

```

1 package graficos;
2 import javax.swing.*;
3
4 public class LibreDisposicion {
5
6     public static void main(String[] args) {
7         // TODO Auto-generated method stub
8         MarcoLibre mimarco=new MarcoLibre();
9         mimarco.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
10    }
11 }
12
13 class MarcoLibre extends JFrame{
14     public MarcoLibre() {
15         setBounds(450,350,600,400);
16         LaminaLibre milamina=new LaminaLibre();
17         add(milamina);
18         setVisible(true);
19     }
20 }
21 class LaminaLibre extends JPanel{
22     public LaminaLibre() {
23         setLayout(null);
24         JLabel nombre=new JLabel("Nombre: ");
25         JLabel apellido=new JLabel("Apellido: ");
26         JTextField c_nombre=new JTextField();
27         JTextField c_apellido=new JTextField();
28         JButton boton1=new JButton("Confirmar");
29         JButton boton2=new JButton("Anular");
30
31         nombre.setBounds(50, 100, 100, 20);
32         c_nombre.setBounds(120, 100, 100,20);
33         apellido.setBounds(50,130,100,20);
34         c_apellido.setBounds(120,130,100, 20);
35         boton1.setBounds(50, 200, 100, 20);
36         boton2.setBounds(200, 200, 100, 20);
37
38         add(nombre);
39         add(c_nombre);
40         add(apellido);
41         add(c_apellido);
42         add(boton1);
43         add(boton2);
44     }
45 }

```

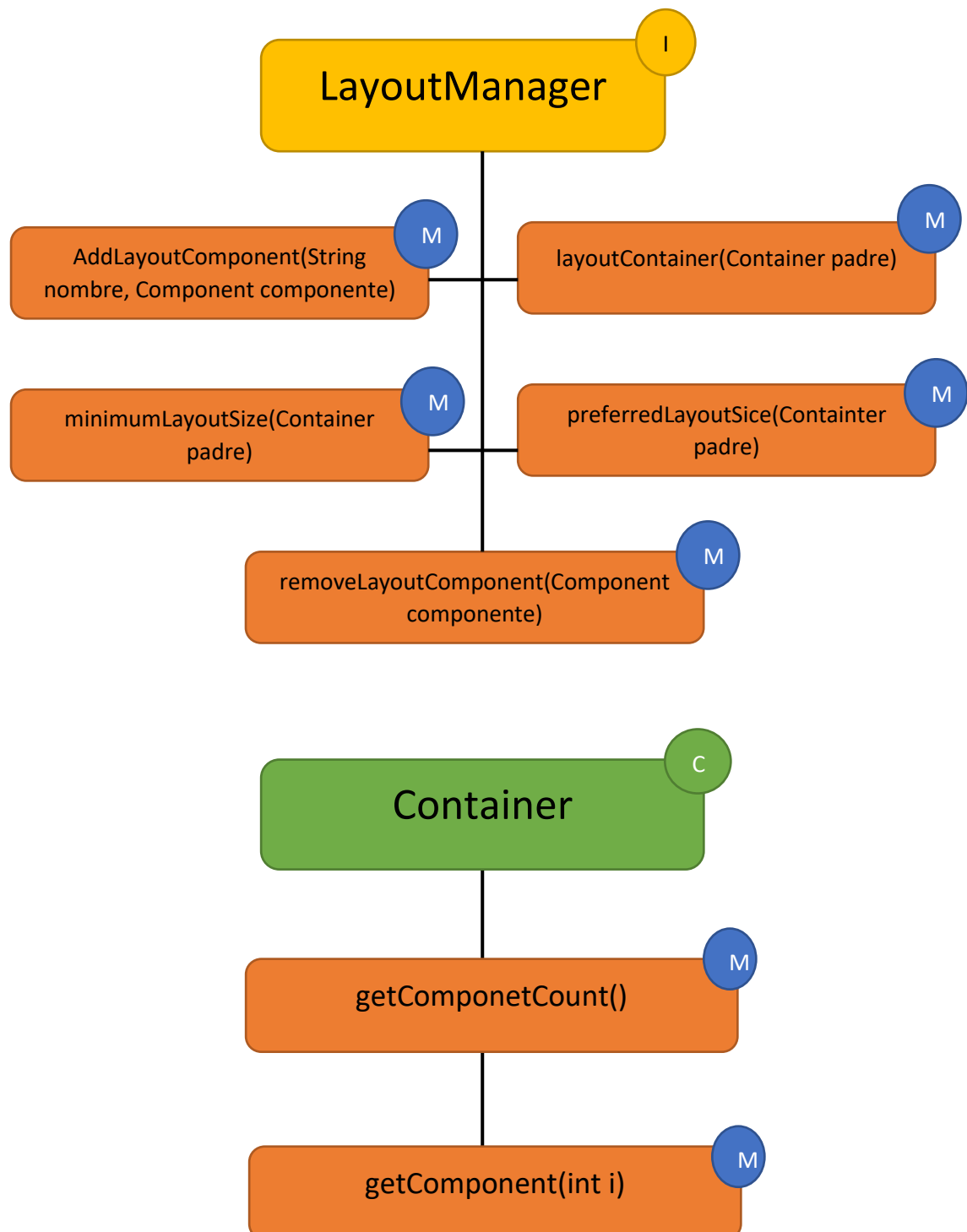
Este será el resultado:





## Aplicaciones gráficas. Disposiciones avanzadas V. Disposiciones libres II. (Vídeo 119)

Crear tus propias disposiciones.



Vamos a escribir el siguiente código:

```
package graficos;
import java.awt.Component;
import java.awt.Container;
import java.awt.Dimension;
import java.awt.LayoutManager;

import javax.swing.*;

public class LibreDisposicion {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        MarcoLibre mimarco=new MarcoLibre();
        mimarco.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}

class MarcoLibre extends JFrame{
    public MarcoLibre() {
        setBounds(450,350,600,400);
        LaminaLibre milamina=new LaminaLibre();
        add(milamina);
        setVisible(true);
    }
}

class LaminaLibre extends JPanel{
    public LaminaLibre() {
        setLayout(new EnColumnas());
        JLabel nombre=new JLabel("Nombre: ");
        JLabel apellido=new JLabel("Apellido: ");
        JTextField c_nombre=new JTextField();
        JTextField c_apellido=new JTextField();
        JButton boton1=new JButton("Confirmar");
        JButton boton2=new JButton("Anular");

        add(nombre);
        add(c_nombre);
        add(apellido);
        add(c_apellido);
        add(boton1);
        add(boton2);
    }
}

class EnColumnas implements LayoutManager{

    @Override
    public void addLayoutComponent(String name, Component comp) {
        // TODO Auto-generated method stub
    }

    @Override
    public void removeLayoutComponent(Component comp) {
        // TODO Auto-generated method stub
    }
}
```

Llamamos a la clase que hemos creado.

Creamos una nueva clase llamada EnColumna que implemente LayoutManager.



```

@Override
public Dimension preferredLayoutSize(Container micontenedor) {
    // TODO Auto-generated method stub
    return null;
}

@Override
public Dimension minimumLayoutSize(Container micontenedor) {
    // TODO Auto-generated method stub
    return null;
}

@Override
public void layoutContainer(Container micontenedor) {
    // TODO Auto-generated method stub
    int contador=0;
    int n=micontenedor.getComponentCount();
    for(int i=0; i<n; i++) {
        contador++;
        Component c=micontenedor.getComponent(i);
        c.setBounds(x,y, 100,20);
        x+=100;
        if(contador%2==0) {
            y+=40;
            x=20;
        }
    }
}

private int x=20;
private int y=20;
}

```

Definimos contador a 0.  
 Definimos n y se asignamos el número de componentes que tiene.  
 Con un bucle for hacemos un recorrido por todos los componentes.  
 Variable contador lo cincrementamos uno.

Componet c= le asignamos el primer componente que está remarcado en rojo, con las cooredenadas c.setBounds(x,y, 100,20).

Incrementamos la x en 100.

Si contador en par que lo controlamos con la condición contador%2==0, incrementamos y en 40 y x vuelve a valer 20.

Cuando ejecutemos este será el resultado:

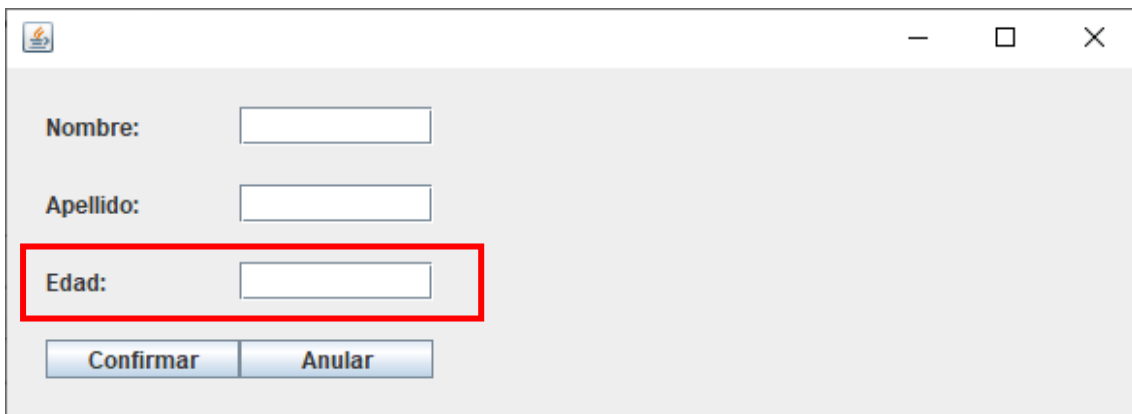
The screenshot shows a standard Java Swing window with a title bar containing a small icon, a minus sign, a maximize button, and a close button. The window's content area has a light gray background. It contains two labels, 'Nombre:' and 'Apellido:', each followed by a white text input field with a thin gray border. Below these fields are two buttons: 'Confirmar' and 'Anular', both with a blue gradient and white text.

Esto nos permite ir agregando componentes sin preocuparnos de las posiciones ya que el trabajo sucio lo realizará la clase EnColumnas.

Vamos a crear una etiqueta que diga Edad: y su correspondiente cuadro de texto.

```
26 class LaminaLibre extends JPanel{
27     public LaminaLibre() {
28         setLayout(new EnColumnas());
29         JLabel nombre=new JLabel("Nombre: ");
30         JLabel apellido=new JLabel("Apellido: ");
31         JLabel edad=new JLabel("Edad: ");
32         JTextField c_nombre=new JTextField();
33         JTextField c_apellido=new JTextField();
34         JTextField c_edad =new JTextField();
35         JButton boton1=new JButton("Confirmar");
36         JButton boton2=new JButton("Anular");
37
38
39
40         add(nombre);
41         add(c_nombre);
42         add(apellido);
43         add(c_apellido);
44         add(edad);
45         add(c_edad);
46         add(boton1);
47         add(boton2);
48     }
49 }
```

Solo hemos de introducir este código, este será el resultado:



The screenshot shows a Java Swing window with a light gray background. It contains three text input fields arranged vertically, each preceded by a label: 'Nombre:', 'Apellido:', and 'Edad:'. The 'Edad:' label and its corresponding text field are highlighted with a red rectangular border. Below the text fields, there are two buttons: 'Confirmar' and 'Anular', positioned side-by-side. The window has standard Windows-style window controls (minimize, maximize, close) in the top right corner.





## Aplicaciones gráficas. Disposiciones avanzadas VI. Disposiciones libres III. (Vídeo 120)

En este capítulo vamos a agregar elementos por parejas y centrados:

---

```
package graficos;
import java.awt.Component;
import java.awt.Container;
import java.awt.Dimension;
import java.awt.LayoutManager;

import javax.swing.*;

public class LibreDisposicion {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        MarcoLibre mimarco=new MarcoLibre();
        mimarco.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}

class MarcoLibre extends JFrame{
    public MarcoLibre() {
        setBounds(450,350,600,400);
        LaminaLibre milamina=new LaminaLibre();
        add(milamina);
        setVisible(true);
    }
}

class LaminaLibre extends JPanel{
    public LaminaLibre() {
        setLayout(new EnColumnas());
        JLabel nombre=new JLabel("Nombre: ");
        JLabel apellido=new JLabel("Apellido: ");
        JLabel edad=new JLabel("Edad: ");
        JLabel telf=new JLabel("Teléfono: ");
        JTextField c_nombre=new JTextField();
        JTextField c_apellido=new JTextField();
        JTextField c_edad =new JTextField();
        JTextField c_telf = new JTextField();
        JButton boton1=new JButton("Confirmar");
        JButton boton2=new JButton("Anular");

        add(nombre);
        add(c_nombre);
        add(apellido);
        add(c_apellido);
        add(edad);
        add(c_edad);
        add(telf);
        add(c_telf);
        add(boton1);
        add(boton2);
    }
}

class EnColumnas implements LayoutManager{

    @Override
```

```

public void addLayoutComponent(String name, Component comp) {
    // TODO Auto-generated method stub

}

@Override
public void removeLayoutComponent(Component comp) {
    // TODO Auto-generated method stub

}

@Override
public Dimension preferredLayoutSize(Container micontenedor) {
    // TODO Auto-generated method stub
    return null;
}

@Override
public Dimension minimumLayoutSize(Container micontenedor) {
    // TODO Auto-generated method stub
    return null;
}

@Override
public void layoutContainer(Container micontenedor) {
    // TODO Auto-generated method stub
    int d=micontenedor.getWidth();
    x=d/2;
    int contador=0;
    int n=micontenedor.getComponentCount();
    for(int i=0; i<n; i++) {
        contador++;
        Component c=micontenedor.getComponent(i);
        c.setBounds(x-100,y, 100,20);
        x+=100;
        if(contador%2==0) {
            y+=40;
            x=d/2;
        }
    }
}

private int x;
private int y=20;
}

```

---

La variable d asume el ancho de micontenedor, etiqueta, caja de texto o botón.

x asume la mitad del valor de la variable d.

En `c.setBounds(x-100,y,100,20)`, a la coordenada x le restamos el ancho del objeto.

En `x0d/2`; nos colocamos de nuevo en el centro de micontenedor para colocar a la siguiente pareja de elementos.

Private int x; la declaramos pero no la inicializamos, la inicializamos cuando decimos `x=d/2`.

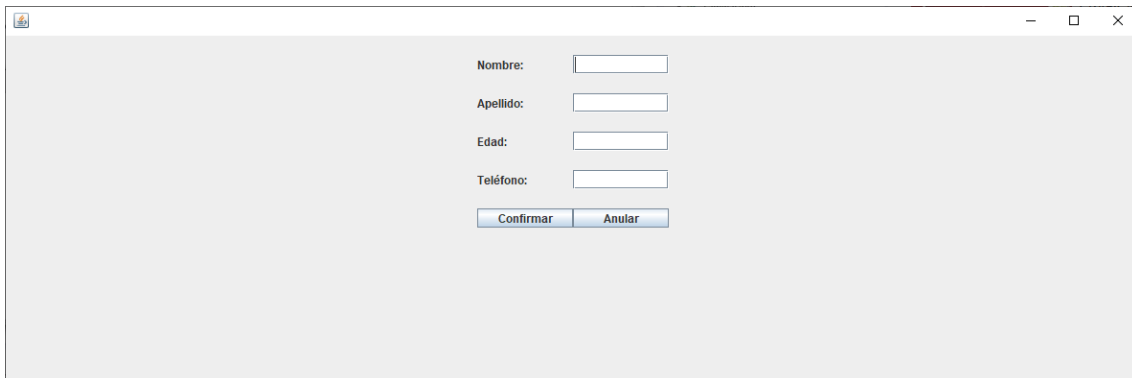
Este será el resultado:



Ahora vamos a modificar el acho de nuestro marco.

```
18 class MarcoLibre extends JFrame{
19     public MarcoLibre() {
20         setBounds(450,350,1200,400);
21         LaminaLibre milamina=new LaminaLibre();
22         add(milamina);
23         setVisible(true);
24     }
25 }
```

Si ejecutamos.

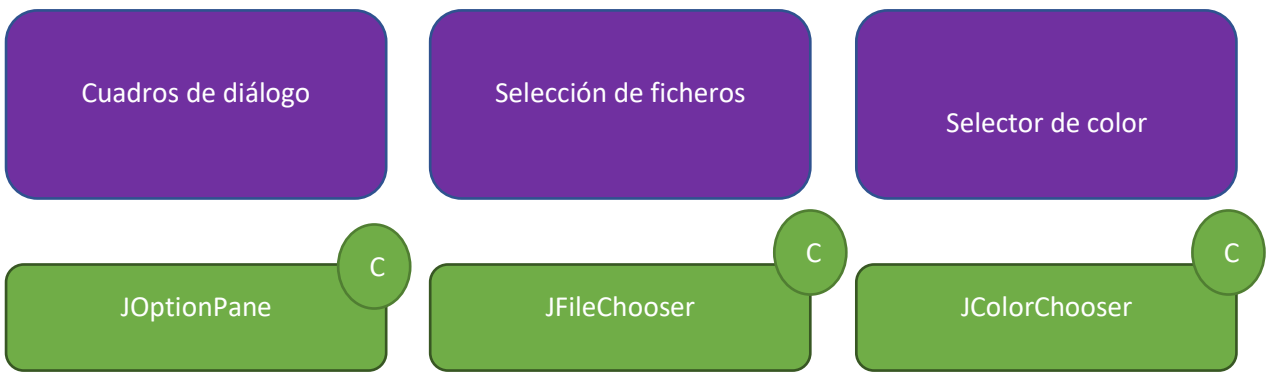
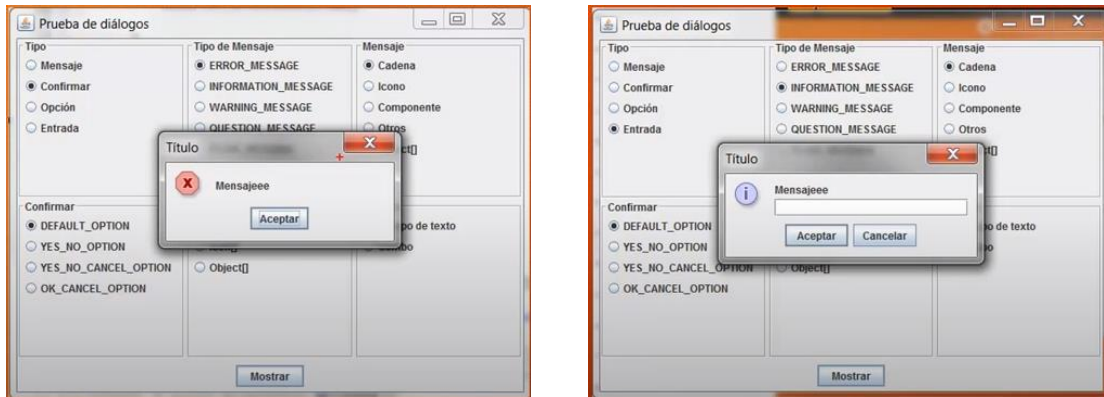


El marco es más ancho pero los elementos siguen centrados, lo dejamos de nuevo a 600.

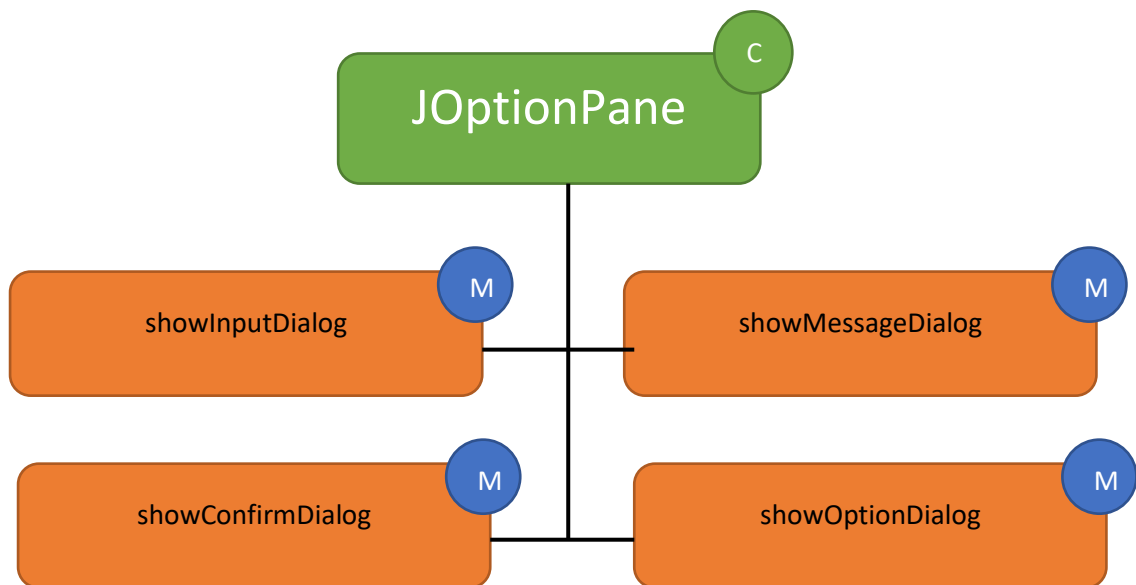




# Aplicaciones gráficas. Ventanas emergentes. Cuadros de diálogo I (Vídeo 121)



JOptionPane





```

package graficos;
import javax.swing.*;
import java.awt.event.*;
public class Tipos_Cuadros_Dialogos {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Cuadros_Dialogos mimarco=new Cuadros_Dialogos();
        mimarco.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        mimarco.setVisible(true);
    }
}

class Cuadros_Dialogos extends JFrame{
    public Cuadros_Dialogos() {
        setBounds(500,300,400,250);
        add(new Lamina_Cuadros_Dialogos());
    }
}

class Lamina_Cuadros_Dialogos extends JPanel{
    public Lamina_Cuadros_Dialogos() {
        boton1=new JButton("boton 1");
        boton2=new JButton("boton 2");
        boton3=new JButton("boton 3");
        boton4=new JButton("boton 4");

        boton1.addActionListener(new Accion_botones());
        boton2.addActionListener(new Accion_botones());
        boton3.addActionListener(new Accion_botones());
        boton4.addActionListener(new Accion_botones());

        add(boton1);
        add(boton2);
        add(boton3);
        add(boton4);
    }

    private class Accion_botones implements ActionListener{

        @Override
        public void actionPerformed(ActionEvent e) {
            // TODO Auto-generated method stub
            if(e.getSource()==boton1) {

                //JOptionPane.showMessageDialog(Lamina_Cuadros_Dialogos.this, "Mensaje
del botón 1");

                JOptionPane.showMessageDialog(Lamina_Cuadros_Dialogos.this, "Mensaje
del botón 1", "Titulo", 0); // 0, 1, 2 y 3
                System.out.println("Has pulsado el botón 1");
            } else if(e.getSource()==boton2) {
                //JOptionPane.showInputDialog("Introduce nombre:
");
                JOptionPane.showInputDialog(Lamina_Cuadros_Dialogos.this, "Introduce
nombre: ", "Titulo", 0); //0,1,2 y 3
            }
        }
    }
}

```

```

        System.out.println("Has pulsado el botón 2");
    }else if(e.getSource()==boton3) {
        System.out.println("Has pulsado el botón 3");
    }else {
        System.out.println("Has pulsado el botón 4");
    }
}
}
JButton boton1, boton2, boton3, boton4;
}

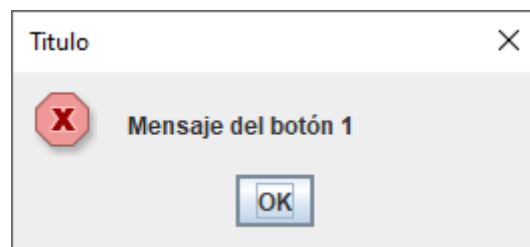
```

---

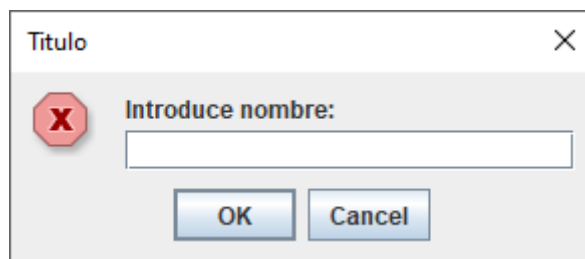
Este será el resultado:



Si seleccionamos el botón 1 este será el resultado:



Si seleccionamos el botón 2 este será el resultado:



En el siguiente capítulo terminaremos los botones 3 y 4.



## Aplicaciones gráficas. Ventanas emergentes. Cuadros diálogo II. (Vídeo 122)

Vamos a terminar el código:

```
private class Accion_botones implements ActionListener{

    @Override
    public void actionPerformed(ActionEvent e) {
        // TODO Auto-generated method stub
        if(e.getSource()==boton1) {

            //JOptionPane.showMessageDialog(Lamina_Cuadros_Dialogos.this, "Mensaje
del botón 1");

            JOptionPane.showMessageDialog(Lamina_Cuadros_Dialogos.this, "Mensaje
del botón 1","Titulo",0);// 0, 1, 2 y 3.
            System.out.println("Has pulsado el botón 1");
        }else if(e.getSource()==boton2) {
            //JOptionPane.showInputDialog("Introduce nombre:
");

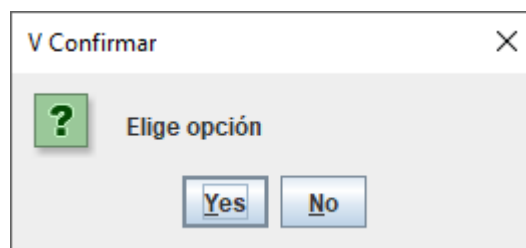
            JOptionPane.showInputDialog(Lamina_Cuadros_Dialogos.this,"Introduce
nombre: ", "Titulo",0);//0, 1, 2 y 3.
            System.out.println("Has pulsado el botón 2");
        }else if(e.getSource()==boton3) {

            JOptionPane.showMessageDialog(Lamina_Cuadros_Dialogos.this, "Elige
opción", "V Confirmar",0);//0, 1 y 2.
            System.out.println("Has pulsado el botón 3");
        }else {

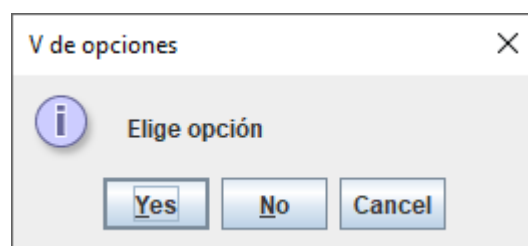
            JOptionPane.showOptionDialog(Lamina_Cuadros_Dialogos.this, "Elige
opción", "V de opciones", 1, 1, null, null, null);
            System.out.println("Has pulsado el botón 4");
        }
    }
}
```

Estas zonas marcadas son los botones 3 y 4.

Resultado botón 3:



Resultado botón 4:





## Ejercicio Práctico I (Vídeo 123)

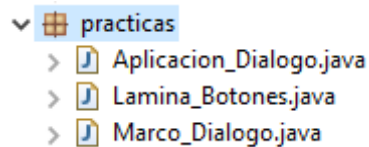
En los siguientes capítulos vamos a realizar una práctica guiada que como resultado final tendrá este formulario:



Para este proyecto vamos a realizar lo siguiente:

Seleccionamos el proyecto PrimerosPasos y creamos un paquete nuevo llamado prácticas.

Creamos el fichero que contiene el main llamado Aplicación\_Dialogos y un fichero llamado Marco\_Dialogos para la clase marco y otro llamado Lamina\_Botones para la clase lámia.



```
Aplicacion_Dialogos.java  Marco_Dialogos.java  Lamina_Botones.java
1 package practicas;
2 import javax.swing.*;
3 public class Marco_Dialogos extends JFrame{
4     public Marco_Dialogos() {
5         setTitle("Prueba de diálogo");
6         setBounds(500,300,600,450);
7     }
8
9 }
```

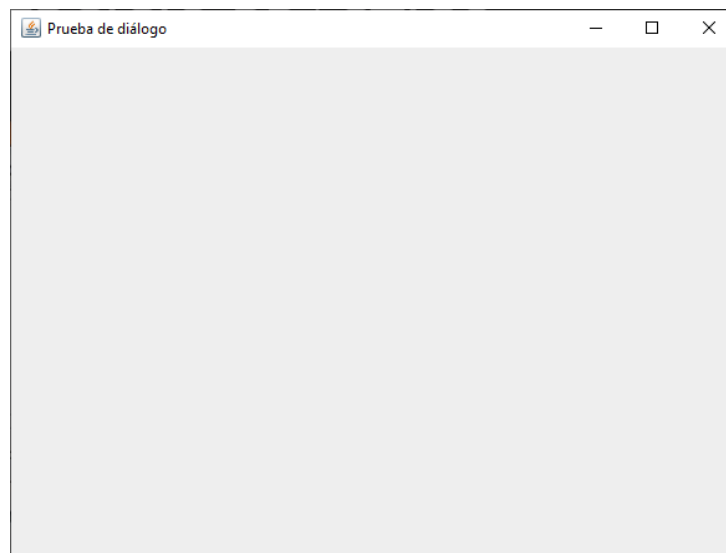
Empezamos seleccionando Marco\_Dialogos y escribimos el siguiente código.

```

1 package practicas;
2
3 import javax.swing.JFrame;
4
5 public class Aplicacion_Dialogos {
6
7     public static void main(String[] args) {
8         // TODO Auto-generated method stub
9         Marco_Dialogos mimarco=new Marco_Dialogos();
10        mimarco.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
11        mimarco.setVisible(true);
12
13    }
14 }

```

En el fichero Aplicación\_Dialogos escribimos el siguiente código, ahora ya estamos a disposición de poder ver el marco si ejecutamos.



```

1 package practicas;
2 import javax.swing.*;
3 import java.awt.*;
4 public class Marco_Dialogos extends JFrame{
5     public Marco_Dialogos() {
6         setTitle("Prueba de diálogo");
7         setBounds(500,300,600,450);
8         JPanel lamina_cuadrricula=new JPanel();
9         lamina_cuadrricula.setLayout(new GridLayout(2,3));
10    }
11
12 }

```

Esto es lo mismo que si en la clase Marco\_Dialogos escribimos lo siguiente:

JPanel milamina=new Lamina\_Superior();

Y luego una clase con el siguiente código:

```
Class Lamina_Superior extends JPanel(){  
    Public Lamina_Superior(){  
        setLayout(new GridLayout(2,3));  
    }  
}
```

Con este caso la lámina Lamina\_Superior tiene pocas instrucciones lo más cómodo será como está en el ejemplo anterior.





## Ejercicio Práctico II. (Vídeo 124)

BorderFactory: nos permite poner bordes de varios tipos y nos permite poner título.

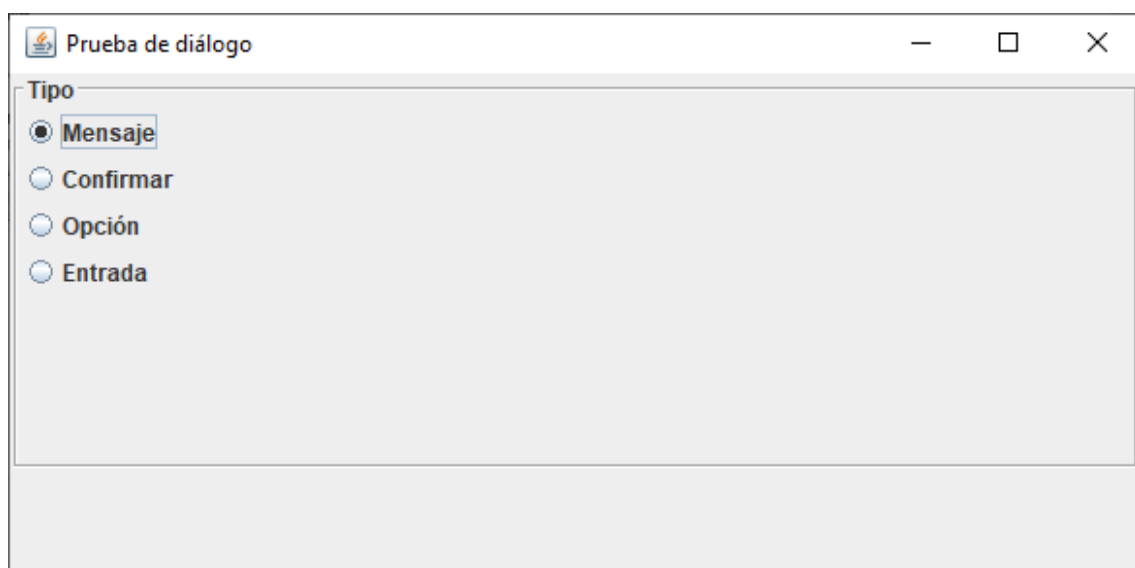
En el fichero Lamina\_Botones vamos a escribir el correspondiente código.

```
Aplicacion_Dialogos.java *Marco_Dialogos.java *Lamina_Botones.java
1 package practicas;
2 import javax.swing.*;
3 public class Lamina_Botones extends JPanel{
4     public Lamina_Botones(String titulo, String[]opciones) {
5         setBorder(BorderFactory.createTitledBorder(BorderFactory.createEtchedBorder(),titulo));
6         setLayout(new BorderLayout(this, BorderLayout.Y_AXIS));
7         ButtonGroup grupo=new ButtonGroup();
8         for (int i=0; i<opciones.length; i++) {
9             JRadioButton bot=new JRadioButton(opciones[i]);
10            add(bot);
11            grupo.add(bot);
12            bot.setSelected(i==0);
13        }
14    }
15 }
```

En el fichero Marco\_Dialogos agregamos el correspondiente código.

```
Aplicacion_Dialogos.java *Marco_Dialogos.java Lamina_Botones.java
1 package practicas;
2 import javax.swing.*;
3 import java.awt.*;
4 public class Marco_Dialogos extends JFrame{
5     public Marco_Dialogos() {
6         setTitle("Prueba de diálogo");
7         setBounds(500,300,600,450);
8         JPanel lamina_cuadrícula=new JPanel();
9         lamina_cuadrícula.setLayout(new GridLayout(2,3));
10        String primero[]= {"Mensaje", "Confirmar", "Opción", "Entrad
11        lamina_tipo = new Lamina_Botones("Tipo",primero);
12        lamina_cuadrícula.add(lamina_tipo);
13        add(lamina_cuadrícula);
14    }
15    private Lamina_Botones lamina_tipo;
16 }
```

Nos vamos al fichero Aplicación\_Dialigos y ejecutamos la aplicación:

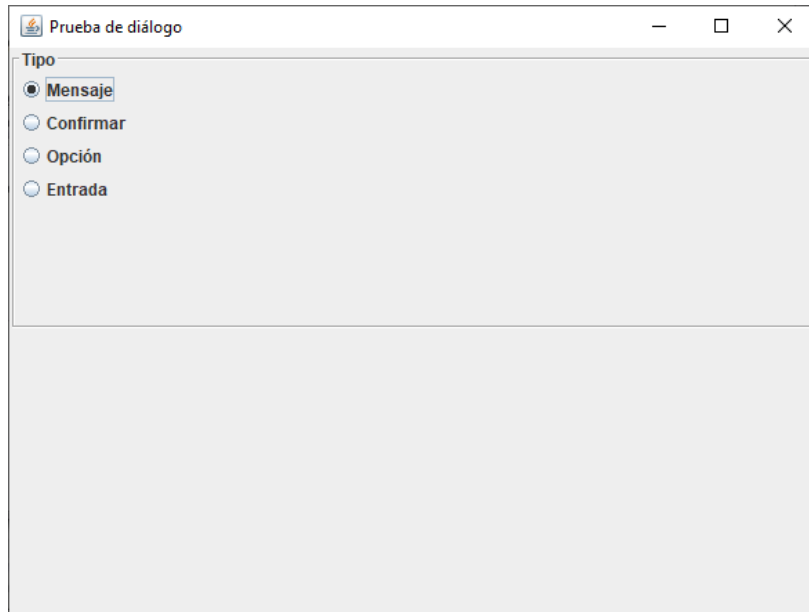


Ya tenemos el primer grupo, esto habrá que repetirlo con el resto de grupos.



## Ejercicio Práctico III. (Vídeo 125)

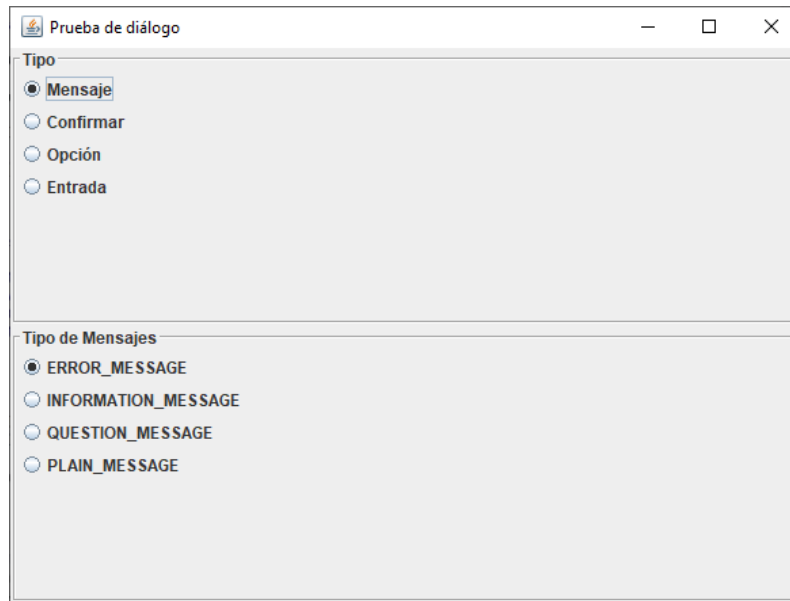
Esto es lo que conseguimos en el ejercicio anterior.



Ya tenemos un box y vamos a construir el resto de boxes.

```
1 package practicas;
2 import javax.swing.*;
3 import java.awt.*;
4 public class Marco_Dialogos extends JFrame{
5     public Marco_Dialogos() {
6         setTitle("Prueba de diálogo");
7         setBounds(500,300,600,450);
8         JPanel lamina_cuadrícula=new JPanel();
9         lamina_cuadrícula.setLayout(new GridLayout(2,3));
10        //-----
11        String primero[]= {"Mensaje", "Confirmar", "Opción", "Entrada"};
12        lamina_tipo = new Lamina_Botones("Tipo",primero);
13        lamina_cuadrícula.add(lamina_tipo);
14        //
15        lamina_tipo_mensaje=new Lamina_Botones("Tipo de Mensajes",new String[] {
16            "ERROR_MESSAGE",
17            "INFORMATION_MESSAGE",
18            "QUESTION_MESSAGE",
19            "PLAIN_MESSAGE",
20        });
21        lamina_cuadrícula.add(lamina_tipo_mensaje);
22        //-----
23
24        add(lamina_cuadrícula);
25        //-----
26
27    }
28    private Lamina_Botones lamina_tipo, lamina_tipo_mensaje;
29 }
30 }
```

Otra forma de agregar un box, ahorrando código.



Vamos a incluir el tercer Box.

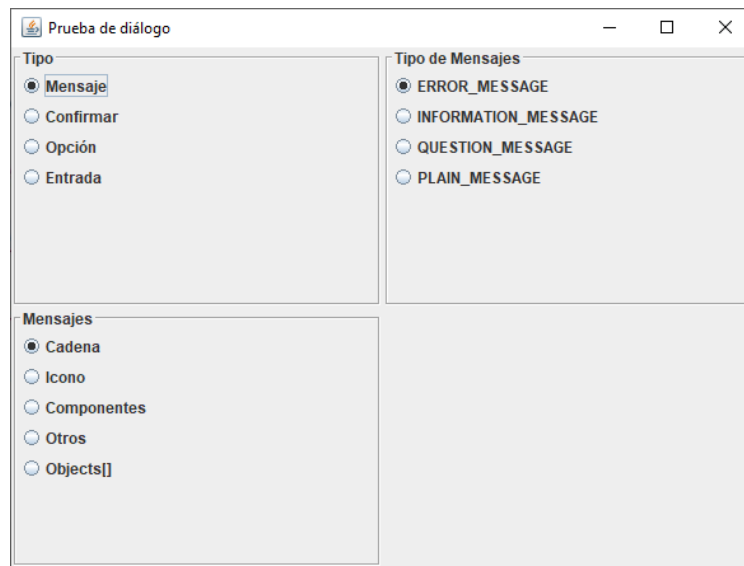
```

1 package practicas;
2 import javax.swing.*;
3 import java.awt.*;
4 public class Marco Dialogos extends JFrame{
5     public Marco_Dialogos() {
6         setTitle("Prueba de diálogo");
7         setBounds(500,300,600,450);
8         JPanel lamina_cuadrricula=new JPanel();
9         lamina_cuadrricula.setLayout(new GridLayout(2,3));
10        //-----
11        String primero[]= {"Mensaje","Confirmar", "Opción","Entrada"};
12        lamina_tipo = new Lamina_Botones("Tipo",primero);
13        lamina_cuadrricula.add(lamina_tipo);
14        //-----
15        lamina_tipo_mensaje=new Lamina_Botones("Tipo de Mensajes",new String[] {
16            "ERROR_MESSAGE",
17            "INFORMATION_MESSAGE",
18            "QUESTION_MESSAGE",
19            "PLAIN_MESSAGE"
20        });
21        lamina_cuadrricula.add(lamina_tipo_mensaje);
22        //-----
23        lamina_mensaje=new Lamina_Botones("Mensajes",new String[] {
24            "Cadena",
25            "Icono",
26            "Componentes",
27            "Otros",
28            "Objects[]"
29        });
30        lamina_cuadrricula.add(lamina_mensaje);
31        //-----
32        add(lamina_cuadrricula);
33        //-----
34
35    }
36    private Lamina_Botones lamina_tipo, lamina_tipo_mensaje, lamina_mensaje;
37 }
38 }

```

Incluimos el código remarcad.

Este será el resultado:



Ahora agregaremos el resto de Boxes.

```
package practicas;
import javax.swing.*;
import java.awt.*;
public class Marco_Dialogos extends JFrame{
    public Marco_Dialogos() {
        setTitle("Prueba de diálogo");
        setBounds(500,300,600,450);
        JPanel lamina_cuadrícula=new JPanel();
        lamina_cuadrícula.setLayout(new GridLayout(2,3));
        //-----
        String primero[]= {"Mensaje", "Confirmar", "Opción", "Entrada"};
        lamina_tipo = new Lamina_Botones("Tipo",primero);
        lamina_cuadrícula.add(lamina_tipo);
        //-----
        lamina_tipo_mensaje=new Lamina_Botones("Tipo de Mensajes",new
String[] {
            "ERROR_MESSAGE",
            "INFORMATION_MESSAGE",
            "WARNING_MESSAGE",
            "QUESTION_MESSAGE",
            "PLAIN_MESSAGE"
});
        lamina_cuadrícula.add(lamina_tipo_mensaje);
        //-----
        lamina_mensaje=new Lamina_Botones("Mensajes",new String[] {
            "Cadena",
            "Icono",
            "Componentes",
            "Otros",
            "Objects[]"
        });
        lamina_cuadrícula.add(lamina_mensaje);
```

```

//-----
String[] {
        lamina_tipo_opciones=new Lamina_Botones("Confirmar",new
                "DEFAULT_OPTION",
                "YES_NO_OPTION",
                "YES_NO_CANCEL_OPTION",
                "OK_CANCEL_OPTION"
        });
        lamina_cuadricula.add(lamina_tipo_opciones);
//-----
        lamina_opciones=new Lamina_Botones("Opción",new String[] {
                "String[]",
                "Icon[]",
                "Object[]"
        });
        lamina_cuadricula.add(lamina_opciones);
//-----
String[] {
        lamina_entrada=new Lamina_Botones("Entrada",new
                "Campo de texto",
                "Combo"
        });
        lamina_cuadricula.add(lamina_entrada);
//-----

        add(lamina_cuadricula);
//-----
}
private Lamina_Botones lamina_tipo, lamina_tipo_mensaje,
lamina_mensaje;
private Lamina_Botones lamina_tipo_opciones, lamina_opciones,
lamina_entrada;
}

```

Este será el resultado:

Tipo	Tipo de Mensajes	Mensajes
<input checked="" type="radio"/> Mensaje <input type="radio"/> Confirmar <input type="radio"/> Opción <input type="radio"/> Entrada	<input checked="" type="radio"/> ERROR_MESSAGE <input type="radio"/> INFORMATION_MESSAGE <input type="radio"/> WARNING_MESSAGE <input type="radio"/> QUESTION_MESSAGE <input type="radio"/> PLAIN_MESSAGE	<input checked="" type="radio"/> Cadena <input type="radio"/> Icono <input type="radio"/> Componentes <input type="radio"/> Otros <input type="radio"/> Objects[]
Confirmar	Opción	Entrada
<input checked="" type="radio"/> DEFAULT_OPTION <input type="radio"/> YES_NO_OPTION <input type="radio"/> YES_NO_CANCEL_OPTION <input type="radio"/> OK_CANCEL_OPTION	<input checked="" type="radio"/> String[] <input type="radio"/> Icon[] <input type="radio"/> Object[]	<input checked="" type="radio"/> Campo de texto <input type="radio"/> Combo

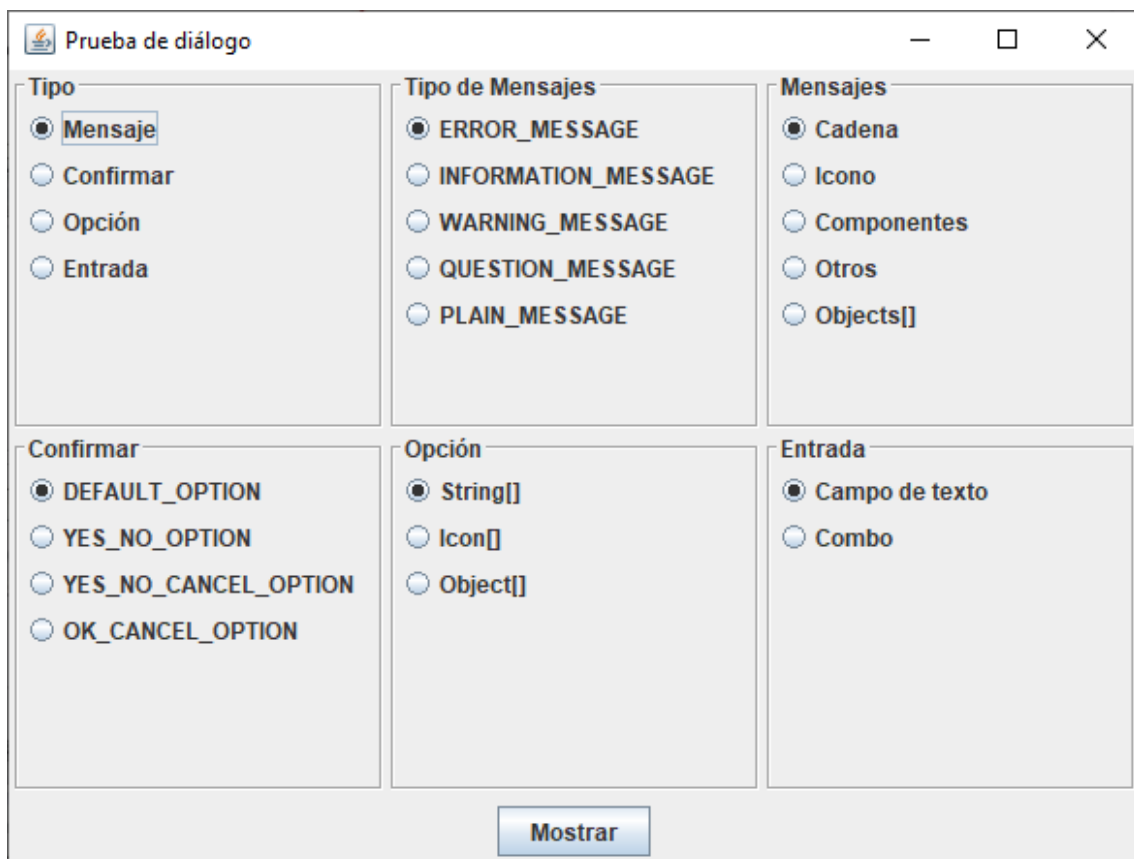
```

54     setLayout(new BorderLayout());
55     //-----
56     JPanel lamina_mostrar=new JPanel();
57     JButton boton_mostrar=new JButton ("Mostrar");
58     lamina_mostrar.add(boton_mostrar);
59
60     add(lamina_mostrar,BorderLayout.SOUTH);
61     add(lamina_cuadrícula,BorderLayout.CENTER);
62     //-----
63 }
64 private Lamina_Botones lamina_tipo, lamina_tipo_mensaje, lamina_mensaje;
65 private Lamina_Botones lamina_tipo_opciones, lamina_opciones, lamina_entrada;
66 }

```

Para distribuir los marcos en la parte superior usamos BorderLayout.CENTER y el botón para situarlo en la parte inferior BorderLayout.SOUTH.

Este será el resultado:







## Ejercicio Práctico IV. (Vídeo 126)

Vamos a dar funcionalidad al botón Mostrar, en esta primera parte vamos a imprimir en consola las opciones que tenemos seleccionadas.

```
1 package practicas;
2 import javax.swing.*;
3 public class Lamina_Botones extends JPanel{
4     public Lamina_Botones(String titulo, String[]opciones) {
5         setBorder(BorderFactory.createTitledBorder(BorderFactory.createEtchedBorder(),titulo));
6         setLayout(new BorderLayout(this, BorderLayout.Y_AXIS));
7         grupo=new ButtonGroup();
8         for (int i=0; i<opciones.length; i++) {
9             JRadioButton bot=new JRadioButton(opciones[i]);
10            bot.setActionCommand(opciones[i]);
11            add(bot);
12            grupo.add(bot);
13            bot.setSelected(i==0);
14        }
15    }
16    public String dameSeleccion() {
17        /*ButtonModel miboton=grupo.getSelection();
18        String s=miboton.getActionCommand();
19        return s;*/
20
21        return grupo.getSelection().getActionCommand(); //Es lo mismo
22    }
23    private ButtonGroup grupo;
24 }
```

En el fichero Lamina\_botones agregamos el siguiente código.

```
58     JPanel lamina_mostrar=new JPanel();
59     JButton boton_mostrar=new JButton ("Mostrar");
60     boton_mostrar.addActionListener(new AccionMostrar());
61     lamina_mostrar.add(boton_mostrar);
62
63     add(lamina_mostrar, BorderLayout.SOUTH);
64     add(lamina_cuadrícula, BorderLayout.CENTER);
65     //-----
66 }
67 private class AccionMostrar implements ActionListener{
68
69     @Override
70     public void actionPerformed(ActionEvent e) {
71         // TODO Auto-generated method stub
72         System.out.println("Funciona");
73     }
74
75 }
76 private Lamina_Botones lamina_tipo, lamina_tipo_mensaje, lamina_mensaje;
77 private Lamina_Botones lamina_tipo_opciones, lamina_opciones, lamina_entrada;
78 }
```

En el fichero Marcos\_Dialogos escribimos el siguiente código, si al seleccionar el botón por consola tiene que imprimir "Funciona".

```

66     }
67     private class AccionMostrar implements ActionListener{
68
69         @Override
70         public void actionPerformed(ActionEvent e) {
71             // TODO Auto-generated method stub
72             System.out.println(lamina_tipo.dameSeleccion()); ←
73         }
74
75     }
76     private Lamina_Botones lamina_tipo, lamina_tipo_mensaje, lamina_mensaje;
77     private Lamina_Botones lamina_tipo_opciones, lamina_opciones, lamina_entrada;
78 }

```

Cambiamos el contenido del System.out.println("Funciona") por el que muestra en la flecha.

Ahora si seleccionas del recuadro Tipo Mensaje, Confirmar, opción y Entrada, este será el mensaje que se muestra en consola:

```

Mensaje
Confirmar
Opción
Entrada

```

Si queremos imprimir todos los botones que tenemos seleccionados.

```

67     private class AccionMostrar implements ActionListener{
68
69         @Override
70         public void actionPerformed(ActionEvent e) {
71             // TODO Auto-generated method stub
72             System.out.print(lamina_tipo.dameSeleccion()+" ");
73             System.out.print(lamina_tipo_mensaje.dameSeleccion()+" ");
74             System.out.print(lamina_mensaje.dameSeleccion()+" ");
75             System.out.print(lamina_tipo_opciones.dameSeleccion()+" ");
76             System.out.print(lamina_opciones.dameSeleccion()+" ");
77             System.out.println(lamina_entrada.dameSeleccion()+" ");
78         }
79
80     }
81     private Lamina_Botones lamina_tipo, lamina_tipo_mensaje, lamina_mensaje;
82     private Lamina_Botones lamina_tipo_opciones, lamina_opciones, lamina_entrada;
83 }

```

Este será el resultado:

```

Mensaje      ERROR_MESSAGE      Cadena      DEFAULT_OPTION      String[]      Campo de texto
Entrada      PLAIN_MESSAGE        Objects[]   OK_CANCEL_OPTION    Icon[]        Combo

```





## Ejercicio Práctico V. (Vídeo 127)

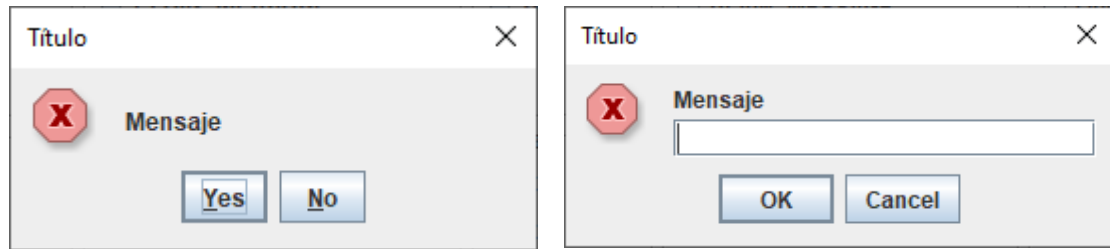
```
67 private class AccionMostrar implements ActionListener{
68
69     @Override
70     public void actionPerformed(ActionEvent e) {
71         // TODO Auto-generated method stub
72         System.out.println(lamina_tipo.dameSeleccion()+" ");
73         System.out.println(lamina_tipo_mensaje.dameSeleccion()+" ");
74         System.out.println(lamina_mensaje.dameSeleccion()+" ");
75         System.out.println(lamina_tipo_opciones.dameSeleccion()+" ");
76         System.out.println(lamina_opciones.dameSeleccion()+" ");
77         System.out.println(lamina_entrada.dameSeleccion()+" ");
78
79         if(lamina_tipo.dameSeleccion().equals("Mensaje")) {
80             JOptionPane.showMessageDialog(Marco_Dialogos.this,"Mensaje", "Título",0);
81         }
82         else if(lamina_tipo.dameSeleccion().equals("Confirmar")){
83             JOptionPane.showConfirmDialog(Marco_Dialogos.this, "Mensaje", "Título", 0,0);
84         }
85         else if(lamina_tipo.dameSeleccion().equals("Opción")){
86             JOptionPane.showOptionDialog(Marco_Dialogos.this, "Mensaje", "Título", 0,0,null,null,null);
87         }
88         else if(lamina_tipo.dameSeleccion().equals("Entrada")){
89             JOptionPane.showInputDialog(Marco_Dialogos.this,"Mensaje","Título",0);
90         }
91     }
92
93 }
94 private Lamina_Botones lamina_tipo, lamina_tipo_mensaje, lamina_mensaje;
95 private Lamina_Botones lamina_tipo_opciones, lamina_opciones, lamina_entrada;
96 }
```

En el fichero Marco\_Dialogos agregaremos el siguiente código que detallamos a continuación para su mejor lectura:

```
if(lamina_tipo.dameSeleccion().equals("Mensaje")) {
    JOptionPane.showMessageDialog(Marco_Dialogos.this,"Mensaje",
    "Título",0);
}
else if(lamina_tipo.dameSeleccion().equals("Confirmar")){
    JOptionPane.showConfirmDialog(Marco_Dialogos.this,
    "Mensaje", "Título", 0,0);
}
else if(lamina_tipo.dameSeleccion().equals("Opción")){
    JOptionPane.showOptionDialog(Marco_Dialogos.this,
    "Mensaje", "Título", 0,0,null,null,null);
}
else if(lamina_tipo.dameSeleccion().equals("Entrada")){
    JOptionPane.showInputDialog(Marco_Dialogos.this,"Mensaje","Título",0);
}
}
```

Si ejecutamos la aplicación las opciones del marco Tipo observaremos las siguientes ventanas:





Vamos a crear las siguientes variables en el fichero Marco\_Dialogos.

```

95     private Lamina_Botones lamina_tipo, lamina_tipo_mensaje, lamina_mensaje;
96     private Lamina_Botones lamina_tipo_opciones, lamina_opciones, lamina_entrada;
97     private String cadenaMensaje="Mensaje";
98     private Icon iconoMensaje=new ImageIcon("src/graficos/bolaAzul.png");
99     private Object objetoMensaje=new Date();
100    private Component ComponenteMensaje=new Lamina_Ejemplo();|
101 }

```

Declaramos las siguientes variables fuera del constructor de la clase AccionMostrar.

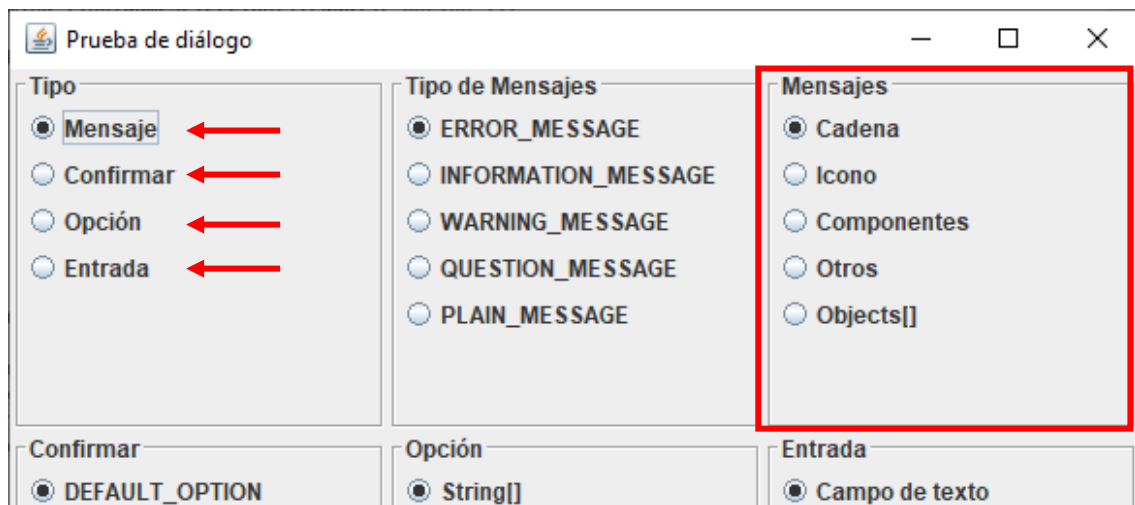
Lamina\_Ejemplo nos señala un error, es porque la clase Lamina\_Ejemplo aun no la hemos declarado, en el siguiente capitulo será lo primero que haremos.



## Ejercicio Práctico VI. (Vídeo 128)

Al final del fichero Marco\_Dialogos vamos a crear una clase de tipo lamina donde dibujaremos un rectángulo que ocupe las dimensiones de la lamina y esté pintado de color amarillo, este es el correspondiente código:

```
130 class Lamina_Ejemplo extends JPanel{
131     public void paintComponent(Graphics g) {
132         super.paintComponents(g);
133         Graphics2D g2=(Graphics2D) g;
134         Rectangle2D rectangulo = new Rectangle2D.Double(0,0,getWidth(), getHeight());
135         g2.setPaint(Color.YELLOW);
136         g2.fill(rectangulo);
137     }
138 }
```



Ahora vamos a crear un método que nos devuelva cadena, icono, componentes, otros o objects[], porque las opciones salen con cualquier tipo de ventana. Haciendo un método evitamos que tener que reprogramas por cada ventana.

El código lo insertaremos después del constructor del fichero Marco\_Dialogos.

```
69     public Object dameMensaje() {
70         String s=lamina_mensaje.dameSeleccion();
71         if(s.equals("Cadena")) {
72             return cadenaMensaje;
73         }else if(s.equals("Icono")){
74             return iconoMensaje;
75         }else if(s.equals("Componentes")) {
76             return ComponenteMensaje;
77         }else if(s.equals("Otros")) {
78             return objetoMensaje;
79         }else if(s.equals("Objects[]")) {
80             return new Object[] {
81                 cadenaMensaje,
82                 iconoMensaje,
83                 ComponenteMensaje,
84                 objetoMensaje
85             };
86         }else {
87             return null;
88         }
89     }
```

dameSeleccion() es un método que se encuentra en el fichero Lamina\_Botones.

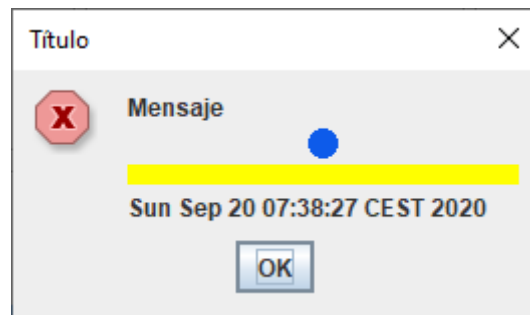
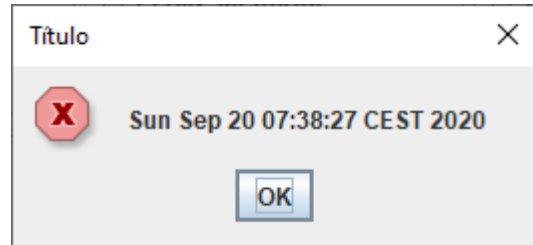
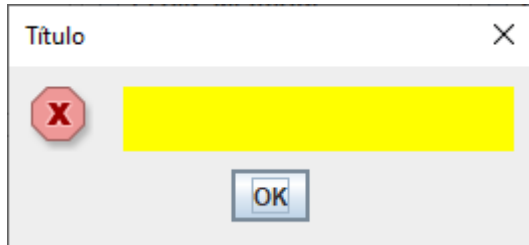
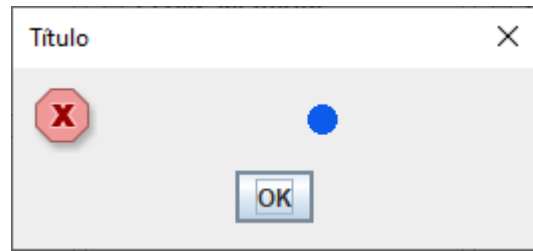
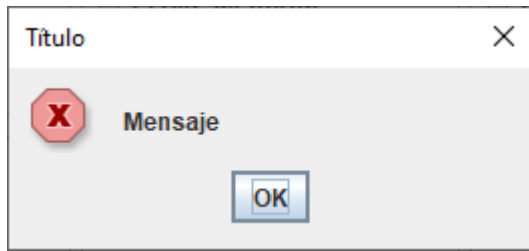
El else con return null estamos obligado a ponerlo ya que el compilador dice que esta obligado a retornar algo y si para el no se cumple ninguna de las condiciones anteriores que no es el caso este método no retornaría nada, solo es para quitar el error.

```
95 private class AccionMostrar implements ActionListener{
96
97     @Override
98     public void actionPerformed(ActionEvent e) {
99         // TODO Auto-generated method stub
100         System.out.print(lamina_tipo.dameSeleccion()+" ");
101         System.out.print(lamina_tipo_mensaje.dameSeleccion()+" ");
102         System.out.print(lamina_mensaje.dameSeleccion()+" ");
103         System.out.print(lamina_tipo_opciones.dameSeleccion()+" ");
104         System.out.print(lamina_opciones.dameSeleccion()+" ");
105         System.out.println(lamina_entrada.dameSeleccion()+" ");
106
107         if(lamina_tipo.dameSeleccion().equals("Mensaje")) {
108             JOptionPane.showMessageDialog(Marco_Dialogos.this,dameMensaje(), "Título",0);
109         }
110         else if(lamina_tipo.dameSeleccion().equals("Confirmar")){
111             JOptionPane.showConfirmDialog(Marco_Dialogos.this, dameMensaje(), "Título", 0,0);
112         }
113         else if(lamina_tipo.dameSeleccion().equals("Opción")){
114             JOptionPane.showOptionDialog(Marco_Dialogos.this, dameMensaje(), "Título", 0,0,null,null,null);
115         }
116         else if(lamina_tipo.dameSeleccion().equals("Entrada")){
117             JOptionPane.showInputDialog(Marco_Dialogos.this,dameMensaje(),"Título",0);
118         }
119     }
120
121 }
```

En el método AccionMostrar cambiamos “Mensaje” por dameMensaje(), lo adjunto de nuevo para que se pueda visualizar mejor.

```
        if(lamina_tipo.dameSeleccion().equals("Mensaje")) {
            JOptionPane.showMessageDialog(Marco_Dialogos.this,dameMensaje(),
            "Título",0);
        }
        else if(lamina_tipo.dameSeleccion().equals("Confirmar")){
            JOptionPane.showConfirmDialog(Marco_Dialogos.this,
            dameMensaje(), "Título", 0,0);
        }
        else if(lamina_tipo.dameSeleccion().equals("Opción")){
            JOptionPane.showOptionDialog(Marco_Dialogos.this,
            dameMensaje(), "Título", 0,0,null,null,null);
        }
        else if(lamina_tipo.dameSeleccion().equals("Entrada")){
            JOptionPane.showInputDialog(Marco_Dialogos.this,dameMensaje(),"Título"
            ,0);
        }
    }
```

Ahora seleccionando del marco tipo la opción Menaje, activaremos cada una de las opciones de Mensaje para ver el resultado:





## Ejercicio Práctico VII. (Vídeo 129)

Ahora vamos a cambiar el tipo de mensaje:

The screenshot shows a Java Swing dialog box titled "Prueba de diálogo". It contains several sections for configuring a dialog box:

- Tipo:** Radio buttons for "Mensaje" (selected), "Confirmar", "Opción", and "Entrada".
- Tipo de Mensajes:** Radio buttons for "ERROR\_MESSAGE" (selected), "INFORMATION\_MESSAGE", "WARNING\_MESSAGE", "QUESTION\_MESSAGE", and "PLAIN\_MESSAGE". This section is highlighted with a red box.
- Mensajes:** Radio buttons for "Cadena" (selected), "Icono", "Componentes", "Otros", and "Objects[]".
- Confirmar:** Radio buttons for "DEFAULT\_OPTION" (selected), "YES\_NO\_OPTION", "YES\_NO\_CANCEL\_OPTION", and "OK\_CANCEL\_OPTION".
- Opción:** Radio buttons for "String[]" (selected), "Icon[]", and "Object[]".
- Entrada:** Radio buttons for "Campo de texto" (selected) and "Combo".

A "Mostrar" button is located at the bottom center of the dialog.

Nos colocaremos a continuación del método `dameMensaje()` del fichero `Marco_Dialogos` donde escribiremos el siguiente método:

```
91 public int dameIcono(){
92     String s=lamina_tipo_mensaje.dameSeleccion();
93     if(s.equals("ERROR_MESSAGE")) {
94         return JOptionPane.ERROR_MESSAGE; //Tambien puede retornar un 0
95     }
96     else if(s.equals("INFORMATION_MESSAGE")){
97         return JOptionPane.INFORMATION_MESSAGE; //Tambien puede retornar un 1
98     }
99     else if(s.equals("WARNING_MESSAGE")) {
100         return JOptionPane.WARNING_MESSAGE; //Tambien puede retornar un 2
101     }
102     else if(s.equals("QUESTION_MESSAGE")) {
103         return JOptionPane.QUESTION_MESSAGE; //Tambien puede retornar un 3
104     }
105     else if(s.equals("PLAIN_MESSAGE")) {
106         return JOptionPane.PLAIN_MESSAGE; //También puede retornar un -1
107     }
108     else {
109         return 0;
110     }
111 }
```

Ahora nos vamos a la clase `AccionMostrar`.

```

115 private class AccionMostrar implements ActionListener{
116
117     @Override
118     public void actionPerformed(ActionEvent e) {
119         // TODO Auto-generated method stub
120         System.out.print(lamina_tipo.dameSeleccion()+" ");
121         System.out.print(lamina_tipo_mensaje.dameSeleccion()+" ");
122         System.out.print(lamina_mensaje.dameSeleccion()+" ");
123         System.out.print(lamina_tipo_opciones.dameSeleccion()+" ");
124         System.out.print(lamina_opciones.dameSeleccion()+" ");
125         System.out.println(lamina_entrada.dameSeleccion()+" ");
126
127         if(lamina_tipo.dameSeleccion().equals("Mensaje")) {
128             JOptionPane.showMessageDialog(Marco_Dialogos.this,dameMensaje(), "Título",dameIcono());
129         }
130         else if(lamina_tipo.dameSeleccion().equals("Confirmar")){
131             JOptionPane.showConfirmDialog(Marco_Dialogos.this, dameMensaje(), "Título",0,dameIcono());
132         }
133         else if(lamina_tipo.dameSeleccion().equals("Opción")){
134             JOptionPane.showOptionDialog(Marco_Dialogos.this, dameMensaje(), "Título", 0,dameIcono(),null,null,null);
135         }
136         else if(lamina_tipo.dameSeleccion().equals("Entrada")){
137             JOptionPane.showInputDialog(Marco_Dialogos.this,dameMensaje(),"Título",dameIcono());
138         }
139     }
140 }

```

Adjunto el texto ampliado:

```

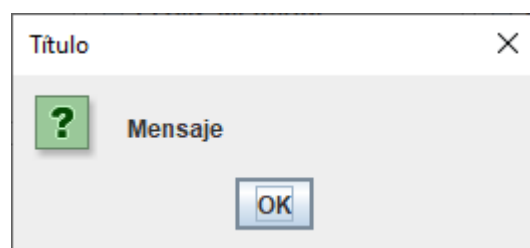
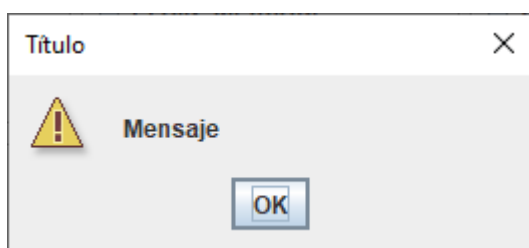
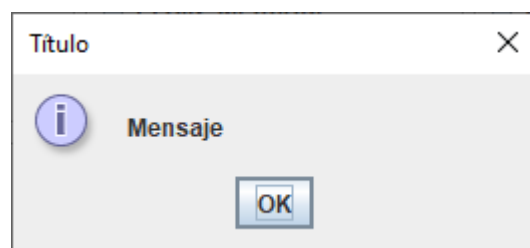
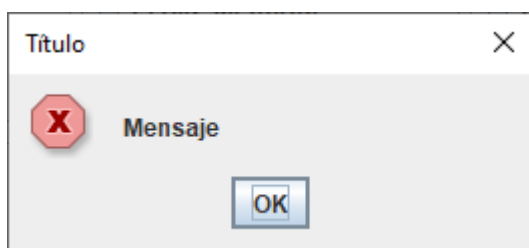
if(lamina_tipo.dameSeleccion().equals("Mensaje")) {

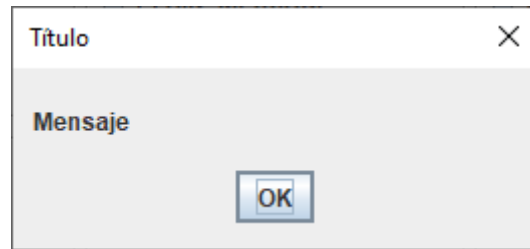
    JOptionPane.showMessageDialog(Marco_Dialogos.this,dameMensaje(),
    "Título",dameIcono());
    }
    else if(lamina_tipo.dameSeleccion().equals("Confirmar")){
        JOptionPane.showConfirmDialog(Marco_Dialogos.this,
dameMensaje(), "Título",0,dameIcono());
    }
    else if(lamina_tipo.dameSeleccion().equals("Opción")){
        JOptionPane.showOptionDialog(Marco_Dialogos.this,
dameMensaje(), "Título", 0,dameIcono(),null,null,null);
    }
    else if(lamina_tipo.dameSeleccion().equals("Entrada")){

        JOptionPane.showInputDialog(Marco_Dialogos.this,dameMensaje(),"Título"
,dameIcono());
    }
}

```

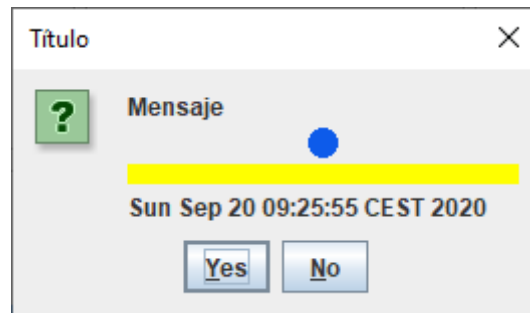
Ahora probamos la aplicación teniendo activada en tipo Menaje, en Mensajes Cadena, vamos a probar las 5 opciones de Tipo de Mensaje.





Hasta ahora ya podemos combinar opciones de Tipo, Tipo Mensaje y Mensaje.

Si del grupo Tipo seleccionamos Confirmar, del grupo Mensaje QUESTION\_MESSAGE y de Mensaje Object[], este será el resultado:



## Ejercicio Práctico VIII. (Vídeo 130)

Prueba de diálogo

Tipo

- Mensaje
- Confirmar
- Opción
- Entrada

Tipo de Mensajes

- ERROR\_MESSAGE
- INFORMATION\_MESSAGE
- WARNING\_MESSAGE
- QUESTION\_MESSAGE
- PLAIN\_MESSAGE

Mensajes

- Cadena
- Icono
- Componentes
- Otros
- Objects[]

Confirmar

- DEFAULT\_OPTION
- YES\_NO\_OPTION
- YES\_NO\_CANCEL\_OPTION
- OK\_CANCEL\_OPTION

Opción

- String[]
- Icon[]
- Object[]

Entrada

- Campo de texto
- Combo

Mostrar

Vamos a gestionar la lámina Confirmar, esto solo afectará si activamos la opción Confirmar de la lámina Tipo.

Vamos a cambiar el método `dameIcono` por `dameTipo` ya que lo vamos a reutilizar para controlar también las opciones de la lámina Confirmar.

```
public int dameTipo(Lamina_Botones lamina){
    String s=lamina.dameSeleccion();
    if(s.equals("ERROR_MESSAGE")) {
        return JOptionPane.ERROR_MESSAGE; //Tambien puede retornar un 0
    }
    else if(s.equals("INFORMATION_MESSAGE")){
        return JOptionPane.INFORMATION_MESSAGE; //Tambien puede retornar un 1
    }
    else if(s.equals("WARNING_MESSAGE")) {
        return JOptionPane.WARNING_MESSAGE; //Tambien puede retornar un 2
    }
    else if(s.equals("QUESTION_MESSAGE")) {
        return JOptionPane.QUESTION_MESSAGE; //Tambien puede retornar un 3
    }
    else if(s.equals("PLAIN_MESSAGE")) {
        return JOptionPane.PLAIN_MESSAGE; //Tambien puede retornar un -1
    }
    else {
        return 0;
    }
}
```

A este método le pasamos el parámetro `lamina` de tipo `Lamina_Botones`.

```

115 private class AccionMostrar implements ActionListener{
116
117     @Override
118     public void actionPerformed(ActionEvent e) {
119         // TODO Auto-generated method stub
120         System.out.print(lamina_tipo.dameSeleccion()+" ");
121         System.out.print(lamina_tipo_mensaje.dameSeleccion()+" ");
122         System.out.print(lamina_mensaje.dameSeleccion()+" ");
123         System.out.print(lamina_tipo_opciones.dameSeleccion()+" ");
124         System.out.print(lamina_opciones.dameSeleccion()+" ");
125         System.out.println(lamina_entrada.dameSeleccion()+" ");
126
127         if(lamina_tipo.dameSeleccion().equals("Mensaje")) {
128             JOptionPane.showMessageDialog(Marco_Dialogos.this,dameMensaje(), "Título",dameTipo (lamina_tipo_mensaje));
129         }
130         else if(lamina_tipo.dameSeleccion().equals("Confirmar")){
131             JOptionPane.showConfirmDialog(Marco_Dialogos.this, dameMensaje(), "Título",0,dameTipo (lamina_tipo_mensaje));
132         }
133         else if(lamina_tipo.dameSeleccion().equals("Opción")){
134             JOptionPane.showOptionDialog(Marco_Dialogos.this, dameMensaje(), "Título", 0,dameTipo ([lamina_tipo_mensaje],null,null,null));
135         }
136         else if(lamina_tipo.dameSeleccion().equals("Entrada")){
137             JOptionPane.showInputDialog(Marco_Dialogos.this,dameMensaje(),"Título",dameTipo(lamina_tipo_mensaje));
138         }
139     }
140 }

```

Cambiamos el nombre del método en la clase AccionMostrar, pasándoles el respectivo parámetro, adjunto texto ampliado:

```

if(lamina_tipo.dameSeleccion().equals("Mensaje")) {
    JOptionPane.showMessageDialog(Marco_Dialogos.this,dameMensaje(),
    "Título",dameTipo (lamina_tipo_mensaje));
}
else if(lamina_tipo.dameSeleccion().equals("Confirmar")){
    JOptionPane.showConfirmDialog(Marco_Dialogos.this,
dameMensaje(), "Título",0,dameTipo (lamina_tipo_mensaje));
}
else if(lamina_tipo.dameSeleccion().equals("Opción")){
    JOptionPane.showOptionDialog(Marco_Dialogos.this,
dameMensaje(), "Título", 0,dameTipo (lamina_tipo_mensaje),null,null,null);
}
else if(lamina_tipo.dameSeleccion().equals("Entrada")){
    JOptionPane.showInputDialog(Marco_Dialogos.this,dameMensaje(),"Título"
,dameTipo(lamina_tipo_mensaje));
}

```

Ya tiene que funcionar cambiando las opciones de "Tipo de Mensaje".

```

115 private class AccionMostrar implements ActionListener{
116
117     @Override
118     public void actionPerformed(ActionEvent e) {
119         // TODO Auto-generated method stub
120         System.out.print(lamina_tipo.dameSeleccion()+" ");
121         System.out.print(lamina_tipo_mensaje.dameSeleccion()+" ");
122         System.out.print(lamina_mensaje.dameSeleccion()+" ");
123         System.out.print(lamina_tipo_opciones.dameSeleccion()+" ");
124         System.out.print(lamina_opciones.dameSeleccion()+" ");
125         System.out.println(lamina_entrada.dameSeleccion()+" ");
126
127         if(lamina_tipo.dameSeleccion().equals("Mensaje")) {
128             JOptionPane.showMessageDialog(Marco_Dialogos.this,dameMensaje(), "Título",dameTipo (lamina_tipo_mensaje));
129         }
130         else if(lamina_tipo.dameSeleccion().equals("Confirmar")){
131             JOptionPane.showConfirmDialog(Marco_Dialogos.this, dameMensaje(), "Título",dameTipo(lamina_tipo_opciones),dameTipo (lamina_tipo_mensaje));
132         }
133         }
134         else if(lamina_tipo.dameSeleccion().equals("Opción")){
135             JOptionPane.showOptionDialog(Marco_Dialogos.this, dameMensaje(), "Título", 0,dameTipo (lamina_tipo_mensaje),null,null,null);
136         }
137         }
138         else if(lamina_tipo.dameSeleccion().equals("Entrada")){
139             JOptionPane.showInputDialog(Marco_Dialogos.this,dameMensaje(),"Título",dameTipo(lamina_tipo_mensaje));
140         }
141     }
142 }

```

Aquí la línea modificada.

```

JOptionPane.showConfirmDialog(Marco_Dialogos.this, dameMensaje(),
"Título",dameTipo(lamina tipo opciones),dameTipo (lamina_tipo_mensaje));

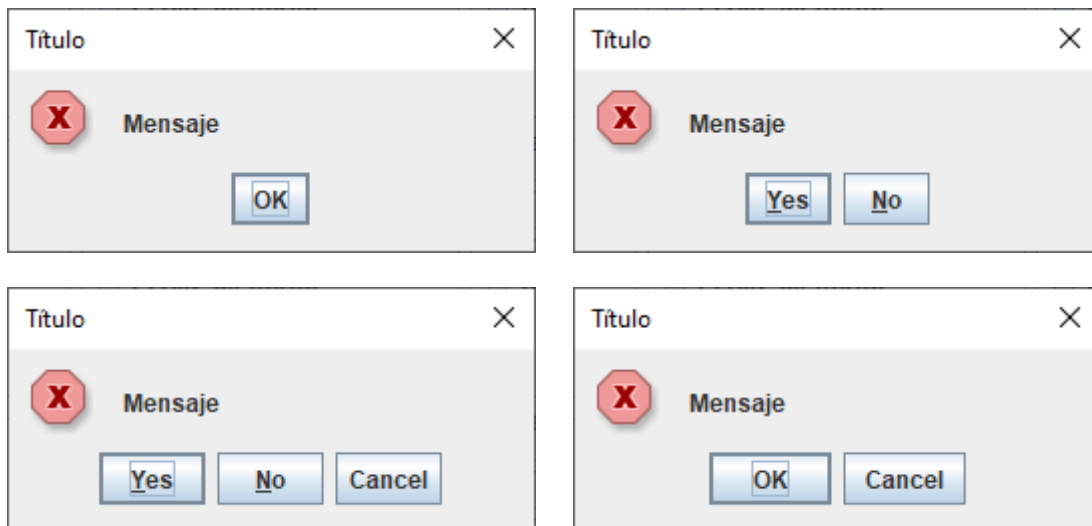
```

```

91 public int dameTipo(Lamina_Botones lamina){
92     String s=lamina.dameSeleccion();
93     if(s.equals("ERROR_MESSAGE")||s.equals("YES_NO_OPTION")) {
94         return 0;
95     }
96     else if(s.equals("INFORMATION_MESSAGE")||s.equals("YES_NO_CANCEL_OPTION")){
97         return 1;
98     }
99     else if(s.equals("WARNING_MESSAGE")||s.equals("OK_CANCEL_OPTION")) {
100         return 2;
101     }
102     else if(s.equals("QUESTION_MESSAGE")) {
103         return 3;
104     }
105     else if(s.equals("PLAIN_MESSAGE") || s.equals("DEFAULT_OPTION")) {
106         return -1;
107     }
108     else {
109         return 0;
110     }
111 }

```

Modificamos el código de la clase dameTipo para controlar las opciones de la lámina confirmar.





## Ejercicio Práctico IX. (Vídeo 131)

```
115@ private class AccionMostrar implements ActionListener{
116
117@     @Override
118     public void actionPerformed(ActionEvent e) {
119         // TODO Auto-generated method stub
120         System.out.println(lamina_tipo.dameSeleccion()+" ");
121         System.out.println(lamina_tipo_mensaje.dameSeleccion()+" ");
122         System.out.println(lamina_mensaje.dameSeleccion()+" ");
123         System.out.println(lamina_tipo_opciones.dameSeleccion()+" ");
124         System.out.println(lamina_opciones.dameSeleccion()+" ");
125         System.out.println(lamina_entrada.dameSeleccion()+" ");
126
127         if(lamina_tipo.dameSeleccion().equals("Mensaje")) {
128             JOptionPane.showMessageDialog(Marco_Dialogos.this,dameMensaje(), "Título",dameTipo (lamina_tipo_mensaje));
129         }
130         else if(lamina_tipo.dameSeleccion().equals("Confirmar")){
131             JOptionPane.showConfirmDialog(Marco_Dialogos.this, dameMensaje(), "Título",dameTipo(lamina_tipo_opciones),dameTipo (lamina_tipo_mensaje));
132         }
133         }
134         else if(lamina_tipo.dameSeleccion().equals("Opción")){
135             JOptionPane.showOptionDialog(Marco_Dialogos.this, dameMensaje(), "Título", 1,dameTipo (lamina_tipo_mensaje),null,null,null);
136         }
137         }
138         else if(lamina_tipo.dameSeleccion().equals("Entrada")){
139             JOptionPane.showInputDialog(Marco_Dialogos.this,dameMensaje(),"Título",dameTipo(lamina_tipo_mensaje));
140         }
141     }
142 }
```

Lo cambiamos a 1.

```
else if(lamina_tipo.dameSeleccion().equals("Opción")){
    JOptionPane.showOptionDialog(Marco_Dialogos.this,
dameMensaje(), "Título", 1,dameTipo (lamina_tipo_mensaje),null,null,null);
}
```

A continuación cambiamos la siguiente instrucción.

```
else if(lamina_tipo.dameSeleccion().equals("Opción")){
    JOptionPane.showOptionDialog(Marco_Dialogos.this,
dameMensaje(), "Título", 1,dameTipo
(lamina_tipo_mensaje),null,dameOpciones(lamina_opciones),null);
}
```

Si seleccionamos en la lamina Tipo y en la lámina Opción String este será el resultado cuando seleccionemos el botón Mostrar.





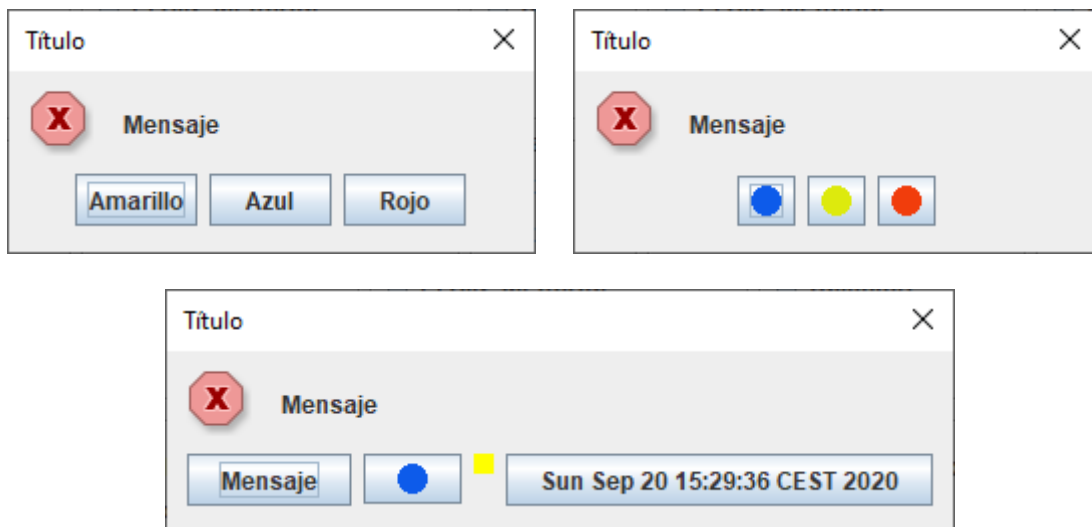
```

113 public Object[] dameOpciones(Lamina_Botones lamina) {
114     String s=lamina.dameSeleccion();
115     if(s.equals("String[]")) {
116         return new String[] {"Amarillo","Azul","Rojo"};
117     }else if(s.equals("Icon[]")) {
118         return new Object[] {new ImageIcon("src/graficos/bolaAzul.png"),
119                             new ImageIcon("src/graficos/bolaAmarilla.png"),
120                             new ImageIcon("src/graficos/bolaRoja.png")};
121
122     }else if(s.equals("Object[]")){
123         return new Object[] {cadenaMensaje,
124                             iconoMensaje,
125                             ComponenteMensaje,
126                             objetoMensaje
127     };
128     }
129     else {
130         return null;
131     }
132 }

```

Creamos el siguiente método de tipo array en el fichero Marco\_Dialogos llamado dameOpciones de tipo Object[].

Si ejecutamos y en la lámina Tipo seleccionamos Opción y en la Opción probamos String[], Icon[] y Object[], este será el resultado:



```

134 private class AccionMostrar implements ActionListener{
135
136 @Override
137 public void actionPerformed(ActionEvent e) {
138 // TODO Auto-generated method stub
139 System.out.println(lamina_tipo.dameSeleccion()+" ");
140 System.out.println(lamina_tipo_mensaje.dameSeleccion()+" ");
141 System.out.println(lamina_mensaje.dameSeleccion()+" ");
142 System.out.println(lamina_tipo_opciones.dameSeleccion()+" ");
143 System.out.println(lamina_opciones.dameSeleccion()+" ");
144 System.out.println(lamina_entrada.dameSeleccion()+" ");
145
146 if(lamina_tipo.dameSeleccion().equals("Mensaje")) {
147 JOptionPane.showMessageDialog(Marco_Dialogos.this,dameMensaje(), "Titulo",dameTipo (lamina_tipo_mensaje));
148 }
149 else if(lamina_tipo.dameSeleccion().equals("Confirmar")){
150
151 JOptionPane.showConfirmDialog(Marco_Dialogos.this, dameMensaje(), "Titulo",
152 dameTipo(lamina_tipo_opciones),dameTipo (lamina_tipo_mensaje));
153
154 }
155 else if(lamina_tipo.dameSeleccion().equals("Opción")){
156 JOptionPane.showOptionDialog(Marco_Dialogos.this, dameMensaje(), "Titulo",
157 1,dameTipo (lamina_tipo_mensaje),null,dameOpciones(lamina_opciones),null);
158 }
159 else if(lamina_tipo.dameSeleccion().equals("Entrada")){
160 if(lamina_entrada.dameSeleccion().equals("Campo de texto")) {
161 JOptionPane.showInputDialog(Marco_Dialogos.this,dameMensaje(),"Titulo",dameTipo(lamina_tipo_mensaje));
162 }else {
163 JOptionPane.showInputDialog(Marco_Dialogos.this, dameMensaje(), "Titulo",
164 dameTipo(lamina_tipo_mensaje), null, new String[] {"Amario", "Azul", "Rojo"},"Azul");
165 }
166 }
167 }
168 }

```

Modificamos el siguiente código que ampliamos a continuación.

```

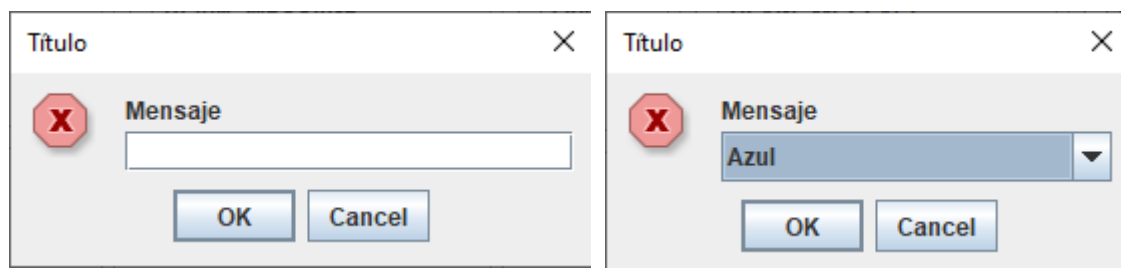
else if(lamina_tipo.dameSeleccion().equals("Entrada")){
    if(lamina_entrada.dameSeleccion().equals("Campo de
texto")) {

        JOptionPane.showInputDialog(Marco_Dialogos.this,dameMensaje(),"Titulo"
,dameTipo(lamina_tipo_mensaje));
    }else {

        JOptionPane.showInputDialog(Marco_Dialogos.this, dameMensaje(),
"Titulo",
                                dameTipo(lamina_tipo_mensaje),
null, new String[] {"Amario", "Azul", "Rojo"},"Azul");
    }
}

```

Si en la lámina Tipo seleccionamos Entrada y en la lámina Entrada seleccionamos "Campo de texto" y luego "Combo" observaremos las siguientes ventanas.



Ahora como resumen adjunto el código completo de los tres ficheros para poder repasar el código si tienes alguna duda.

Fichero Aplicacion\_Dialogos:

---

```
package practicas;

import javax.swing.JFrame;

public class Aplicacion_Dialogos {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Marco_Dialogos mimarco=new Marco_Dialogos();
        mimarco.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        mimarco.setVisible(true);
    }
}
```

---

Fichero Lamina\_Botones:

---

```
package practicas;

import javax.swing.JFrame;

public class Aplicacion_Dialogos {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Marco_Dialogos mimarco=new Marco_Dialogos();
        mimarco.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        mimarco.setVisible(true);
    }
}
```

---

Fichero Marco\_Dialogos:

---

```
package practicas;
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.geom.Rectangle2D;
import java.util.Date;
public class Marco_Dialogos extends JFrame{
    public Marco_Dialogos() {
        setTitle("Prueba de diálogo");
        setBounds(500,300,600,450);
        JPanel lamina_cuadrícula=new JPanel();
        lamina_cuadrícula.setLayout(new GridLayout(2,3));
        //-----
        String primero[]= {"Mensaje", "Confirmar", "Opción", "Entrada"};
        lamina_tipo = new Lamina_Botones("Tipo",primero);
        lamina_cuadrícula.add(lamina_tipo);
        //-----
        lamina_tipo_mensaje=new Lamina_Botones("Tipo de Mensajes",new
String[] {
                "ERROR_MESSAGE",
```

```

        "INFORMATION_MESSAGE",
        "WARNING_MESSAGE",
        "QUESTION_MESSAGE",
        "PLAIN_MESSAGE"
    });
    lamina_cuadrícula.add(lamina_tipo_mensaje);
    //-----
    lamina_mensaje=new Lamina_Botones("Mensajes",new String[] {
        "Cadena",
        "Icono",
        "Componentes",
        "Otros",
        "Objects[]"
    });
    lamina_cuadrícula.add(lamina_mensaje);
    //-----
    String[] {
        lamina_tipo_opciones=new Lamina_Botones("Confirmar",new
            "DEFAULT_OPTION",
            "YES_NO_OPTION",
            "YES_NO_CANCEL_OPTION",
            "OK_CANCEL_OPTION"
        });
        lamina_cuadrícula.add(lamina_tipo_opciones);
    //-----
        lamina_opciones=new Lamina_Botones("Opción",new String[] {
            "String[]",
            "Icon[]",
            "Object[]"
        });
        lamina_cuadrícula.add(lamina_opciones);
    //-----
    String[] {
        lamina_entrada=new Lamina_Botones("Entrada",new
            "Campo de texto",
            "Combo"
        });
        lamina_cuadrícula.add(lamina_entrada);

    setLayout(new BorderLayout());
    //-----

    JPanel lamina_mostrar=new JPanel();
    JButton boton_mostrar=new JButton ("Mostrar");
    boton_mostrar.addActionListener(new AccionMostrar());
    lamina_mostrar.add(boton_mostrar);

    add(lamina_mostrar, BorderLayout.SOUTH);
    add(lamina_cuadrícula, BorderLayout.CENTER);
    //-----
}
public Object dameMensaje() {
    String s=lamina_mensaje.dameSeleccion();
    if(s.equals("Cadena")) {
        return cadenaMensaje;
    }else if(s.equals("Icono")){
        return iconoMensaje;
    }else if(s.equals("Componentes")) {
        return ComponenteMensaje;
    }
}

```

```

    }else if(s.equals("Otros")) {
        return objetoMensaje;
    }else if(s.equals("Objects[]")) {
        return new Object[] {
            cadenaMensaje,
            iconoMensaje,
            ComponenteMensaje,
            objetoMensaje
        };
    }else {
        return null;
    }
}

public int dameTipo(Lamina_Botones lamina){
    String s=lamina.dameSeleccion();
    if(s.equals("ERROR_MESSAGE")||s.equals("YES_NO_OPTION")) {
        return 0;
    }
    else
    if(s.equals("INFORMATION_MESSAGE")||s.equals("YES_NO_CANCEL_OPTION")){
        return 1;
    }
    else
    if(s.equals("WARNING_MESSAGE")||s.equals("OK_CANCEL_OPTION")) {
        return 2;
    }
    else if(s.equals("QUESTION_MESSAGE")) {
        return 3;
    }
    else if(s.equals("PLAIN_MESSAGE") || s.equals("DEFAULT_OPTION"))
    {
        return -1;
    }
    else {
        return 0;
    }
}

public Object[] dameOpciones(Lamina_Botones lamina) {
    String s=lamina.dameSeleccion();
    if(s.equals("String[]")) {
        return new String[] {"Amarillo","Azul","Rojo"};
    }else if(s.equals("Icon[]")) {
        return new Object[] {new
ImageIcon("src/graficos/bolaAzul.png"),
        new
ImageIcon("src/graficos/bolaAmarilla.png"),
        new ImageIcon("src/graficos/bolaRoja.png")};
    }else if(s.equals("Object[]")){
        return new Object[] {cadenaMensaje,
            iconoMensaje,
            ComponenteMensaje,
            objetoMensaje
        };
    }
    else {
        return null;
    }
}

```

```

    }
}

private class AccionMostrar implements ActionListener{

    @Override
    public void actionPerformed(ActionEvent e) {
        // TODO Auto-generated method stub
        System.out.print(lamina_tipo.dameSeleccion()+" ");
        System.out.print(lamina_tipo_mensaje.dameSeleccion()+"
");

        System.out.print(lamina_mensaje.dameSeleccion()+" ");
        System.out.print(lamina_tipo_opciones.dameSeleccion()+"
");

        System.out.print(lamina_opciones.dameSeleccion()+" ");
        System.out.println(lamina_entrada.dameSeleccion()+" ");

        if(lamina_tipo.dameSeleccion().equals("Mensaje")) {
            JOptionPane.showMessageDialog(Marco_Dialogos.this,dameMensaje(),
"Título",dameTipo (lamina_tipo_mensaje));
        }
        else if(lamina_tipo.dameSeleccion().equals("Confirmar")){
            JOptionPane.showConfirmDialog(Marco_Dialogos.this,
dameMensaje(), "Título",
dameTipo(lamina_tipo_opciones),dameTipo (lamina_tipo_mensaje));
        }
        else if(lamina_tipo.dameSeleccion().equals("Opción")){
            JOptionPane.showOptionDialog(Marco_Dialogos.this,
dameMensaje(), "Título",
1,dameTipo
(lamina_tipo_mensaje),null,dameOpciones(lamina_opciones),null);
        }
        else if(lamina_tipo.dameSeleccion().equals("Entrada")){
            if(lamina_entrada.dameSeleccion().equals("Campo de
texto")) {
                JOptionPane.showInputDialog(Marco_Dialogos.this,dameMensaje(),"Título"
,dameTipo(lamina_tipo_mensaje));
            }else {
                JOptionPane.showInputDialog(Marco_Dialogos.this, dameMensaje(),
"Título",
dameTipo(lamina_tipo_mensaje),
null, new String[] {"Amarillo", "Azul", "Rojo"},"Azul");
            }
        }
    }

    private Lamina_Botones lamina_tipo, lamina_tipo_mensaje,
lamina_mensaje;
    private Lamina_Botones lamina_tipo_opciones, lamina_opciones,
lamina_entrada;
    private String cadenaMensaje="Mensaje";
    private Icon iconoMensaje=new ImageIcon("src/graficos/bolaAzul.png");
}

```

```
private Object objetoMensaje=new Date();
private Component ComponenteMensaje=new Lamina_Ejemplo();
}

class Lamina_Ejemplo extends JPanel{
public void paintComponent(Graphics g) {
super.paintComponents(g);
Graphics2D g2=(Graphics2D) g;
Rectangle2D rectangulo = new Rectangle2D.Double(0,0,getWidth(),
getHeight());
g2.setPaint(Color.YELLOW);
g2.fill(rectangulo);
}
}
```

---

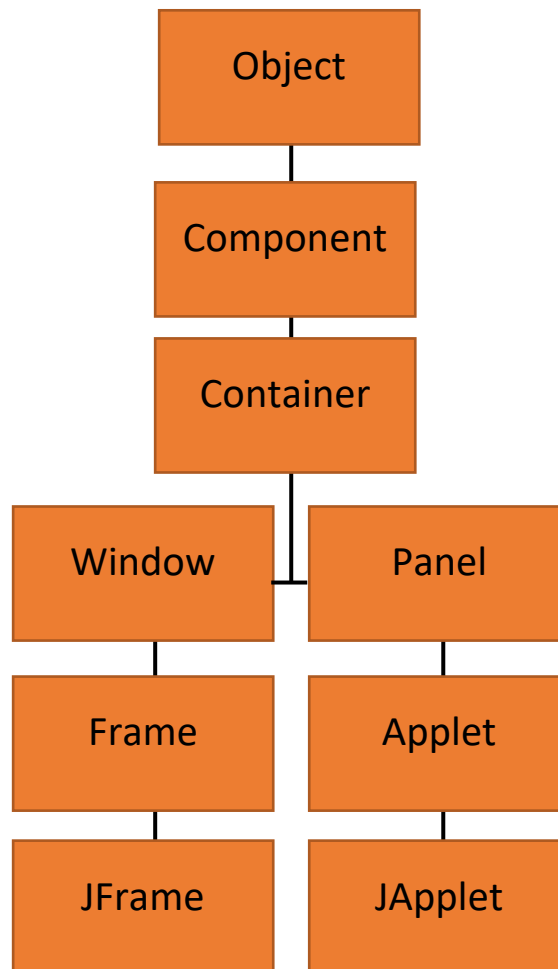


## Despliegue Aplicaciones. Applets I. (VÍdeo 132)

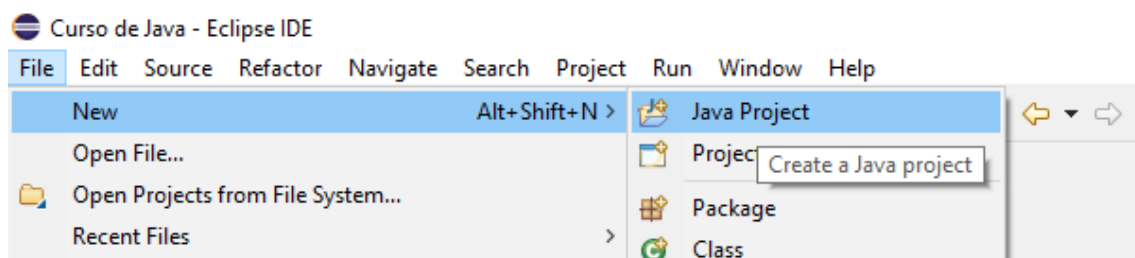
Despliegue de aplicaciones



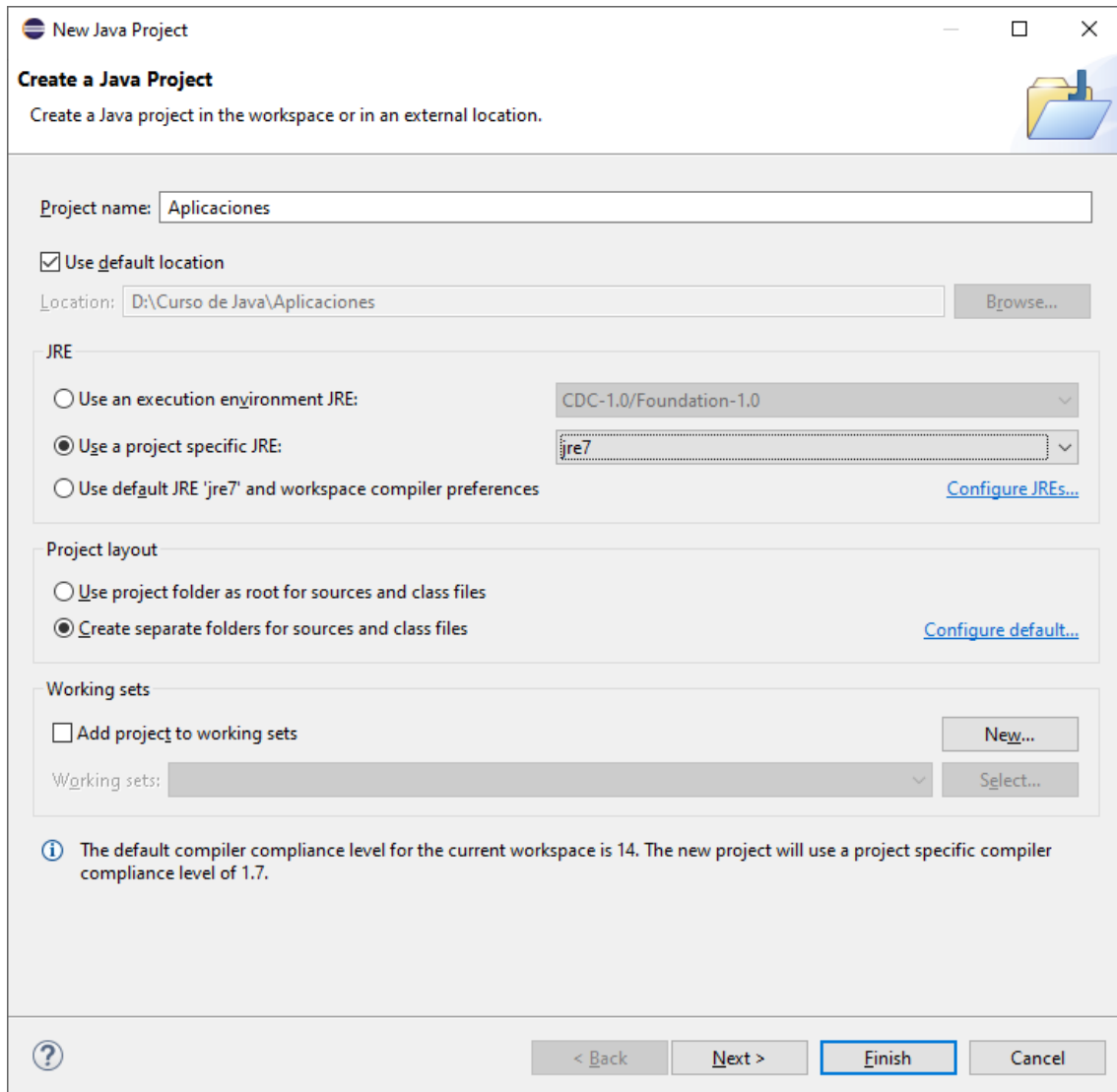
JERARQUÍA DE HERENCIA DE UN APPLLET



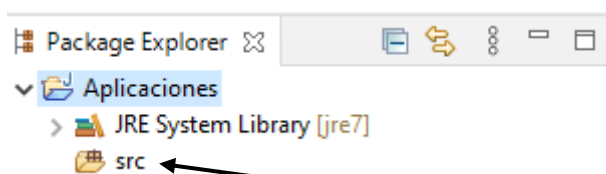
Ejecutamos Eclipse y vamos a realizar un proyecto nuevo para ello del menú File seleccionamos New y de este Java Project.



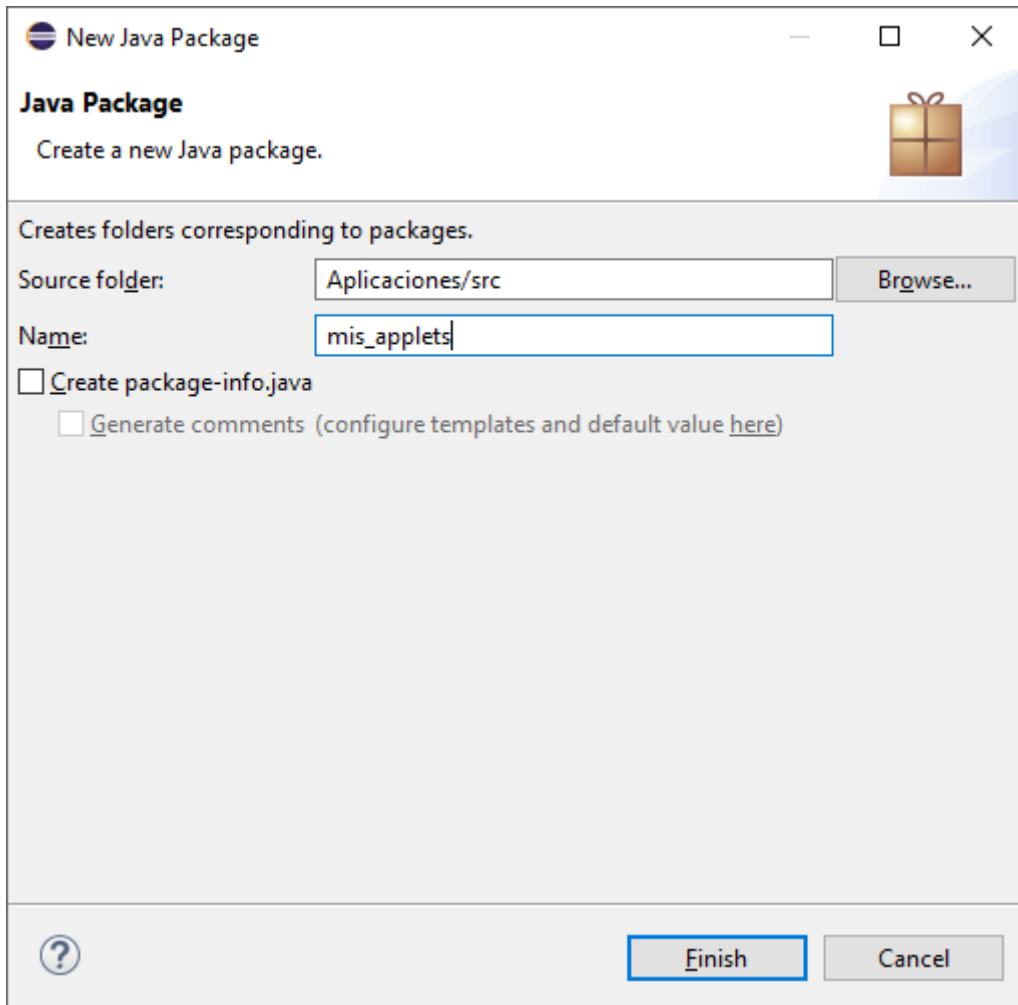




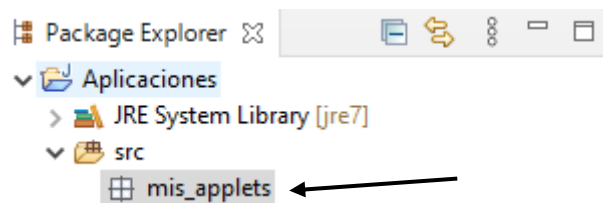
Le damos al botón Finish.



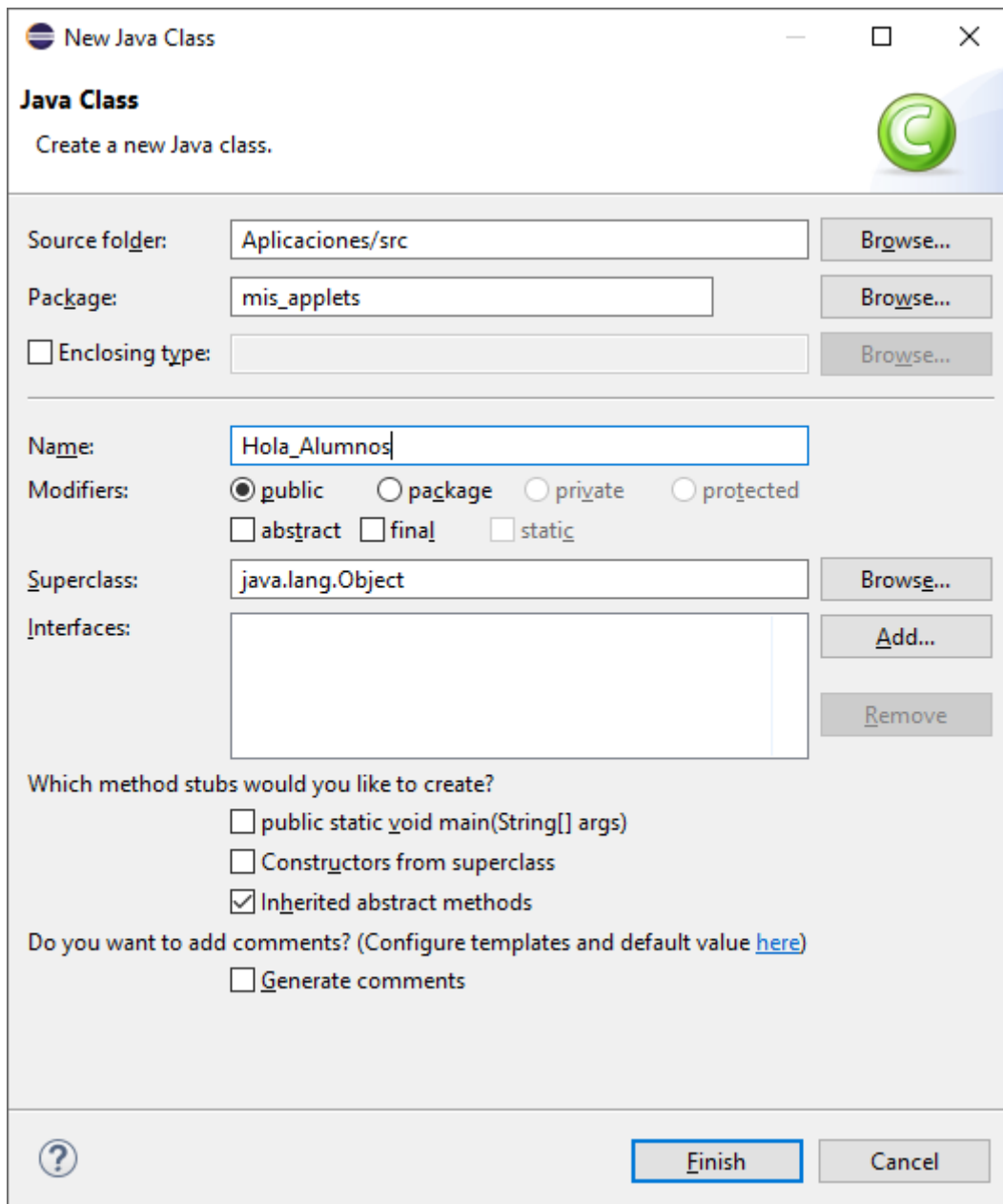
Seleccionamos la carpeta src y vamos a crear un paquete nuevo llamado mis\_applets.



Seguido del botón Finish.



Seleccionamos mis\_applets y vamos a crear una nueva clase.



No seleccionaremos public static void main(String[] args).

```
Hola_Alumnos.java ✕
▶ Aplicaciones ▶ src ▶ mis_applets ▶ Hola_Alumnos
1 package mis_applets;
2
3 public class Hola_Alumnos {
4
5 }
6
```

```

*Hola_Alumnos.java
Aplicaciones > src > mis_applets > Hola_Alumnos
1 package mis_applets;
2 import javax.swing.*;
3
4 public class Hola_Alumnos extends JApplet {
5
6 }
7

```

Todas las clases que construyen un applet deben de heredar JApplet y además debemos importar el paquete javax.swing.\*;

```

*Hola_Alumnos.java
Aplicaciones > src > mis_applets > Hola_Alumnos > init() : void
1 package mis_applets;
2 import javax.swing.*;
3
4 public class Hola_Alumnos extends JApplet {
5     public void init() {
6
7     }
8 }

```

La clase principal tiene que contener public void init().

Estamos sobrescribiendo el método init.

```

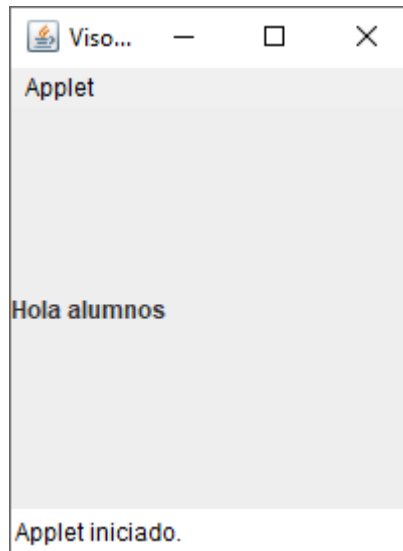
*Hola_Alumnos.java
Aplicaciones > src > mis_applets > Hola_Alumnos > init() : void
1 package mis_applets;
2 import javax.swing.*;
3
4 public class Hola_Alumnos extends JApplet {
5     public void init() {
6         JLabel rotulo=new JLabel("Hola alumnos");
7         add(rotulo);
8     }
9 }

```

Como no queremos imprimir en la consola de Eclipse utilizaremos los componentes swing como JLabel.

Esta aplicación se tendrá que ver en un navegador, pero para ello también tenemos un visor de applets llamado appletviewer.

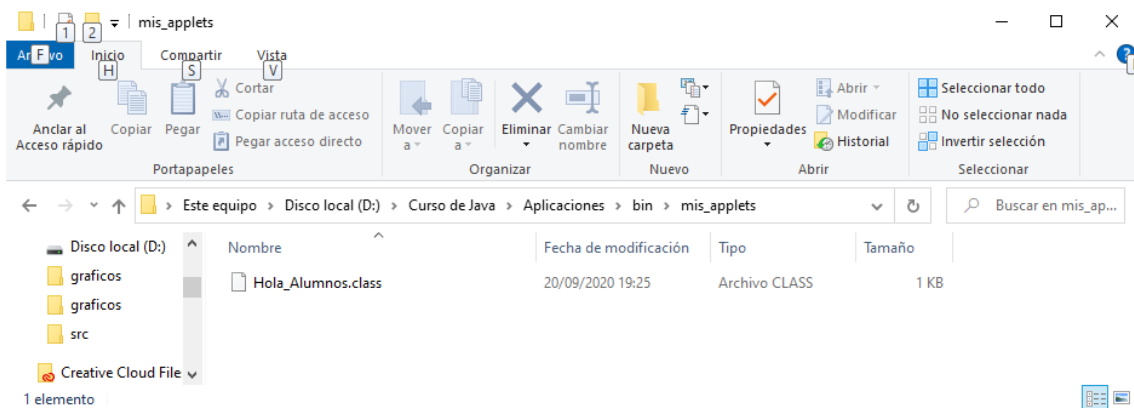
Vamos a ejecutar como siempre pero Eclipse tiene que ejecutarlos en dicho visor.



Pero el objetivo es verlo en el navegador.

Necesitamos tener pequeños conocimientos de html.

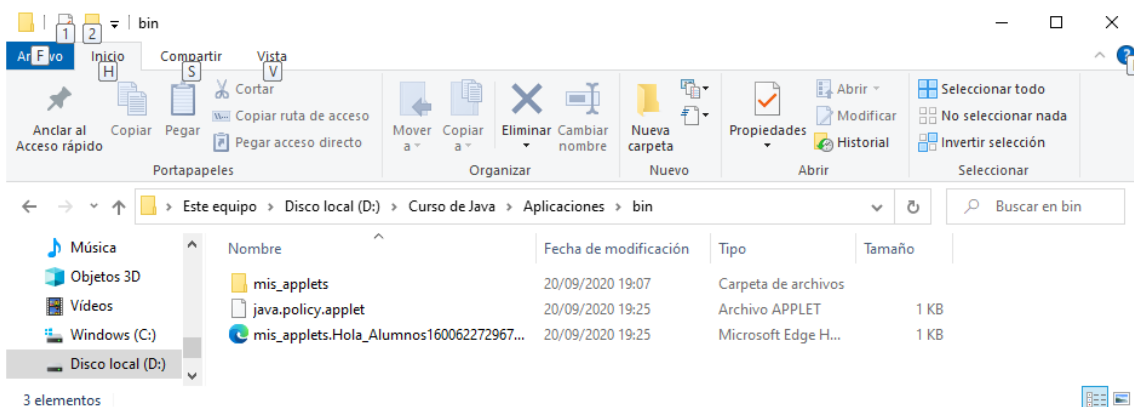
Ahora hemos de buscar donde se encuentra el archivo Hola\_Alumnos.class.



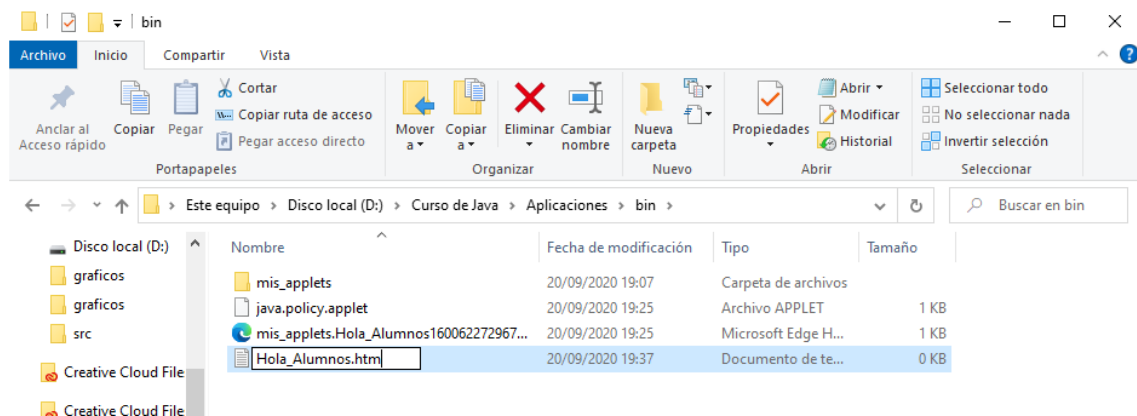
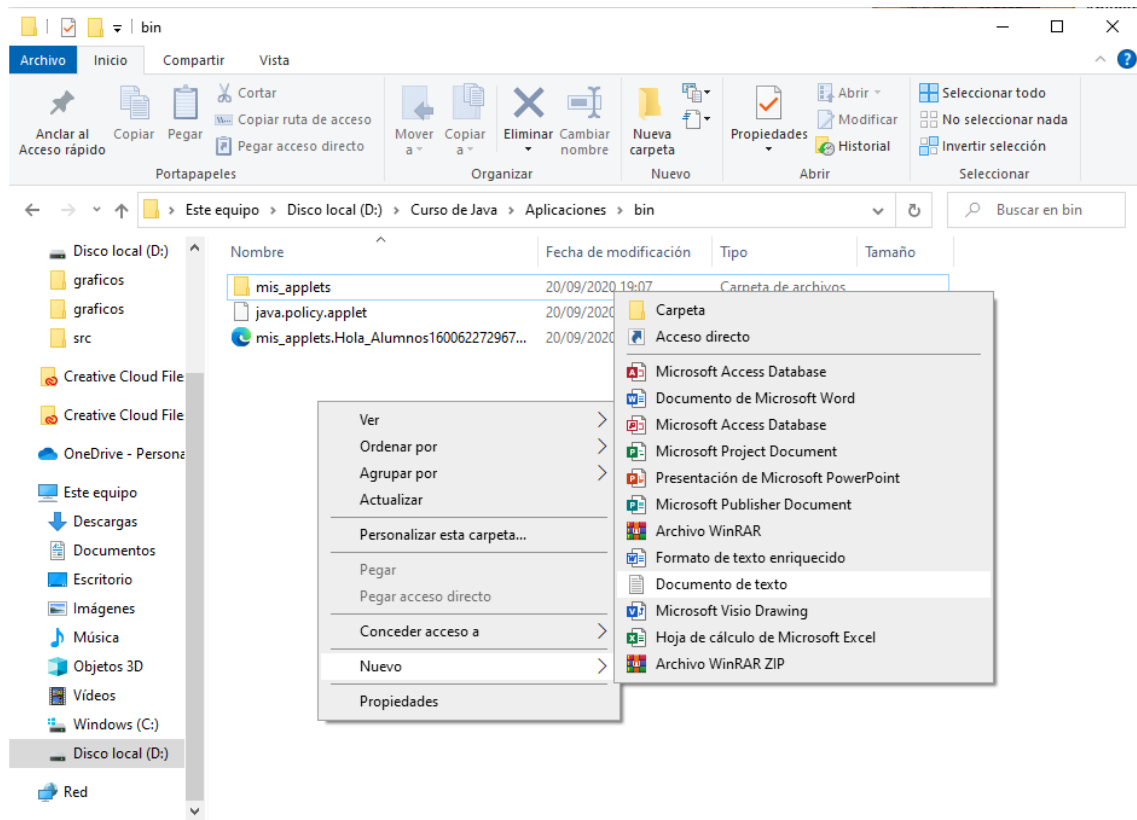
D:\Curso de Java\Aplicaciones\bin\mis\_applets

Cuando ejecutamos la primera vez Eclipse este nos pidió donde queríamos guardar nuestros proyectos, en mi caso en el disco “D” en la carpeta “Curso de Java”.

Salimos de esta carpeta para irnos a la carpeta bin.



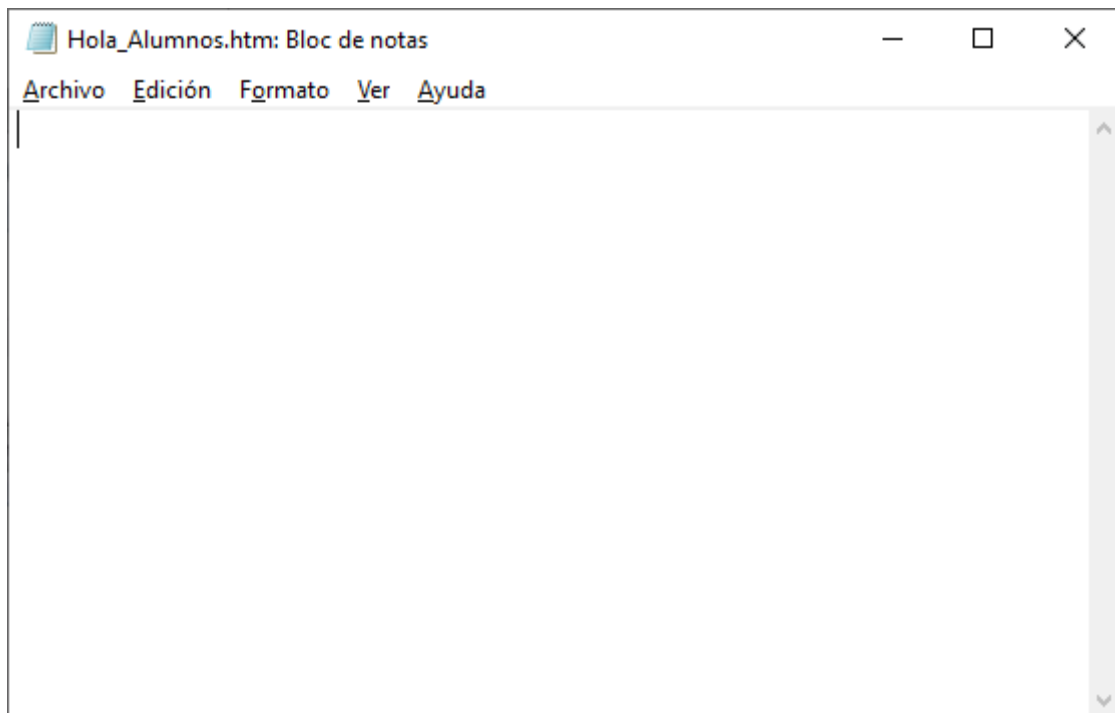
Seleccionamos el botón derecho en esta carpeta y del menú seleccionaremos Nuevo y de este Documento de texto.



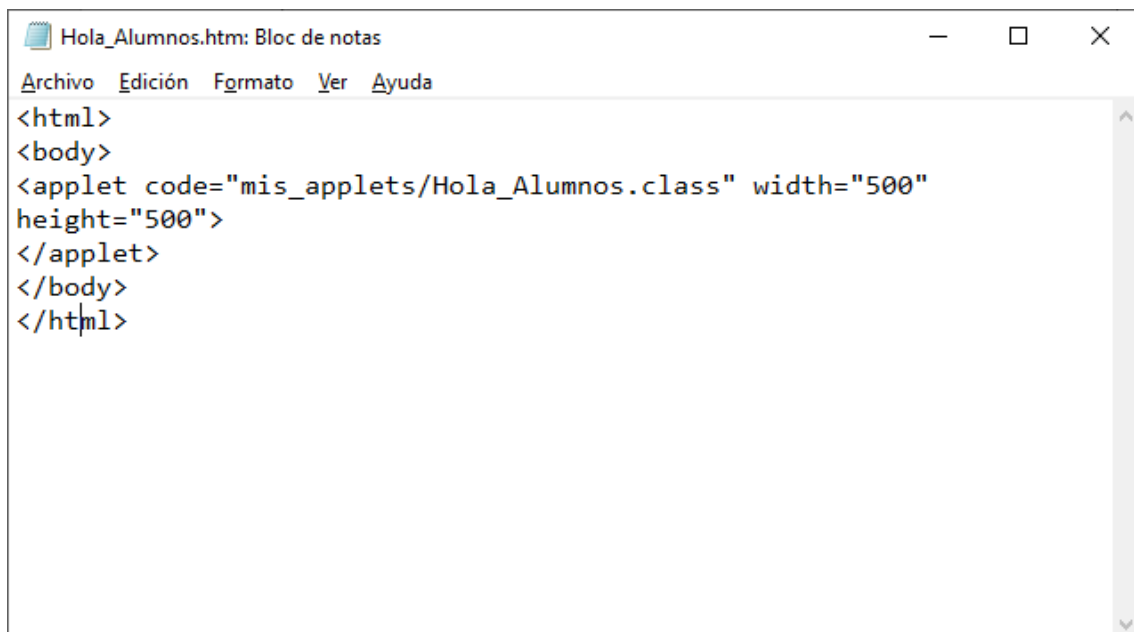
Nombre	Fecha de modificación	Tipo	Tamaño
mis_applets	20/09/2020 19:07	Carpeta de archivos	
Hola_Alumnos.htm	20/09/2020 19:37	Microsoft Edge H...	0 KB
java.policy.applet	20/09/2020 19:25	Archivo APPLET	1 KB
mis_applets.Hola_Alumnos160062272967...	20/09/2020 19:25	Microsoft Edge H...	1 KB

Ya lo reconoce como una página web.

La tenemos que abrir con el bloc de notas, botón derecho, abrir y con Bloc de notas.

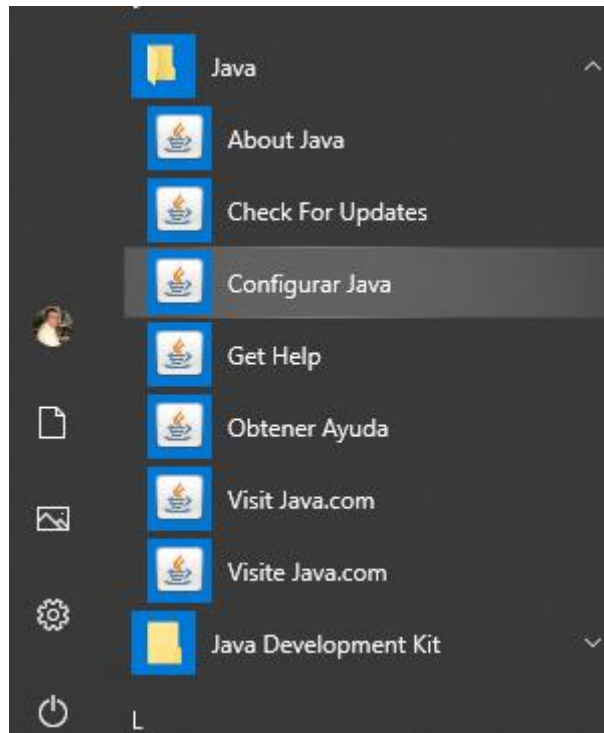


Escribimos el siguiente código:

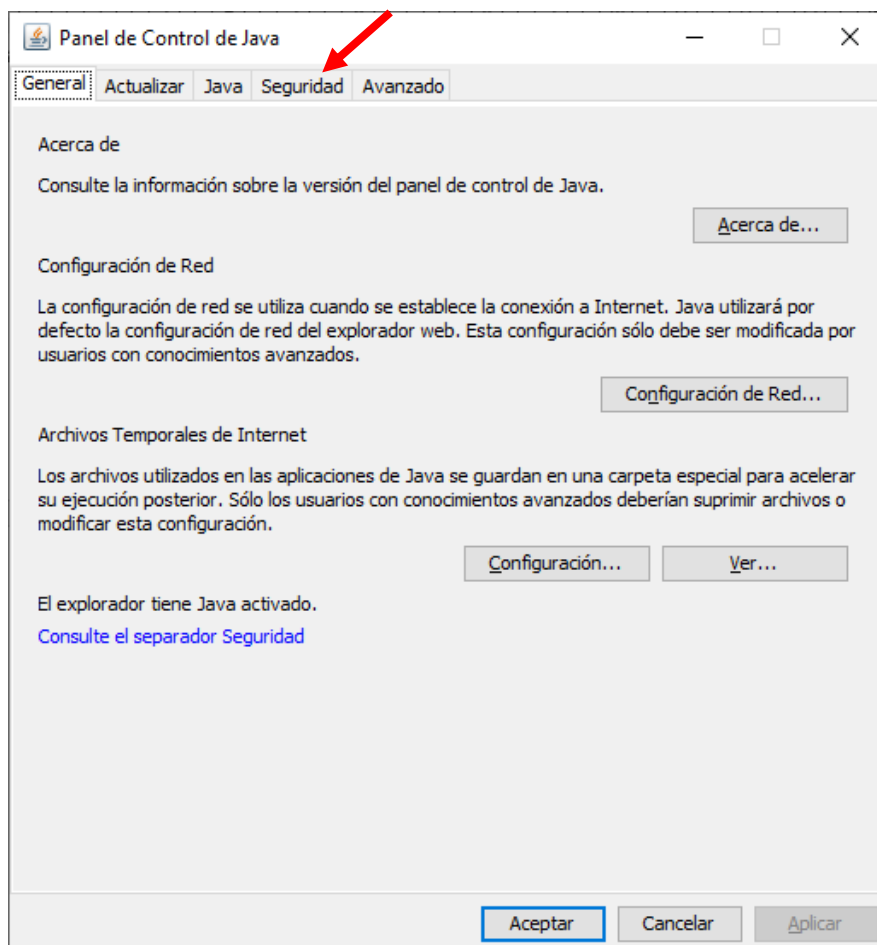


Guardamos cambios y cerramos la aplicación.

Para poder modificar la seguridad de java vamos a ejecutar la consola de java.

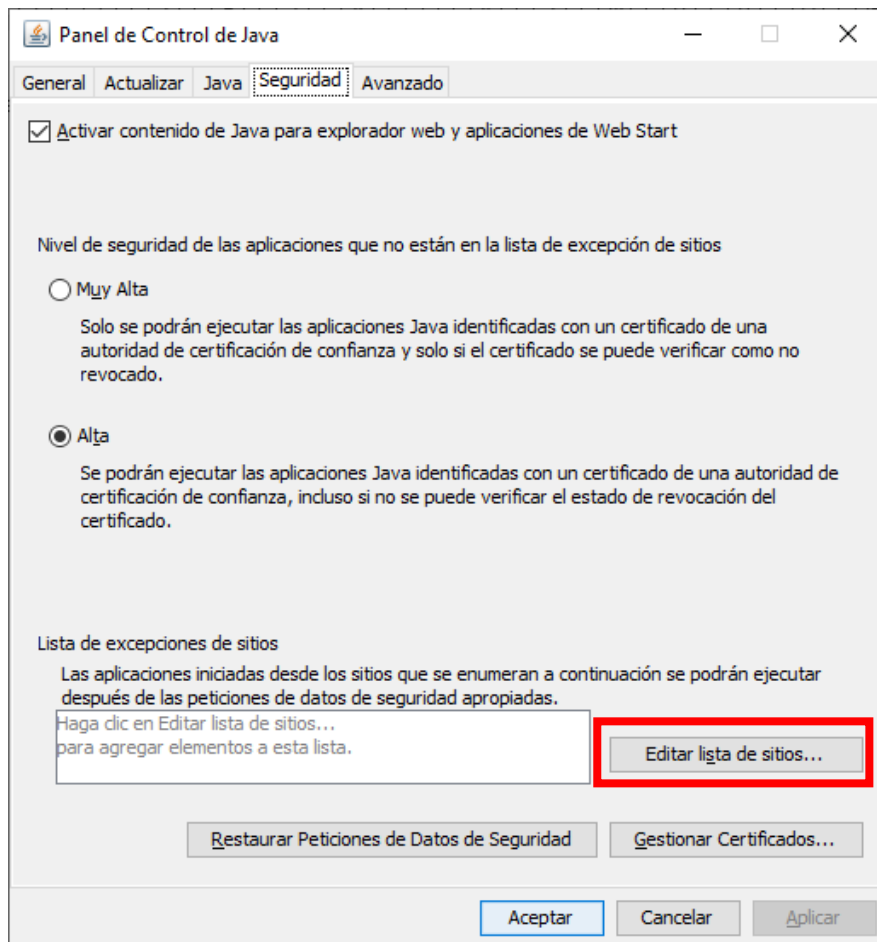


En el botón inicio buscaremos la carpeta Java y encontraremos Configurar Java.

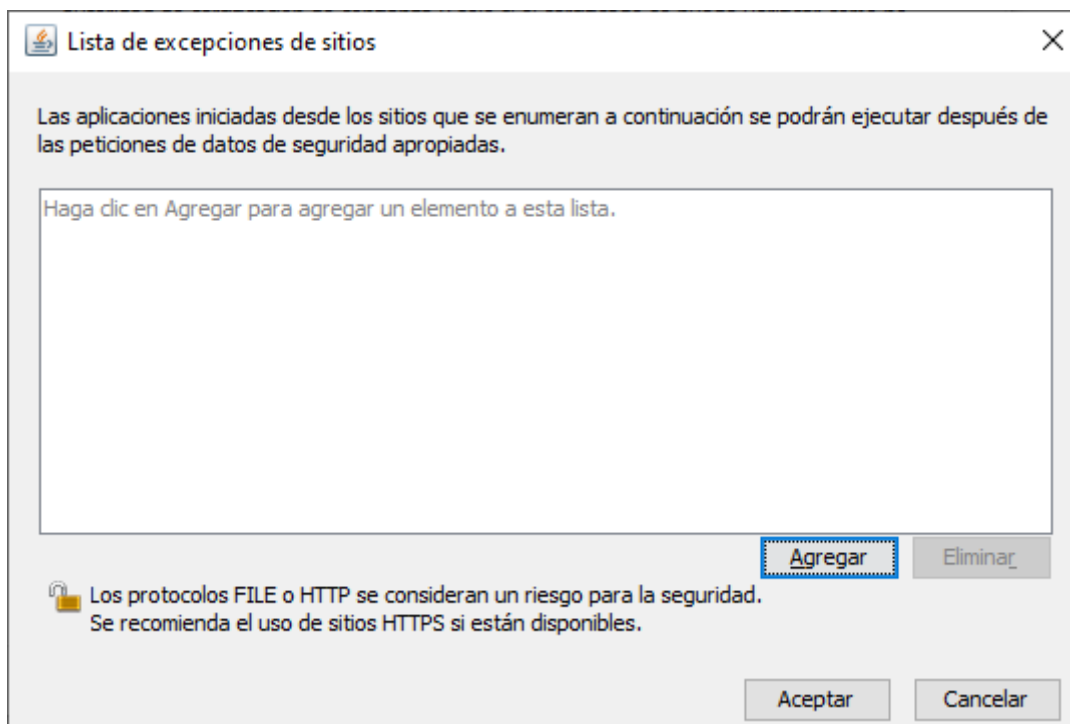


Nos vamos a la pestaña de Seguridad.

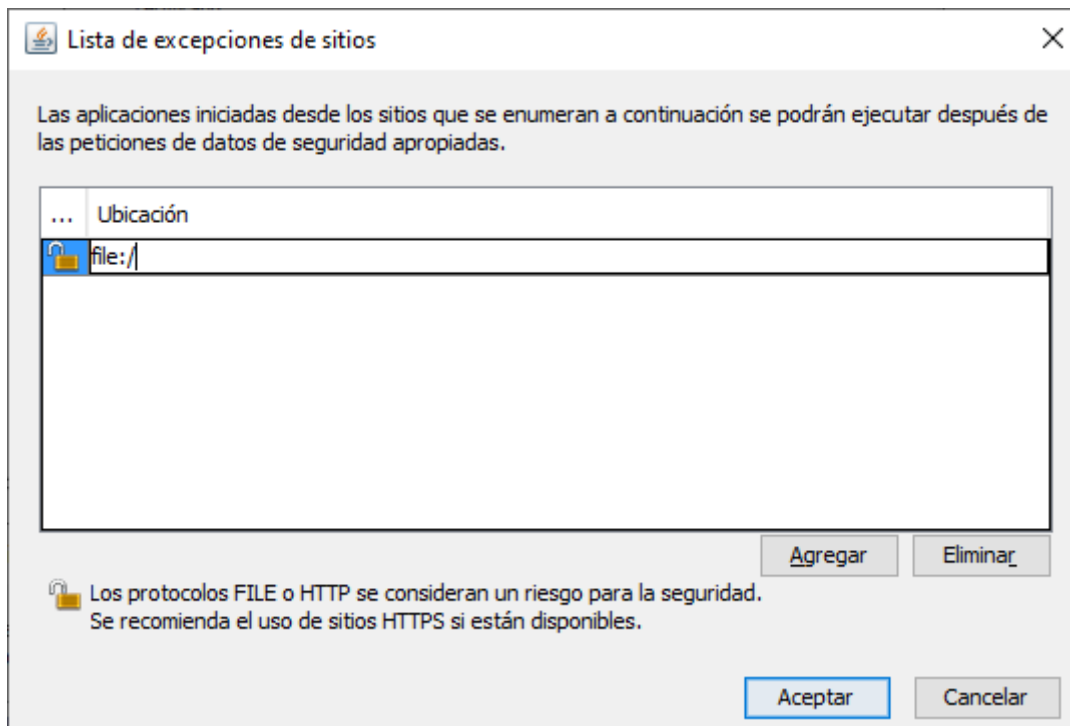




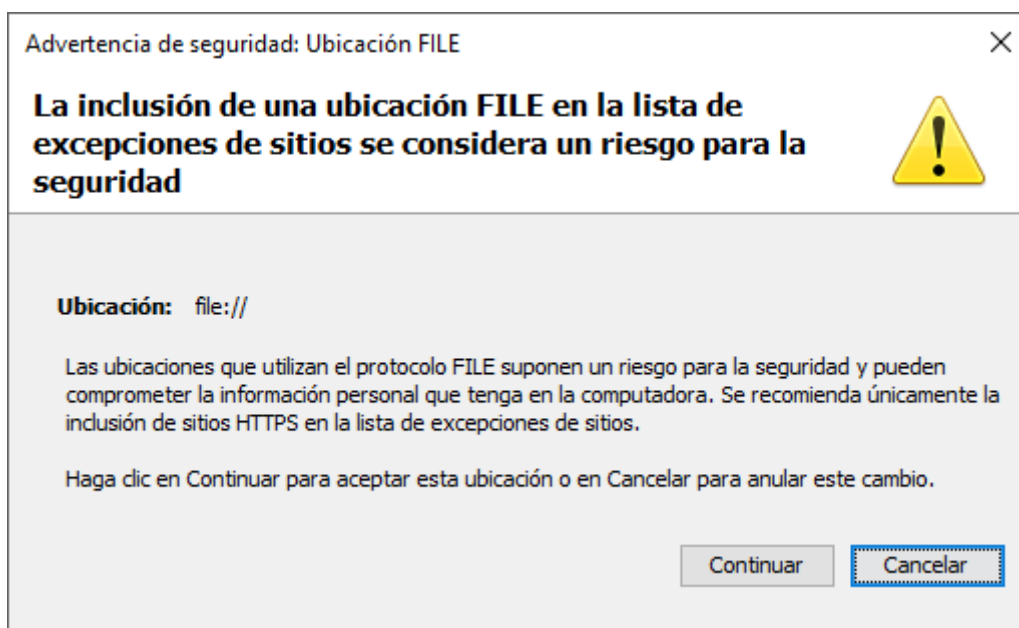
Seleccionaremos Editar lista de sitios...



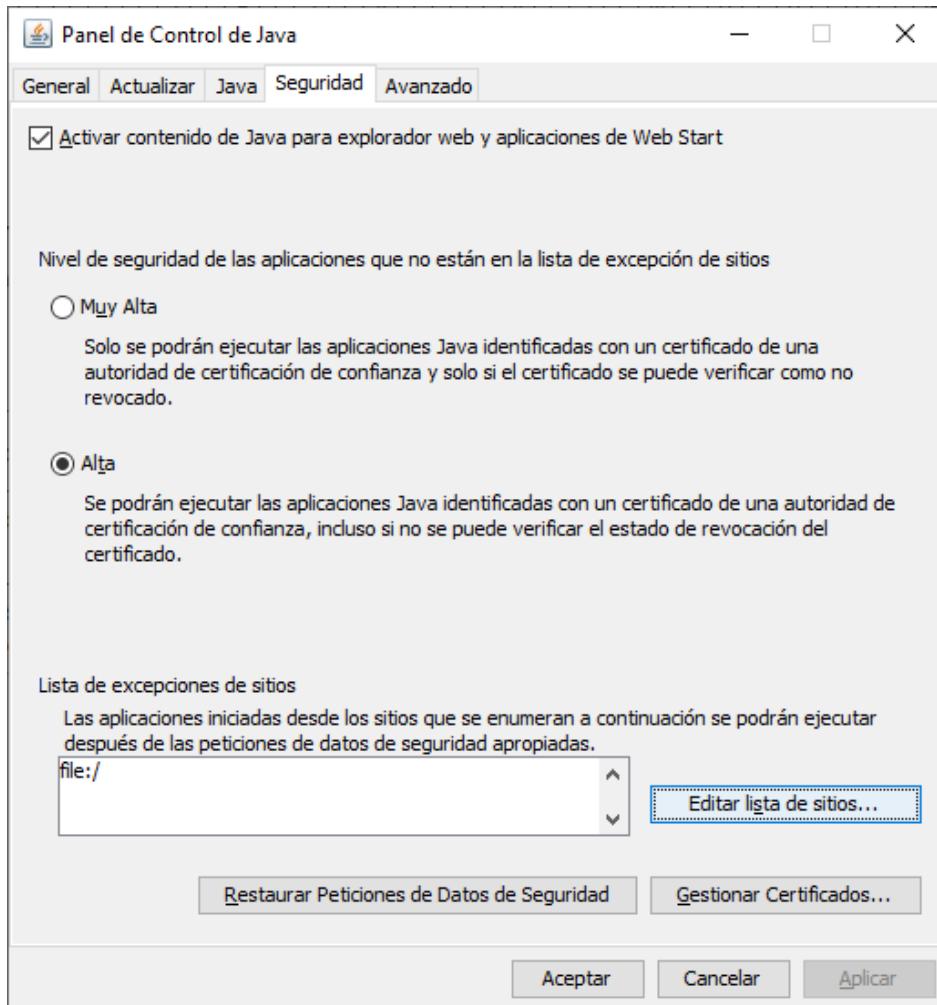
Seleccionar el botón Agregar



Le damos al botón Aceptar.

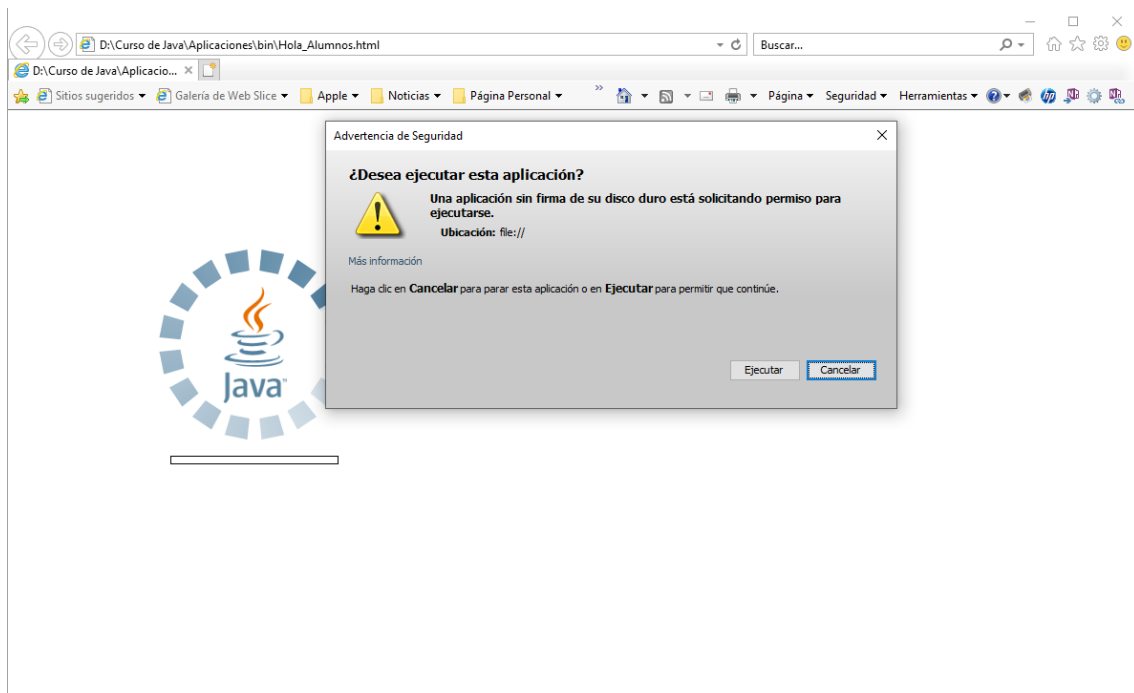


Le damos al botón Continuar.

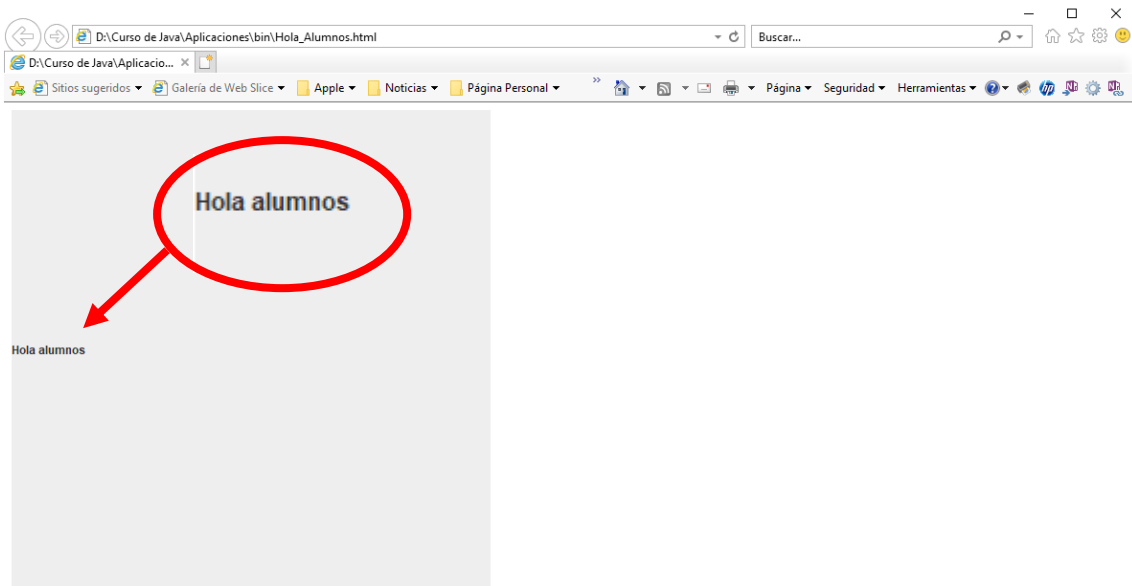


Le damos al botón Aceptar.

Vamos a hacer doble clic en el archivo Hola\_Alumnos.html.



Le damos a ejecutar.



El navegador que a mí me funciona bien es Internet Explorer.



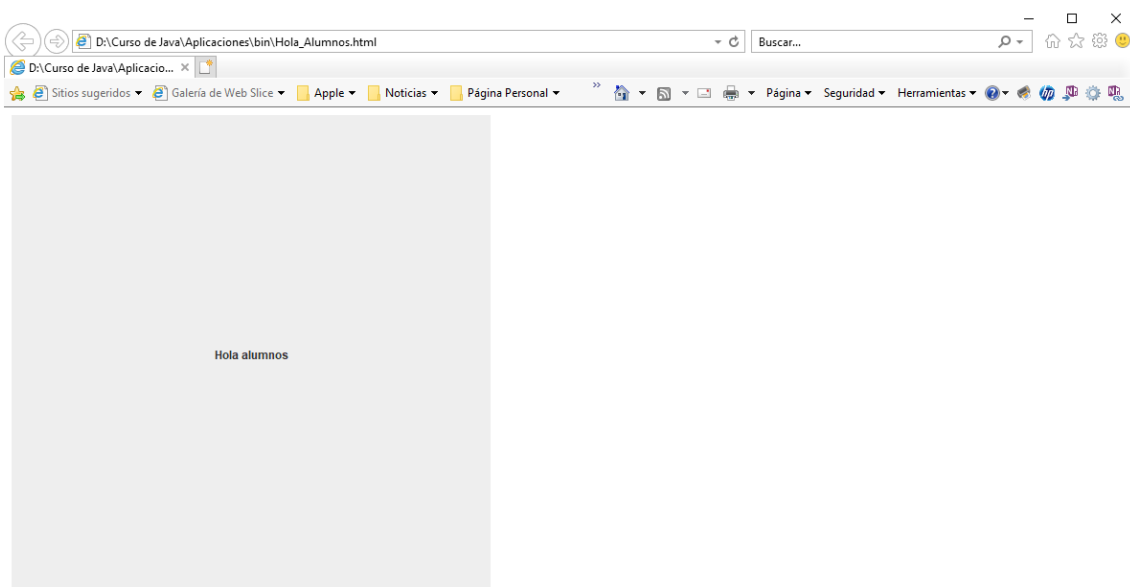
## Despliegue Aplicaciones. Applet II. (VÍdeo 133)

Convertir aplicaciones en Applets.

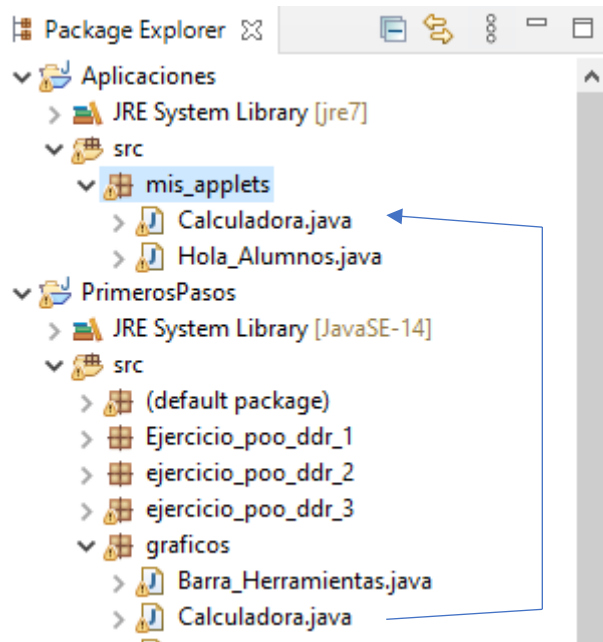
```
▶ Aplicaciones ▶ src ▶ mis_applets ▶ Hola_Alumnos ▶ init() : void
1 package mis_applets;
2 import javax.swing.*;
3
4 public class Hola_Alumnos extends JApplet {
5     public void init() {
6         JLabel rotulo=new JLabel("Hola alumnos", SwingConstants.CENTER);
7         add(rotulo);
8     }
9 }
```

Para que el mensaje aparezca centrado.

Ejecutamos Eclipse este hace que se sobrescriba el fichero class, ya lo podemos ver desde el navegador.



Ahora vamos a abrir el proyecto de la calculadora en Eclipse.



Hemos copiado la calculado en el paquete mis\_applets.

```

8 public class Calculadora extends JApplet{
9
10 public void init() {
11
12     LaminaCalculadora milamina=new LaminaCalculadora();
13     add(milamina);
14
15 }
16 }

```

La clase main se ha de cambiar por init.

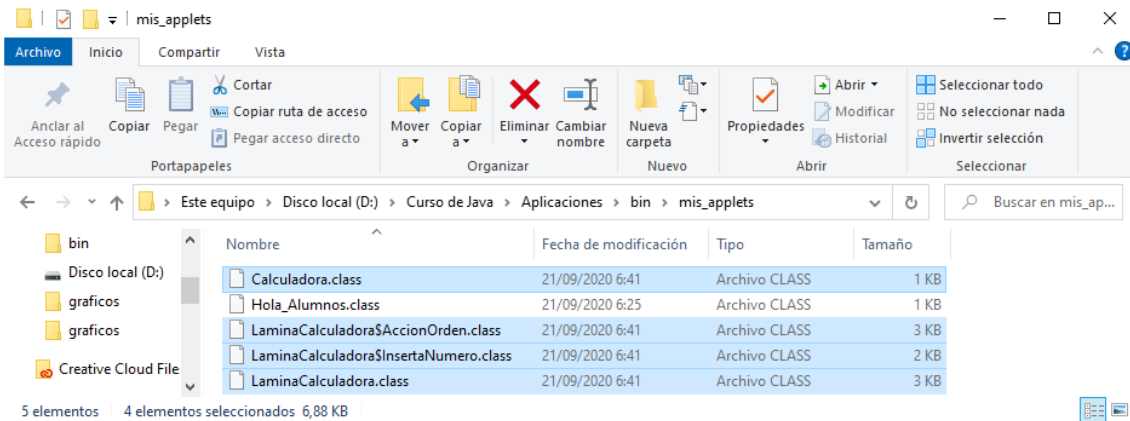
Parta de la clase Marco se pasa a init y posterior mente eliminamos la clase Marco.

Le damos al play de Eclipse.

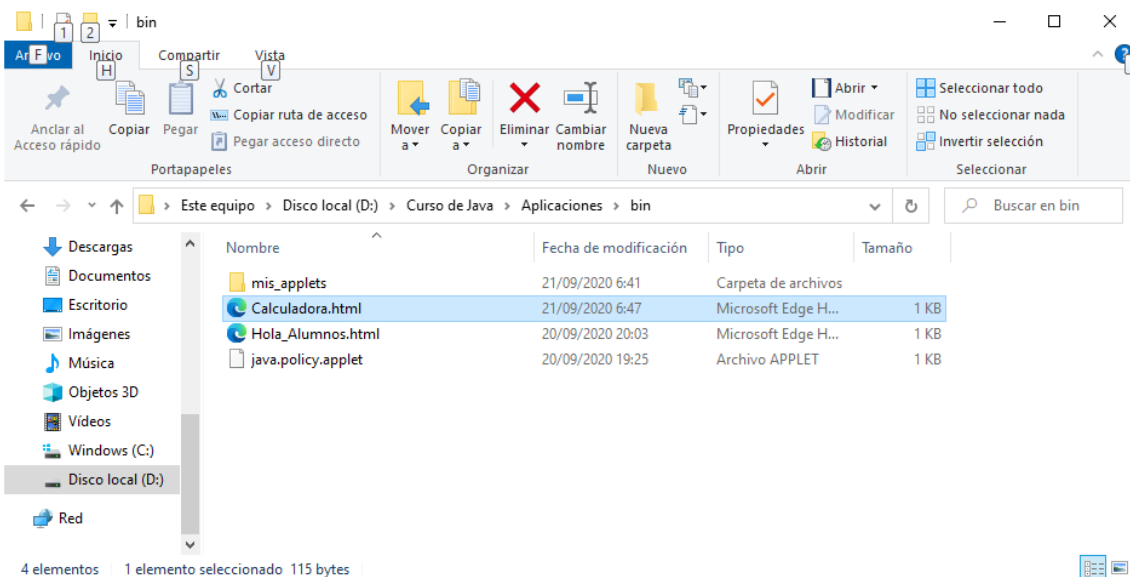


Vemos la aplicación en el visualizador de Applets.

Nos vamos a la carpeta donde está la calculadora.class



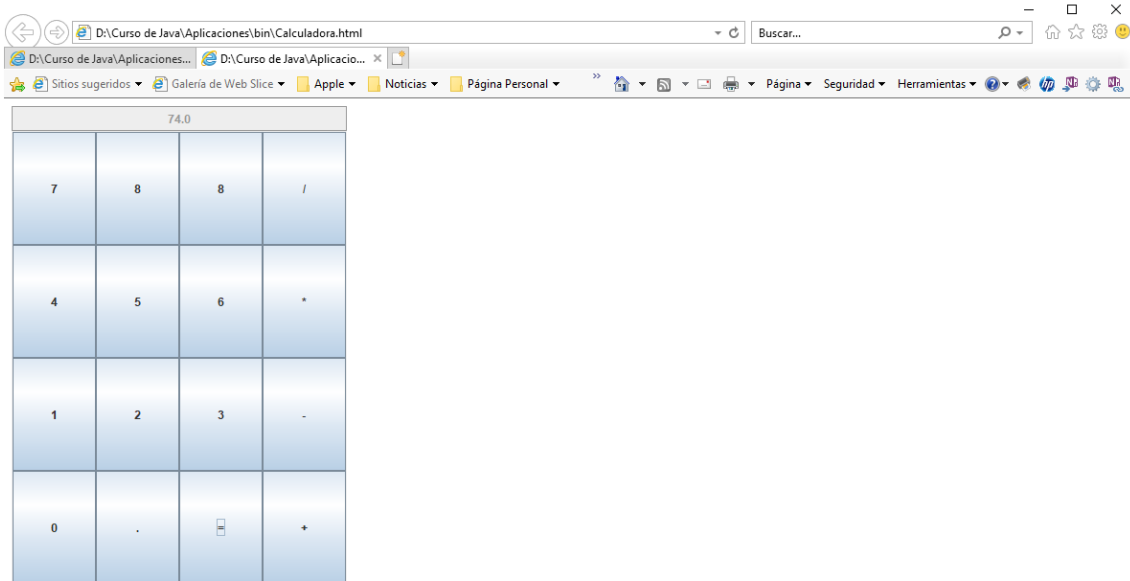
Nos vamos a la carpeta bin de Aplicaciones.



Creamos un archivo html con el siguiente contenido:

```
Calculadora.html: Bloc de notas
Archivo Edición Formato Ver Ayuda
<html>
<body>
<applet code="mis_applets/Calculadora.class" width="350" height="500">
</applet>
</body>
</html>
```

Ahora vamos a abrir el archivo creado con nuestro navegador.



La calculadora ya funciona en nuestro navegador.

Vamos a realizar algunas modificaciones en el archivo html.

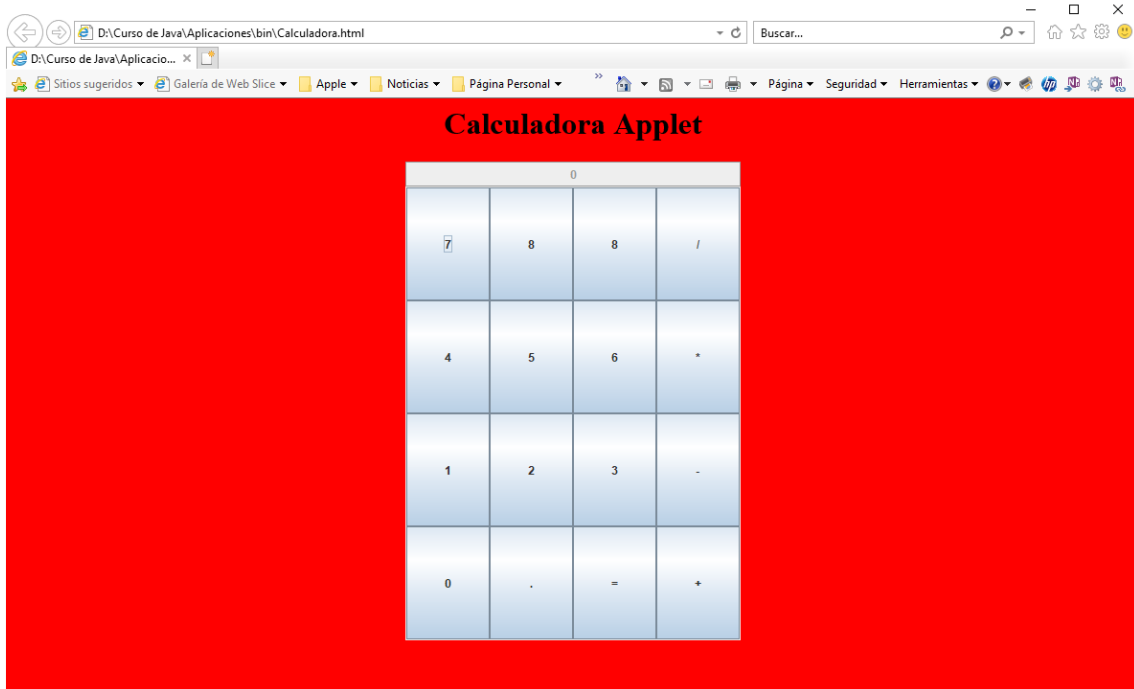
```
Calculadora.html: Bloc de notas
Archivo Edición Formato Ver Ayuda
<html>

<style>
#miapplet{
margin:auto;
position:relative;
width: 350px;
}
h1{
text-align:center;
}
</style>
<body bgcolor="red">
<h1>Calculadora Applet</h1>
<div id="miapplet">
<applet code="mis_applets/Calculadora.class" width="350" height="500" >
</applet>
</div>
</body>
</html>
```

Guardamos los cambios.

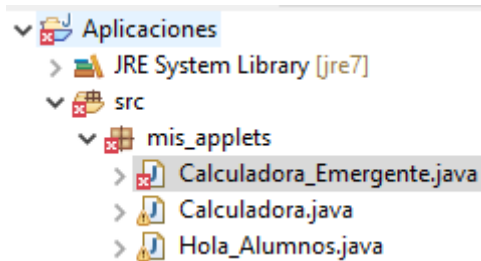
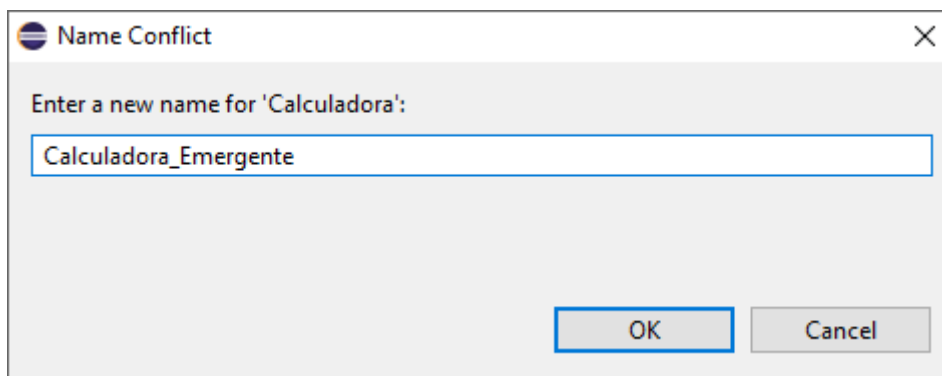
Lo abrimos desde el navegador:





## Despliegue Aplicaciones. Applets III. Ventanas emergentes. (Vídeo 134)

Vamos a copiar el archivo de la calculadora.



Vamos a modificar el código de la clase Calculadora\_Emergente.

```
8 public class Calculadora_Emergente extends JApplet{
9
10 public void init() {
11
12
13     final JFrame mimarco=new JFrame();
14     mimarco.setSize(500,300);
15     LaminaCalculadora milamina=new LaminaCalculadora();
16     mimarco.add(milamina);
17     JButton boton_Calculadora=new JButton("Calculadora");
18     boton_Calculadora.addActionListener(new ActionListener() {
19
20
21         @Override
22         public void actionPerformed(ActionEvent arg0) {
23             // TODO Auto-generated method stub
24             mimarco.setVisible(!mimarco.isVisible());
25         }
26     });
27     add(boton_Calculadora);
28
29 }
30 }
```

Clase interna

Ejecutamos desde Eclipse.

Modificamos el código del fichero html.

```
Calculadora_emergente.html: Bloc de notas
Archivo Edición Formato Ver Ayuda
<html>

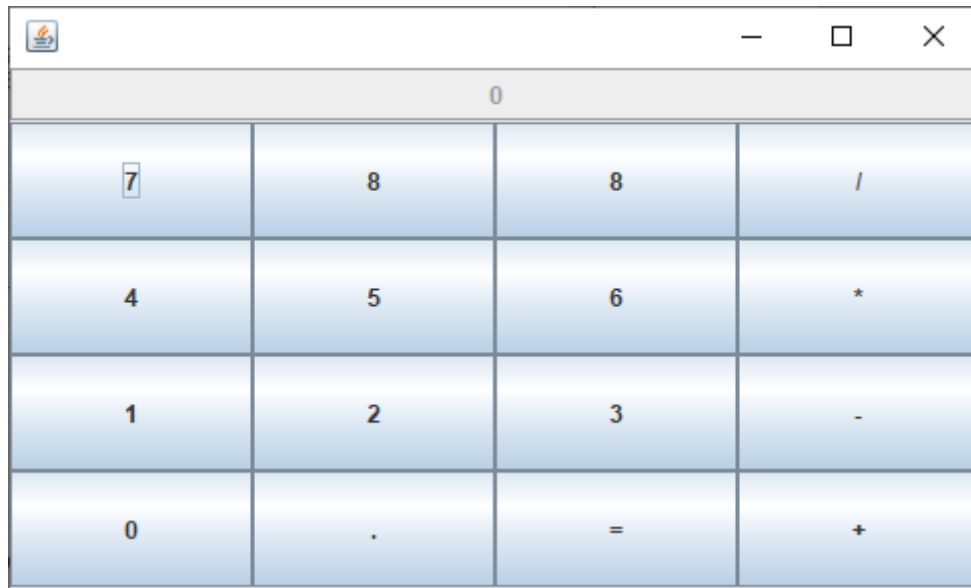
<style>
#miapplet{
margin:auto;
position:relative;
width: 350px;

}
h1{
text-align:center;
}
</style>
<body bgcolor="red">
<h1>Calculadora Applet</h1>
<div id="miapplet">Pulsa el bot&ocute;n para abrir la &nbsp;
<applet code="mis_applets/Calculadora_Emergente.class" width="125"
height="25" >
</applet>
</div>
</body>
</html>
```

Lo guardamos y lo abrimos con el navegador.



Hacemos clic al botón Calculadora.

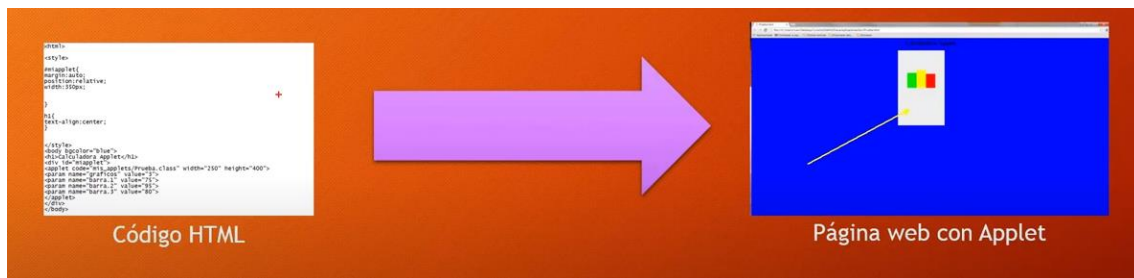


Si hacemos de nuevo clic en el botón Calculadora esta se cerrar.

En este ejemplo estamos ejecutando una aplicación fuera de nuestro navegador, con su propio marco.



## Despliegue Aplicaciones. Applets IV. Paso de parámetros. (Vídeo 135)



Supongamos que tenemos una página que tiene un gráfico y este gráfico se tiene que actualizar cada día.

El paso de parámetros nos ahorra bastante el trabajo.

Para ello tiene que haber una instrucción en el código html y otra instrucción en el Java.

Instrucción en html:

```
<param name="nombre" value="valor">
```

Por parte del programa java la siguiente instrucción.

```
getParameter("nombre");
```

Vamos a partir del siguiente ejemplo que llamaremos Grafico\_Applet que guardaremos en mis\_applets.

```
package mis_applets;
```

```
import javax.swing.*;
```

```
import java.awt.*;
```

```
import java.awt.geom.*;
```

```
public class Grafico_Applet {
```

```
    public static void main(String [] args){
```

```
        MarcoGrafico mimarco=new MarcoGrafico();
```

```
        mimarco.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

```
        mimarco.setVisible(true);
```

```
    }
```

```
}
```

```
class MarcoGrafico extends JFrame{
```

```
    public MarcoGrafico(){
```

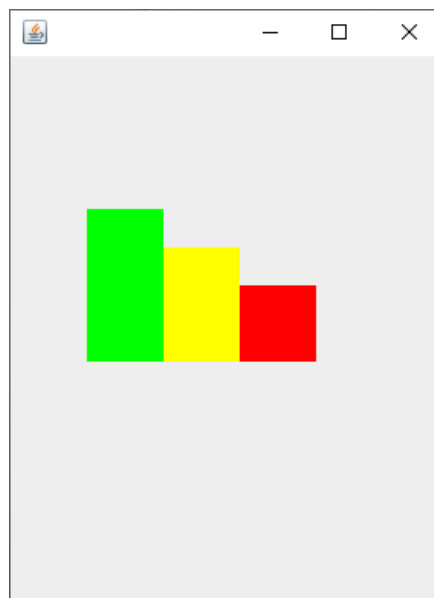
```
        setBounds(250,400,300,400);
```

```

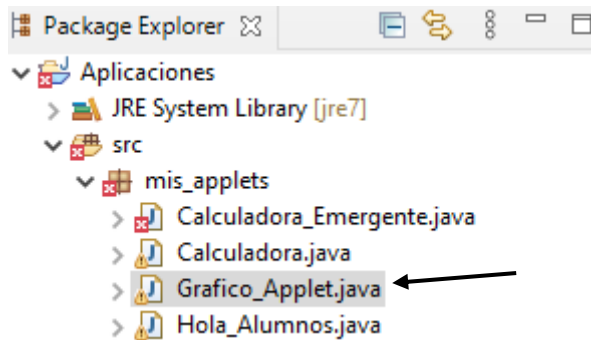
        LaminaGrafico milamina=new LaminaGrafico();
        add(milamina);
        setVisible(true);
    }
}
class LaminaGrafico extends JPanel{
    public void paint(Graphics g){
        super.paintComponent(g);
        Graphics2D g2=(Graphics2D) g;
        g2.rotate( 180/180.0 * Math.PI, 125, 100);
        Rectangle2D graf1=new Rectangle2D.Double(50,0,50,50);
        Rectangle2D graf2=new Rectangle2D.Double(100,0,50,75);
        Rectangle2D graf3=new Rectangle2D.Double(150,0,50,100);
        g2.setPaint(Color.RED);
        g2.fill(graf1);
        g2.setPaint(Color.YELLOW);
        g2.fill(graf2);
        g2.setPaint(Color.GREEN);
        g2.fill(graf3);
    }
}
}

```

Este será el resultado:



Donde tiene que estar guardado.



Ahora vamos a transformar a Applet.

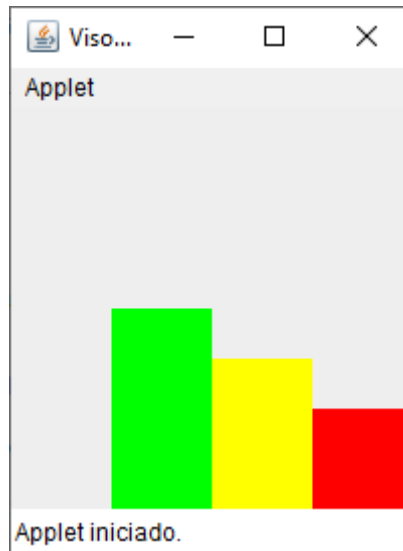
```
1 package mis_applets;
2
3 import javax.swing.*;
4
5 import java.awt.*;
6 import java.awt.geom.*;
7
8 public class Grafico_Applet extends JApplet { ←
9
10 public void init() {
11     LaminaGrafico milamina=new LaminaGrafico();
12     add(milamina);
13
14 }
15
16 }
17
18 }
```

Cambiamos la clase main por lo señalado en la flecha, este código es parte del código

```
class MarcoGrafico extends JFrame{
    public MarcoGrafico(){
        setBounds(250,400,300,400);
        LaminaGrafico milamina=new LaminaGrafico();
        add(milamina);
        setVisible(true);
    }
}
```

Aprovechamos el Código seleccionado en el recuadro rojo que pasamos a la clase principal llamada `init()`, a continuación eliminados la clase `MarcoGrafico`.

Vamos a ejecutar para poderlo ver en el visor de Applet.



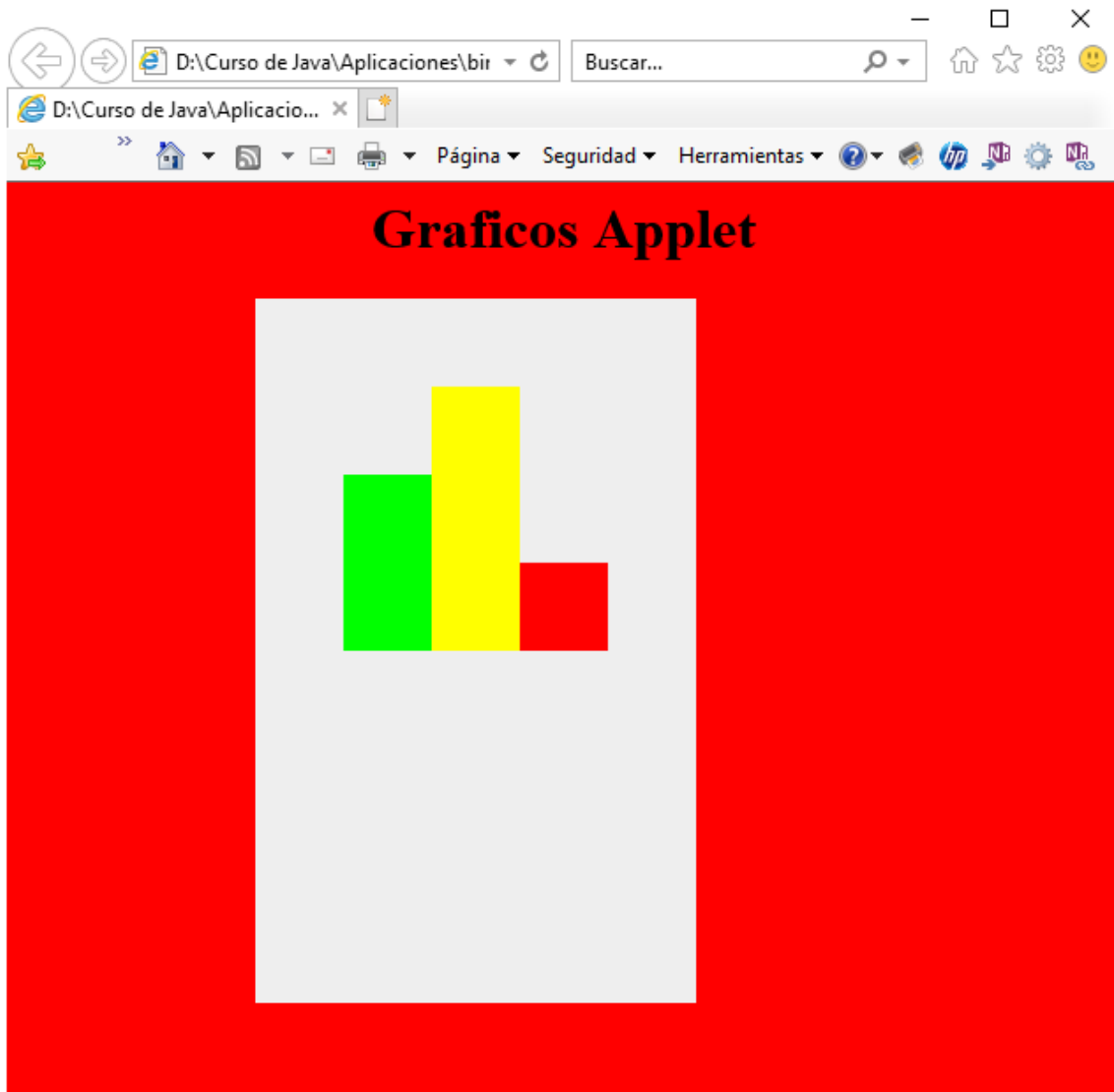
Ahora nos toca crear el archivo html que cargue el archivo class.

Vamos a crear un archivo llamado Grafico\_Applet.html en la carpeta Bin de Aplicaciones del Curso Java.

```
21 class LaminaGrafico extends JPanel{
22
23
24 public void paint(Graphics g){
25
26     super.paintComponent(g);
27
28     Graphics2D g2=(Graphics2D) g;
29
30     g2.rotate( 180/180.0 * Math.PI, 125, 100);
31
32     Rectangle2D graf1=new Rectangle2D.Double(50,0,50,50);
33
34     Rectangle2D graf2=new Rectangle2D.Double(100,0,50,150);
35
36     Rectangle2D graf3=new Rectangle2D.Double(150,0,50,100);
37
38     g2.setPaint(Color.RED);
39
40     g2.fill(graf1);
41
42     g2.setPaint(Color.YELLOW);
43
44     g2.fill(graf2);
45
46     g2.setPaint(Color.GREEN);
47
48     g2.fill(graf3);
49
50 }
51
52 }
```

Y hemos guardado los cambios.





Si abrimos el archivo con un navegador veremos los cambios, pero para ello tenemos que cerrar el navegador y abrirlo de nuevo ya que si esta ya está abierto este no se actualiza automáticamente.

Agregaremos el siguiente código al archivo html.

```
Gráfico_Applet.html: Bloc de notas
Archivo Edición Formato Ver Ayuda
<html>

<style>
#miapplet{
margin:auto;
position:relative;
width: 350px;
}
h1{
text-align:center;
}
</style>
<body bgcolor="red">
<h1>Gráficos Applet</h1>
<div id="miapplet">
<applet code="mis applets/Gráfico_Applet.class" width="250" height="400" >
<param name="barra.1" value="75">
<param name="barra.2" value="25">
<param name="barra.3" value="55">
</applet>
</div>
</body>
</html>
```

Son los parámetros que le pasamos.

Y guardamos los cambios.

Nos vamos a nuestro código java.

```
10 public void init() {
11     int numero1=Integer.parseInt(getParameter("barra.1"));
12     int numero2=Integer.parseInt(getParameter("barra.2"));
13     int numero3=Integer.parseInt(getParameter("barra.3"));
14
15     LaminaGrafico milamina=new LaminaGrafico(numero1, numero2, numero3);
16
17     add(milamina);
18
19 }
20
21
22
23 }
```

En la clase init() agregamos el siguiente código:

Las variables numero1, numero2, numero3 reciben los valores por parámetro que le pasamos del archivo html, que luego pasamos por parámetros al método LaminaGrafico.

```

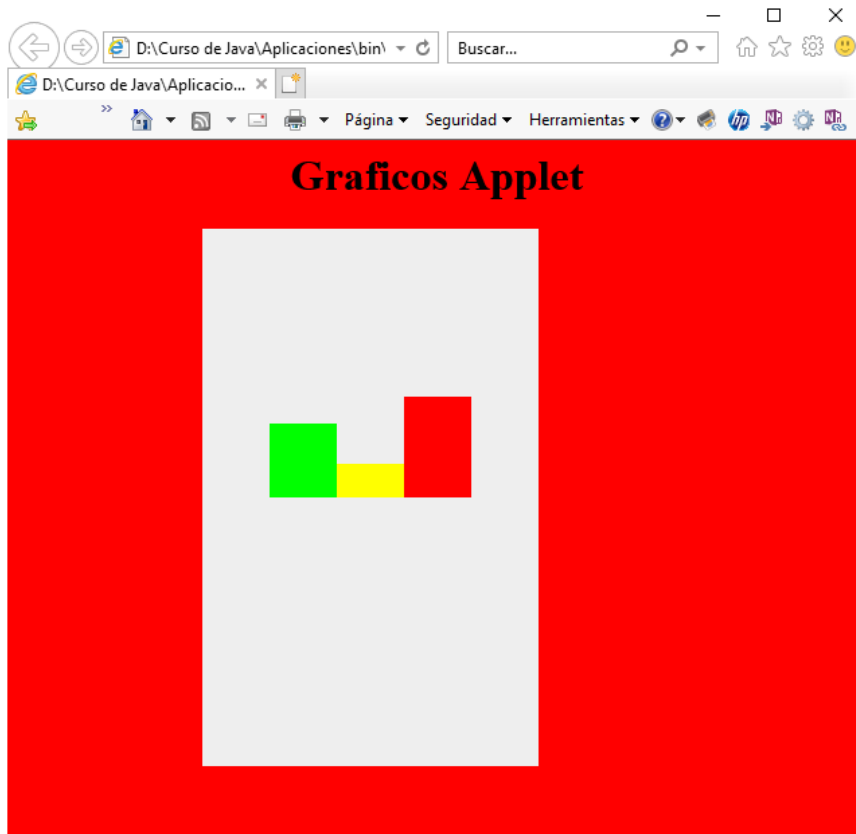
26 class LaminaGrafico extends JPanel{
27
28 public LaminaGrafico(int a, int b, int c) {
29
30     a1=a;
31     a2=b;
32     a3=c;
33 }
34 public void paint(Graphics g){
35
36     super.paintComponent(g);
37
38     Graphics2D g2=(Graphics2D) g;
39
40     g2.rotate( 180/180.0 * Math.PI, 125, 100);
41
42     Rectangle2D graf1=new Rectangle2D.Double(50,0,50,a1);
43
44     Rectangle2D graf2=new Rectangle2D.Double(100,0,50,a2);
45
46     Rectangle2D graf3=new Rectangle2D.Double(150,0,50,a3);
47
48     g2.setPaint(Color.RED);
49
50     g2.fill(graf1);
51
52     g2.setPaint(Color.YELLOW);
53
54     g2.fill(graf2);
55
56     g2.setPaint(Color.GREEN);
57
58     g2.fill(graf3);
59
60 }
61
62 private int a1,a2,a3;
63
64 }

```

Al método LaminaGrafico recibe por parámetros tres variables de tipo int, a, b, c.

Para que puedan pasar a la clase interna, declaramos tres variables a1, a2 y a3 que pueda reconocer dicho método, por lo cual se crea un constructor con las variables a1, a2 y a3 que almacenan los valores que se les pasa por parámetro.

Por último en Rectangle2D al último valor que era fijo se le cambia por su respectiva variable.



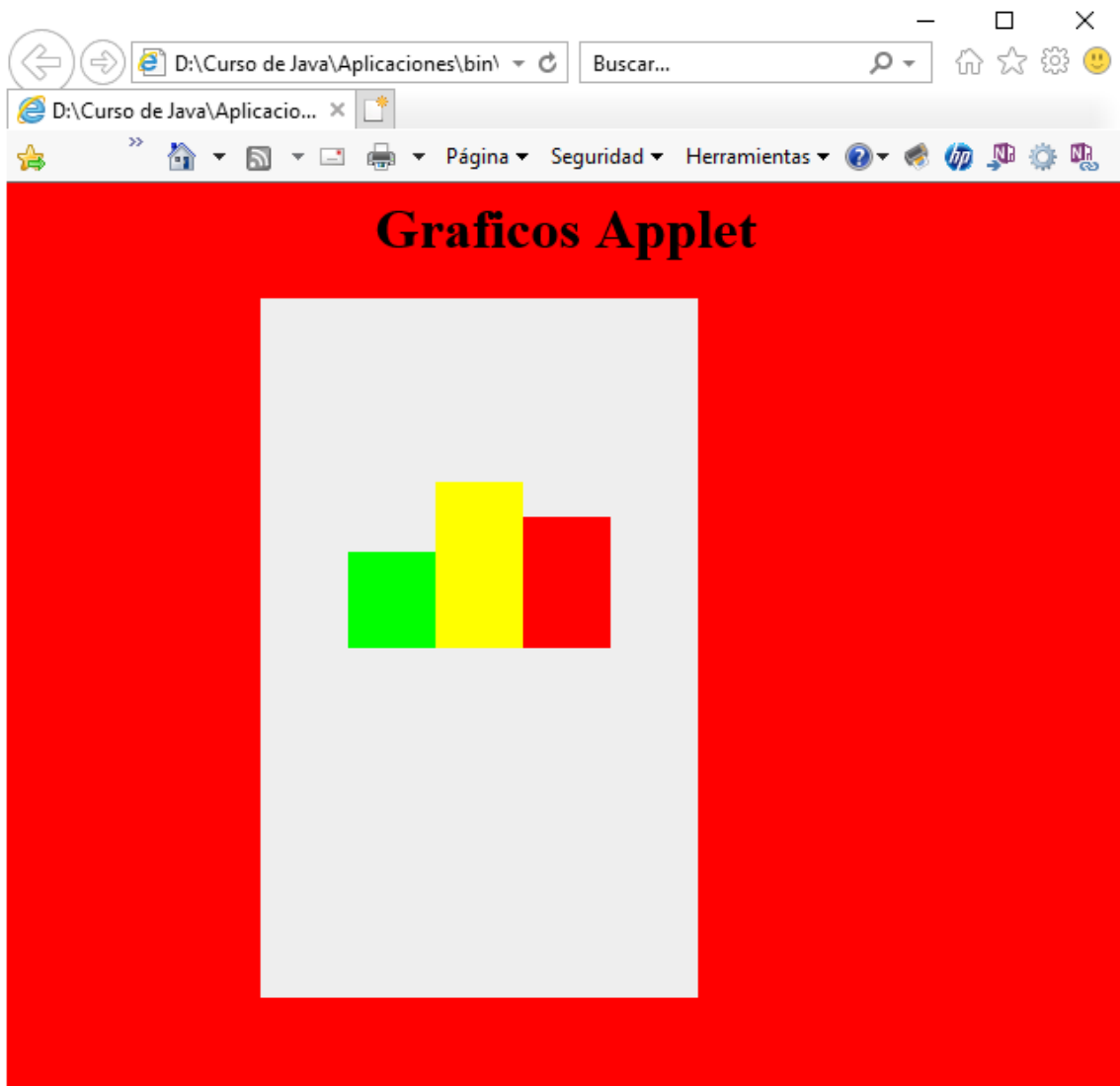
Ya se han cambiado los valores.

Ahora abrimos el código html con el bloc de notas.

```
Gráfico_Applet.html: Bloc de notas
Archivo Edición Formato Ver Ayuda
<html>
<style>
#miapplet{
margin:auto;
position:relative;
width: 350px;
}
h1{
text-align:center;
}
</style>
<body bgcolor="red">
<h1>Graficos Applet</h1>
<div id="miapplet">
<applet code="mis_applets/Gráfico_Applet.class" width="250" height="400" >
<param name="barra.1" value="75">
<param name="barra.2" value="95">
<param name="barra.3" value="55">
</applet>
</div>
</body>
</html>
```

Hemos cambiado el valor de la barra.2 a 95 y guardamos los cambios.

Visualizamos en navegador que lo teníamos minimizado con el Applet cargado y solo pulsamos F5 haber si este se actualiza.



Efectivamente vemos como las barras se han actualizado.

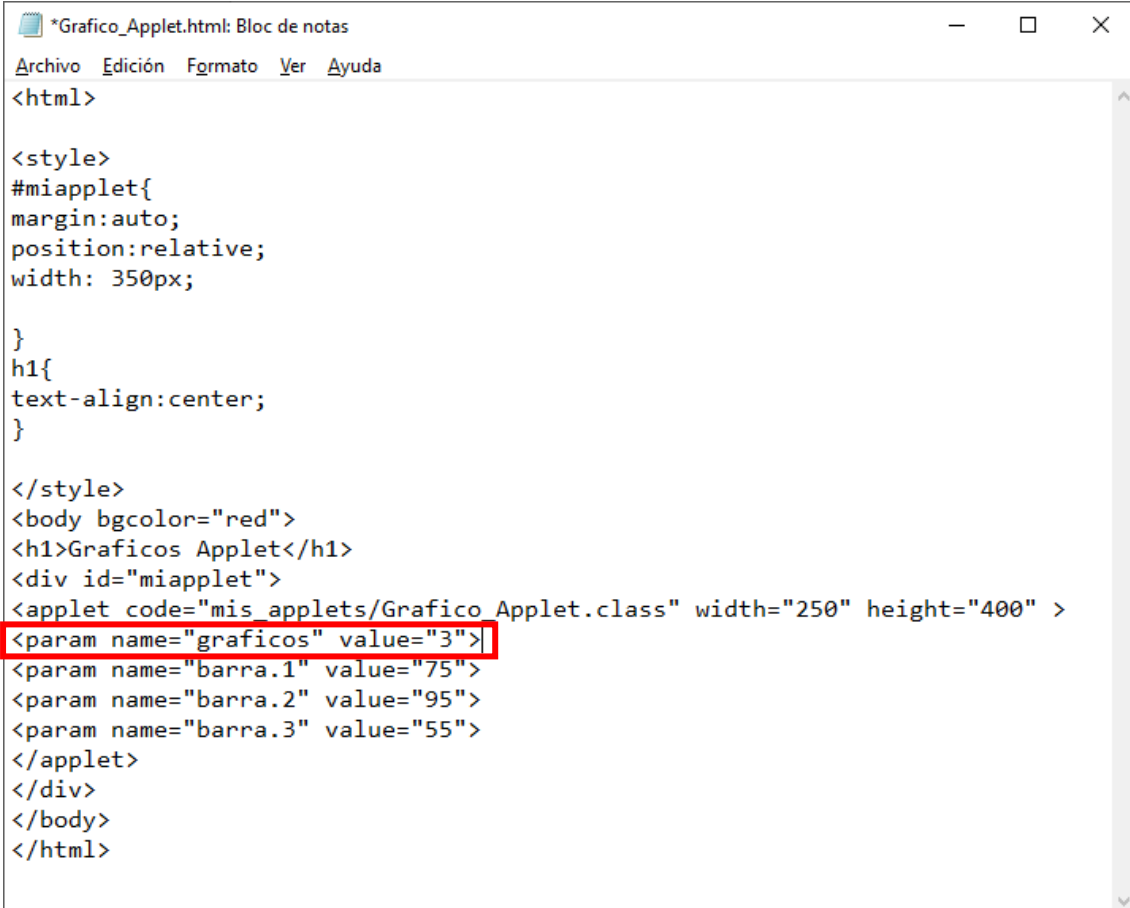




## Despliegue Aplicaciones. Applets V. (Vídeo 136)

### Paso de parámetros a Applets II

Vamos a mejorar el paso de parámetros para modificar el número de barras.



```
*Grafico_Applet.html: Bloc de notas
Archivo Edición Formato Ver Ayuda
<html>

<style>
#miapplet{
margin:auto;
position:relative;
width: 350px;

}
h1{
text-align:center;
}

</style>
<body bgcolor="red">
<h1>Graficos Applet</h1>
<div id="miapplet">
<applet code="mis_applets/Grafico Applet.class" width="250" height="400" >
<param name="graficos" value="3">
<param name="barra.1" value="75">
<param name="barra.2" value="95">
<param name="barra.3" value="55">
</applet>
</div>
</body>
</html>
```

En el documento html agregamos la siguiente línea y guardamos los cambios.

```

1 package mis_applets;
2 import javax.swing.*;
3 import java.awt.*;
4 import java.awt.geom.*;
5
6 public class Grafico_Applet extends JApplet {
7     public void init() {
8
9         String numero=getParameter("graficos");
10        int n_graficos=Integer.parseInt(numero);
11        int arrayGraficos[]=new int[n_graficos];
12        for(int i=0; i<arrayGraficos.length;i++) {
13            arrayGraficos[i]=Integer.parseInt(getParameter("barra."+(i+1)));
14        }
15        LaminaGrafico milamina=new LaminaGrafico(arrayGraficos);
16
17        add(milamina);
18    }
19 }
20
21 class LaminaGrafico extends JPanel{
22
23     public LaminaGrafico(int valores[]) {
24         valores_graficos=valores;
25     }
26     public void paint(Graphics g){
27         super.paintComponent(g);
28         Graphics2D g2=(Graphics2D) g;
29         g2.rotate( 180/180.0 * Math.PI, 125, 100);
30
31         int pos_x=0;
32         for(int i=0; i<valores_graficos.length; i++){
33             int color_rojo=(int)Math.round(Math.random()*200);
34             int color_verde=(int)Math.round(Math.random()*200);
35             int color_azul=(int)Math.round(Math.random()*200);
36
37             Rectangle2D graf=new Rectangle2D.Double(pos_x,0,25,valores_graficos[i]);
38             Color color_barra=new Color(color_rojo,color_verde, color_azul);
39             g2.setPaint(color_barra);
40             g2.fill(graf);
41             pos_x+=25;
42         }
43     }
44     int[] valores_graficos;
45 }

```

Modificamos el código tal como se muestra en el ejemplo.



```
Gráfico_Applet.html: Bloc de notas
Archivo Edición Formato Ver Ayuda
<html>

<style>
#miapplet{
margin:auto;
position:relative;
width: 350px;

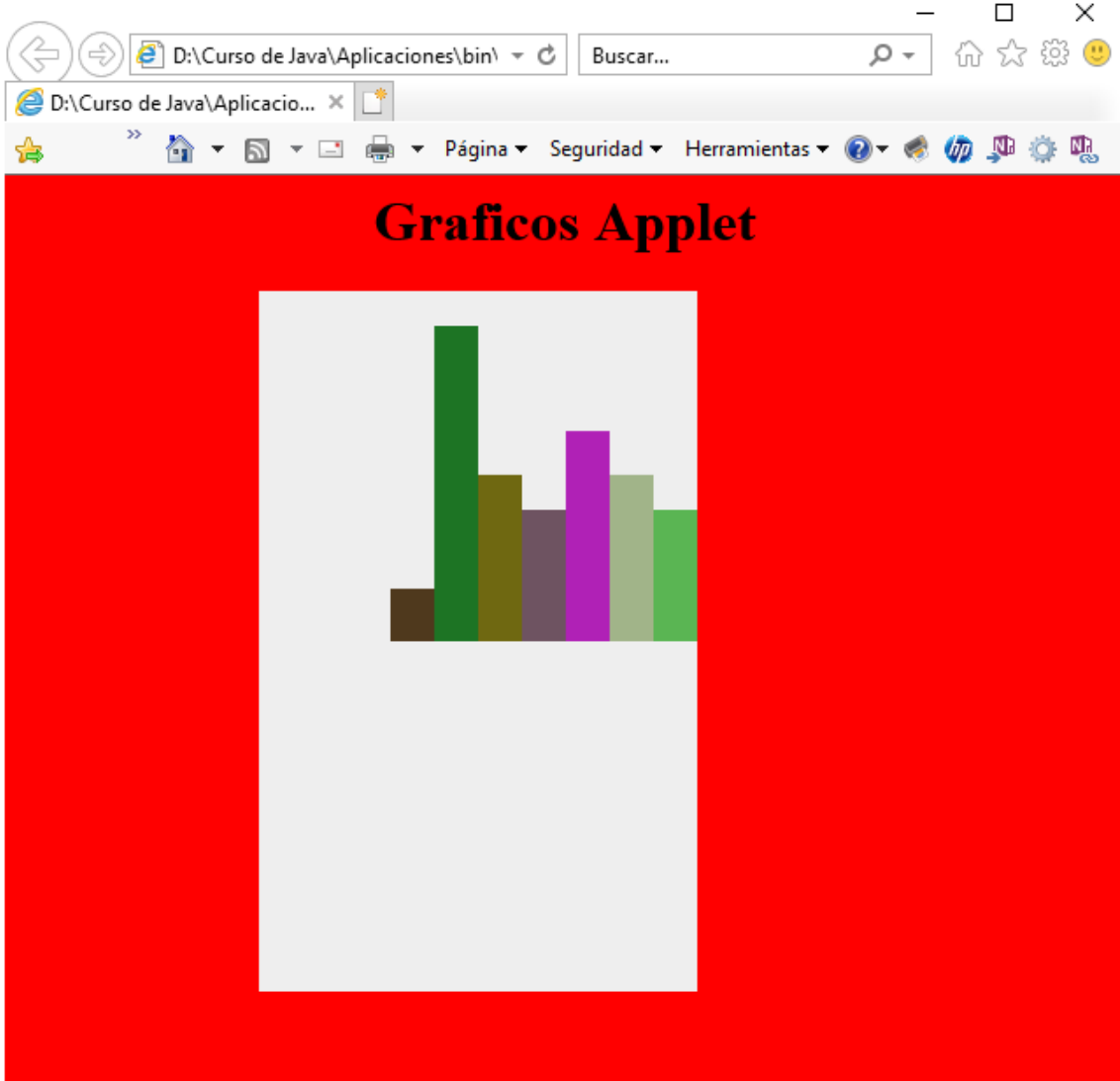
}
h1{
text-align:center;
}
</style>
<body bgcolor="red">
<h1>Gráficos Applet</h1>
<div id="miapplet">
<applet code="mis_applets/Gráfico_Applet.class" width="250" height="400" >
<param name="graficos" value="7">
<param name="barra.1" value="75">
<param name="barra.2" value="95">
<param name="barra.3" value="120">
<param name="barra.4" value="75">
<param name="barra.5" value="95">
<param name="barra.6" value="180">
<param name="barra.7" value="30">
</applet>
</div>
</body>
</html>
```

Modificamos el código del documento html y guardamos los cambios.

Abrimos dicho documento con el navegador.

Este será el resultado:

Browser window showing a Java applet titled "Graficos Applet". The applet displays a bar chart with six bars of varying heights and colors (brown, dark green, olive, purple, light green, and lime green).



Bar Color	Approximate Height
Brown	Low
Dark Green	High
Olive	Medium
Purple	Medium-High
Light Green	Medium
Lime Green	Low-Medium

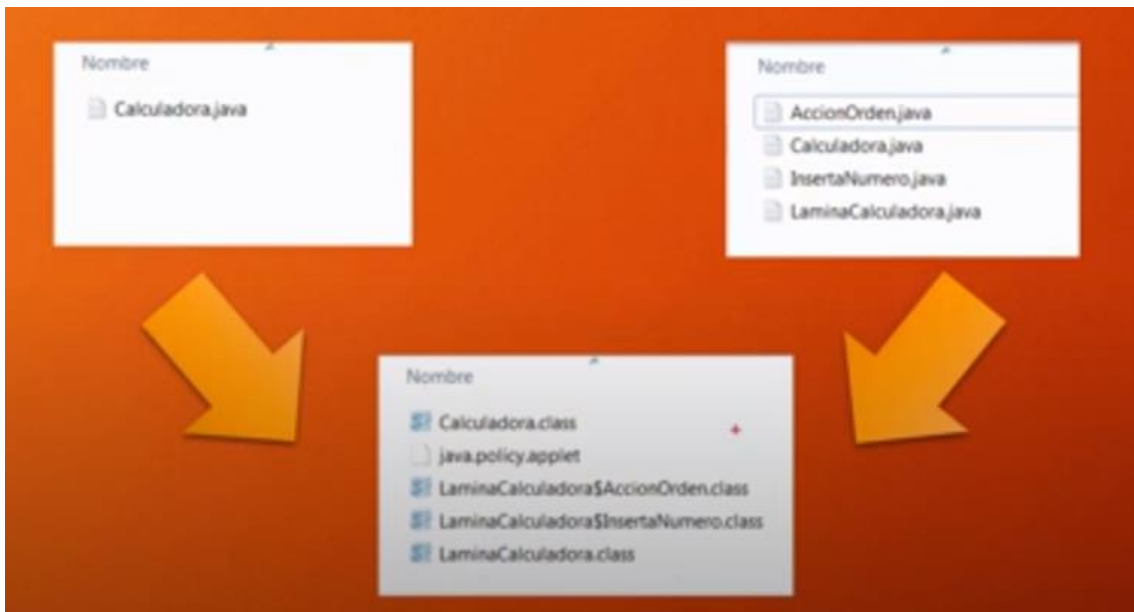




## Despliegue Aplicaciones. Archivos JAR I. (Vídeo 137)

Empaquetado y distribución de aplicaciones.

Modularización consiste en dividir nuestros programas en distintos ficheros.



Tanto si nuestra aplicación es de un fichero o varios lo empaquetará de la misma forma.



Una vez empaquetado este será un archivo JAR.



En un paquete solo puede haber el proyecto que vayamos a empaquetar, en nuestro caso queremos empaquetar la calculadora, pero esta se encuentra mezcladas con otros proyectos, por ese motivo vamos a crear un paquete nuevo llamado Calculadora.

Vamos a crear un proyecto nuevo llamado Calculadora.

**New Java Project**

Create a Java project in the workspace or in an external location.

Project name:

Use default location

Location:  [Browse...](#)

**JRE**

Use an execution environment JRE:

Use a project specific JRE:

Use default JRE 'jre7' and workspace compiler preferences [Configure JREs...](#)

**Project layout**

Use project folder as root for sources and class files

Create separate folders for sources and class files [Configure default...](#)

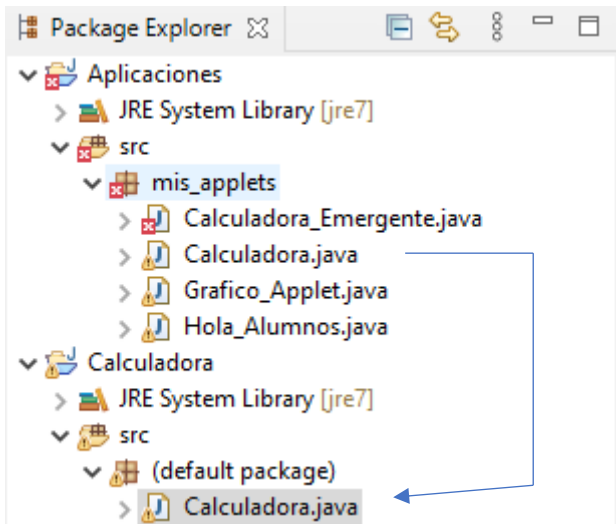
**Working sets**

Add project to working sets [New...](#)

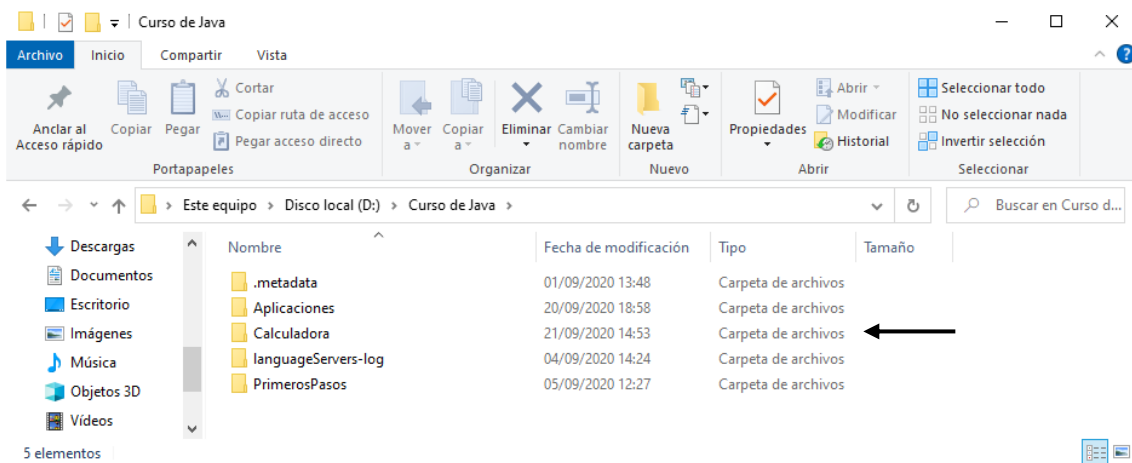
Working sets:  [Select...](#)

**i** The default compiler compliance level for the current workspace is 14. The new project will use a project specific compiler compliance level of 1.7.

[?](#) [< Back](#) [Next >](#) [Finish](#) [Cancel](#)

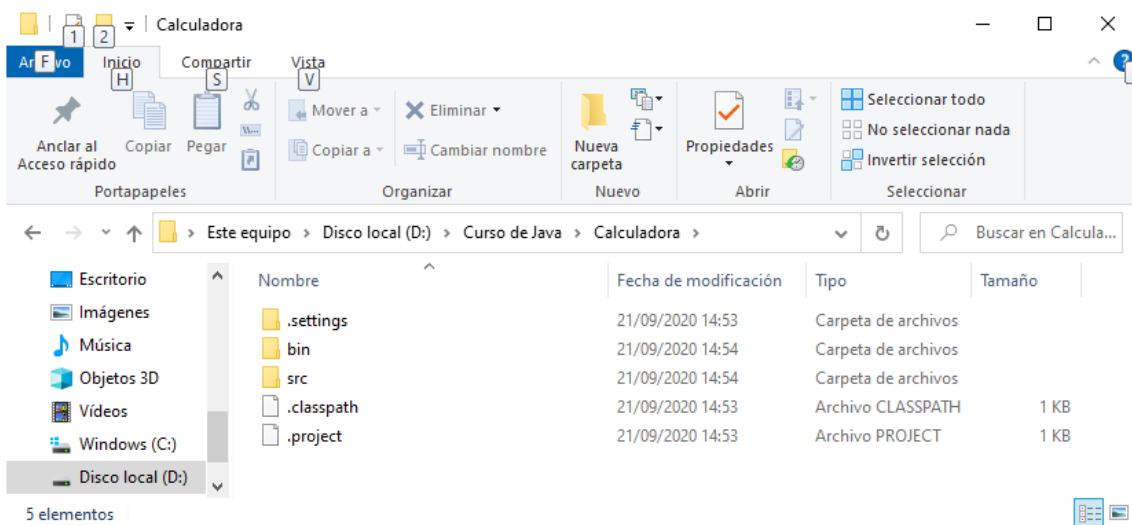


La calculadora que tenemos en mis\_applets la copiamos al nuevo proyecto que hemos creado con el nombre Calculadora dentro de la carpeta src.

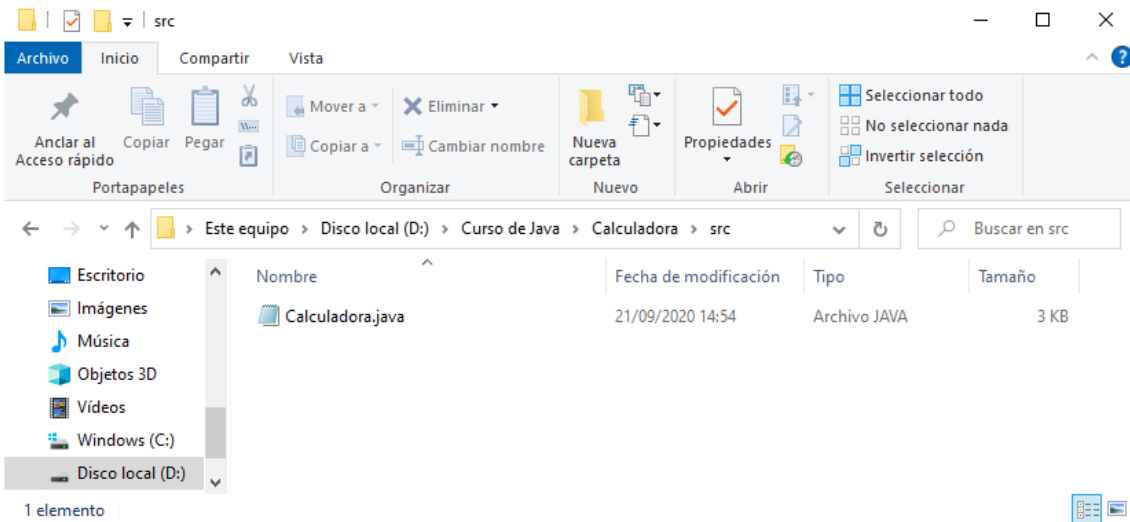


Ya tenemos la carpeta Calculadora, además de Aplicaciones y PrimerosPasos.

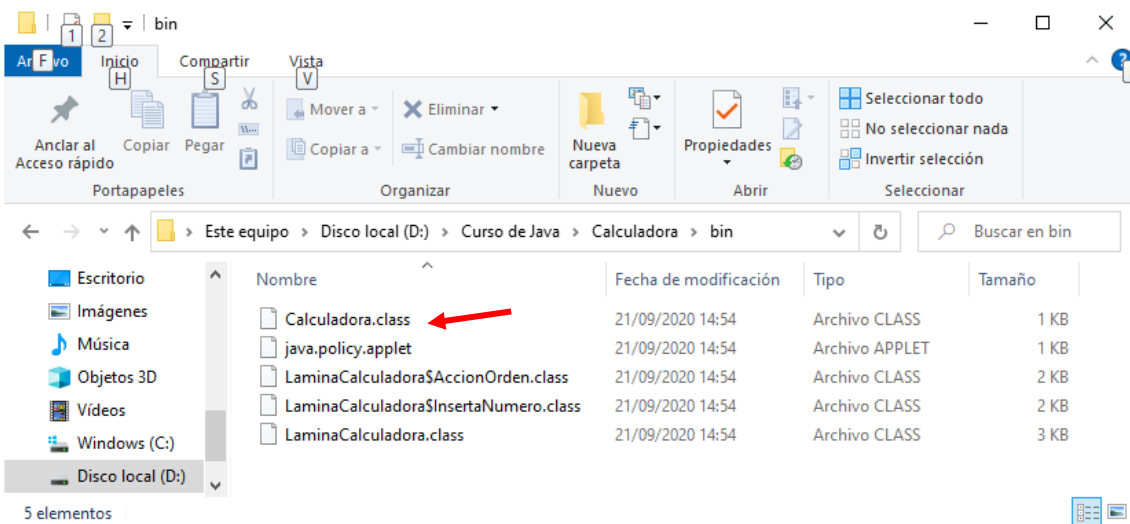
Si la abrimos veremos las siguientes directorios y archivos.



Si nos vamos a la carpeta src veremos un único archivo.



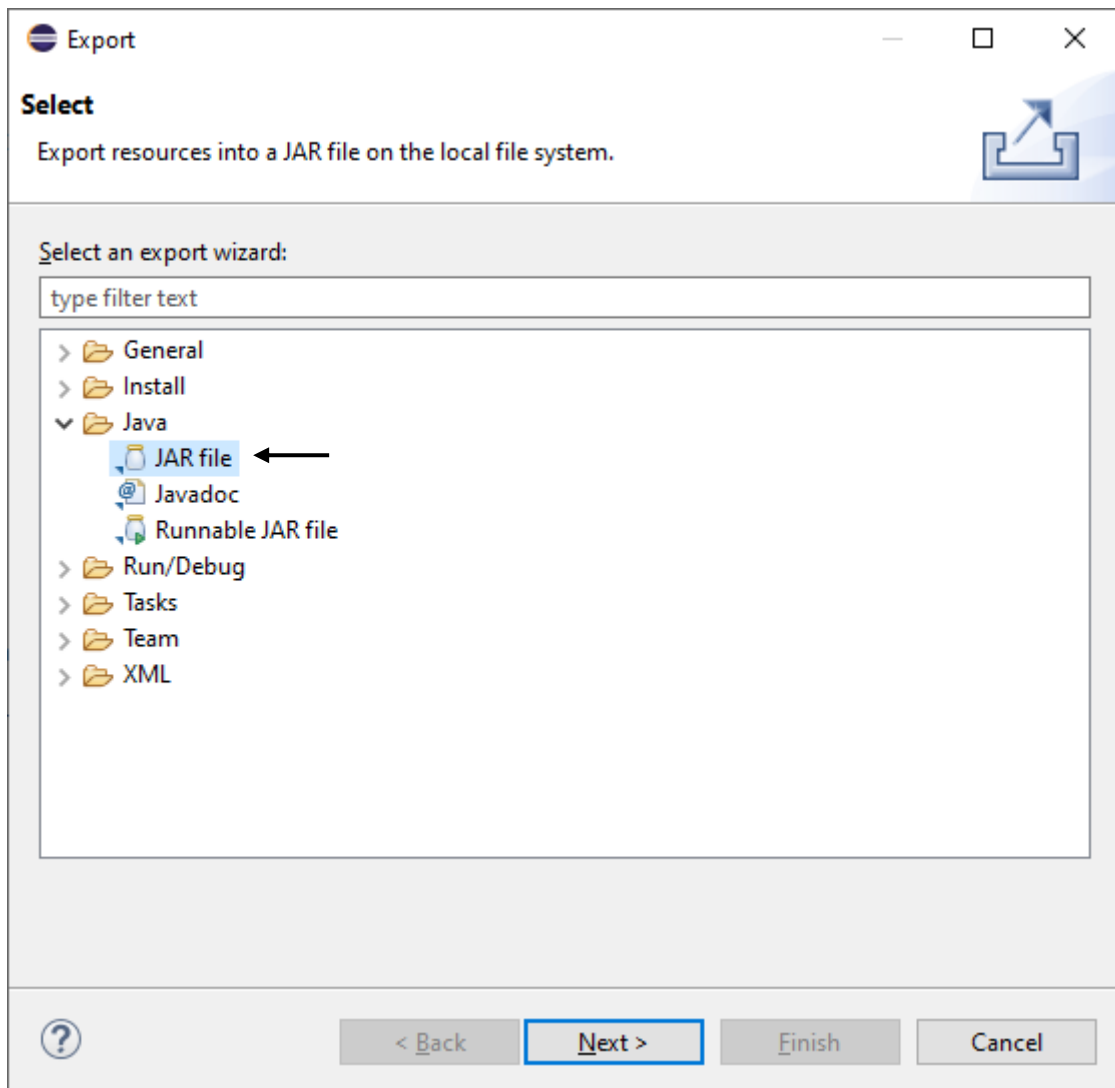
Si nos vamos a la carpeta bin veremos un archivo por clase.



Como archivo principal Calculadora.class

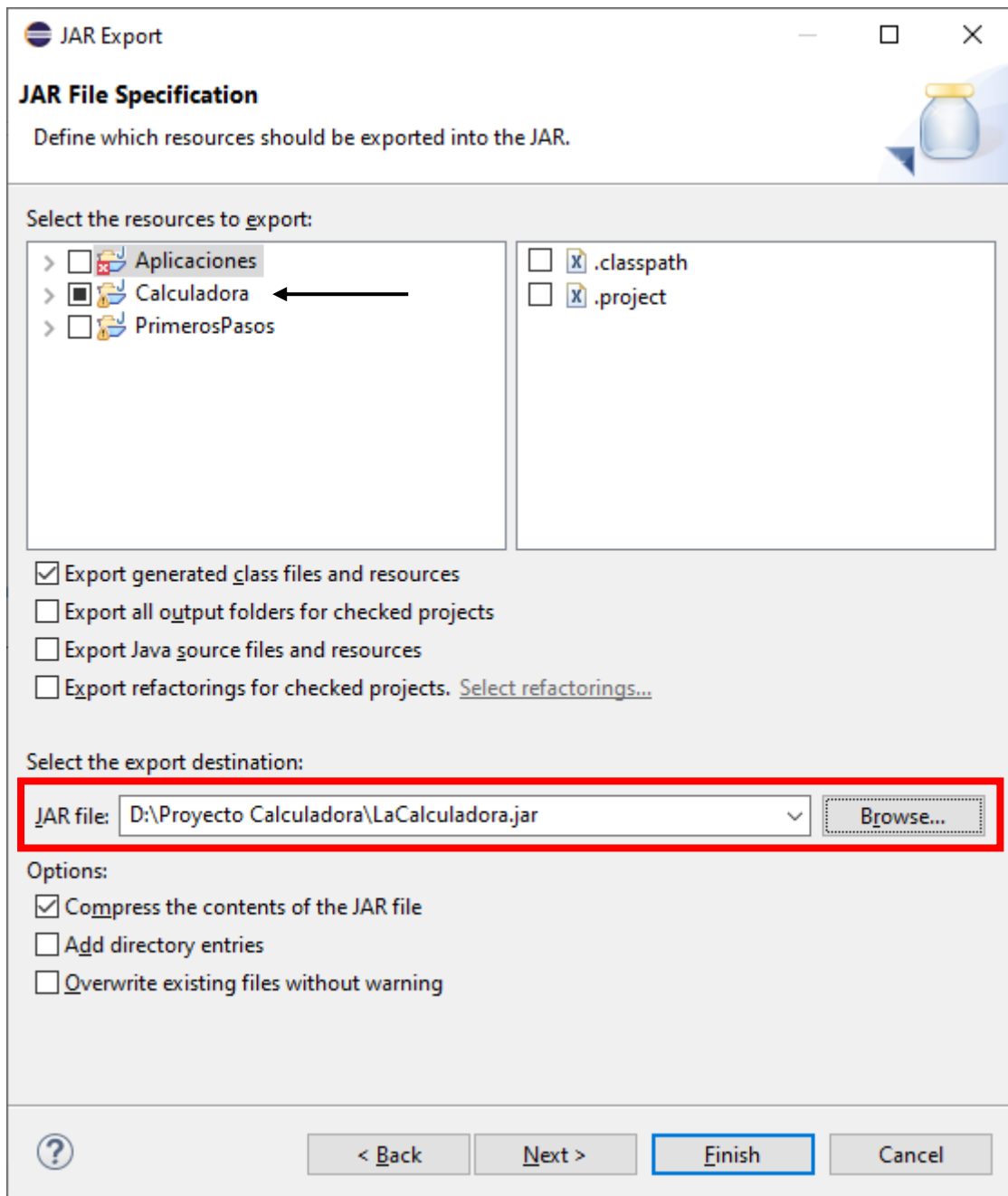
Ahora para empaquetar con el archivo abierto del menú File seleccionaremos Export...

Saldrá un asistente que nos guiará paso a paso en la exportación.



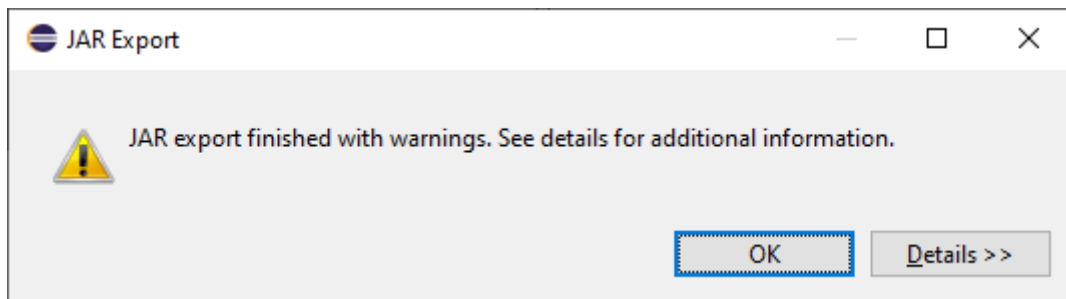
Seleccionamos JAR file seguido del botón Next.





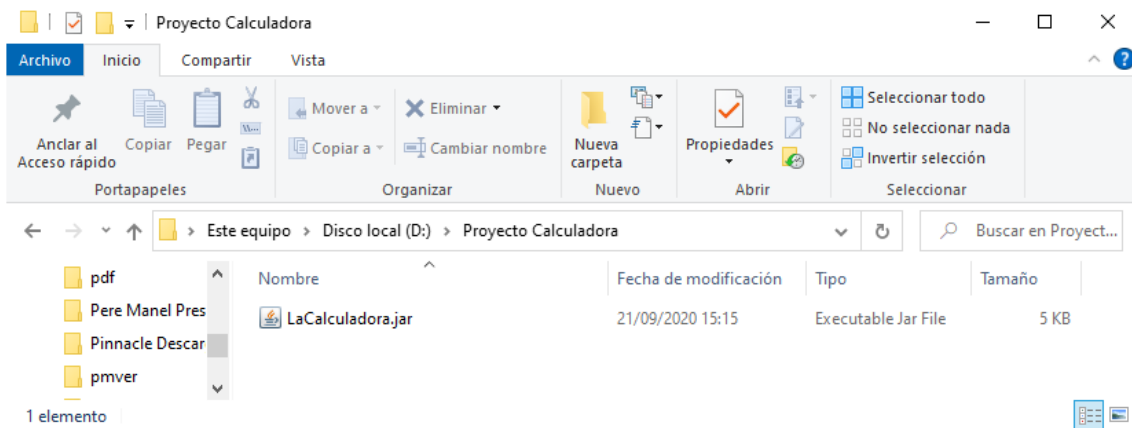
Marcamos el proyecto que queremos empaquetar y donde queremos guardar el proyecto empaquetado y le decimos la ubicación no nombre para dicho proyecto.

Seguidamente le damos a Finish.



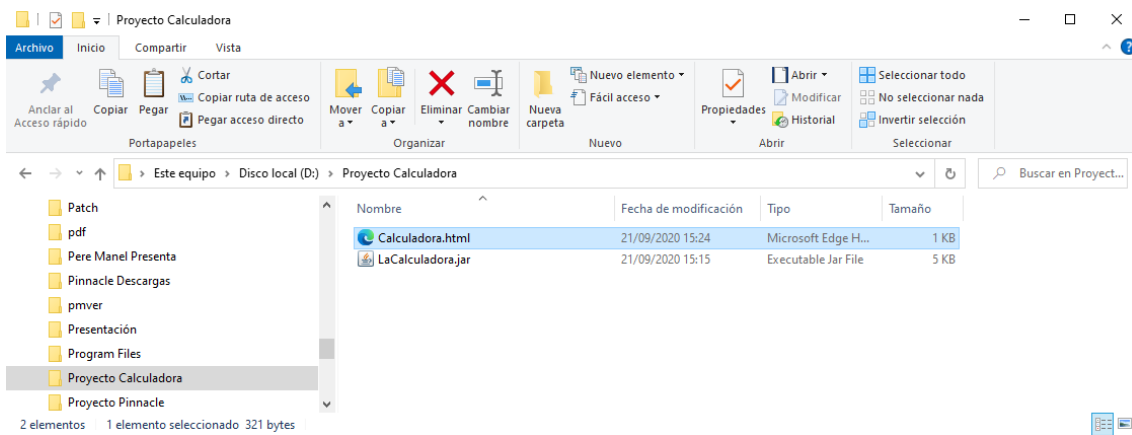
Respondemos con OK.

Si nos vamos a la carpeta donde exportamos el proyecto.



Encontraremos LaCalculadora.jar

Tenemos que hacer un archivo html.

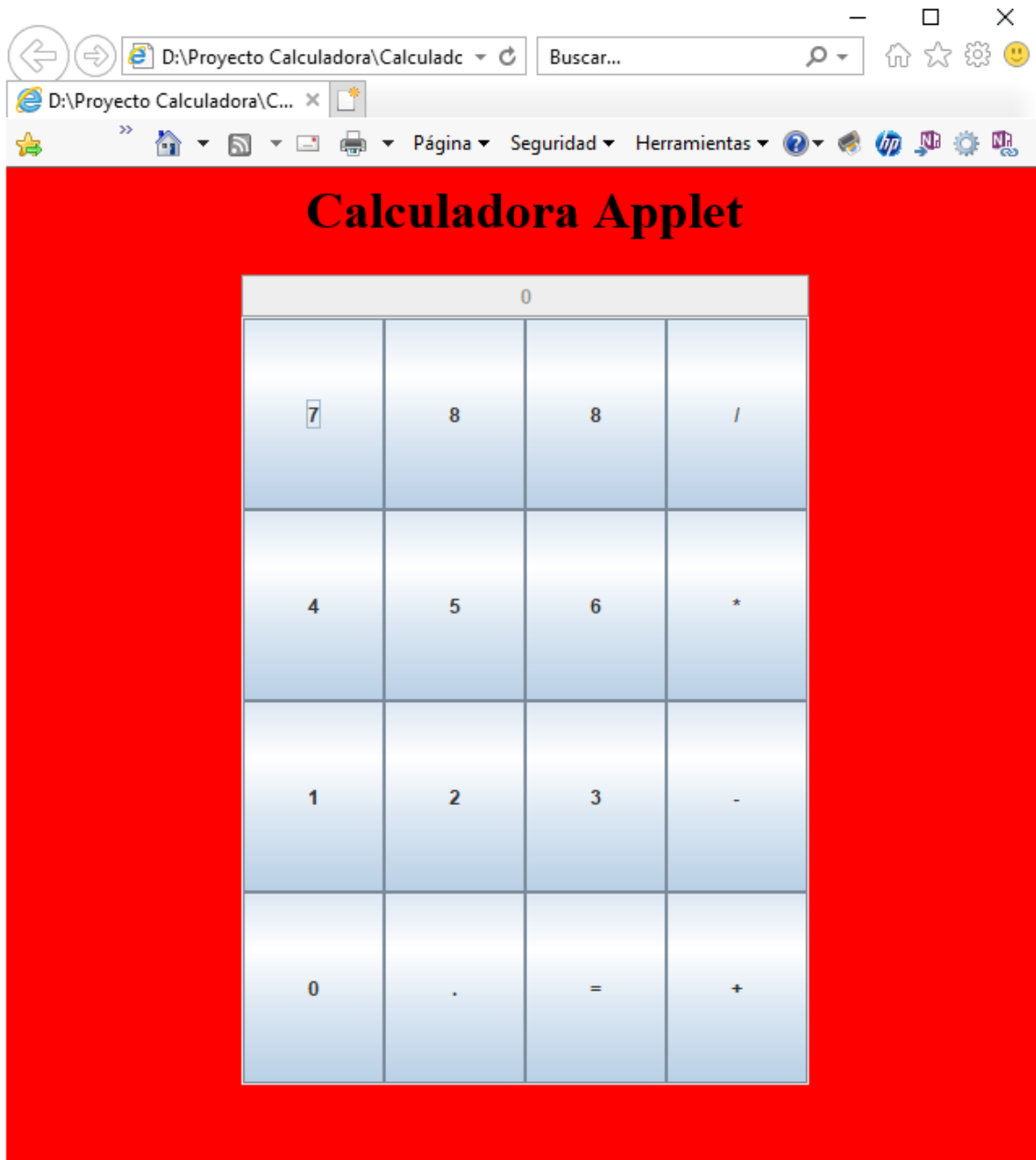


Este es su contenido.

```
Calculadora.html: Bloc de notas
Archivo Edición Formato Ver Ayuda
<html>

<style>
#miapplet{
margin:auto;
position:relative;
width: 350px;
}
h1{
text-align:center;
}
</style>
<body bgcolor="red">
<h1>Calculadora Applet</h1>
<div id="miapplet">
<applet code="Calculadora.class" archive="LaCalculadora.jar" width="350" height="500" >
</applet>
</div>
</body>
</html>
```

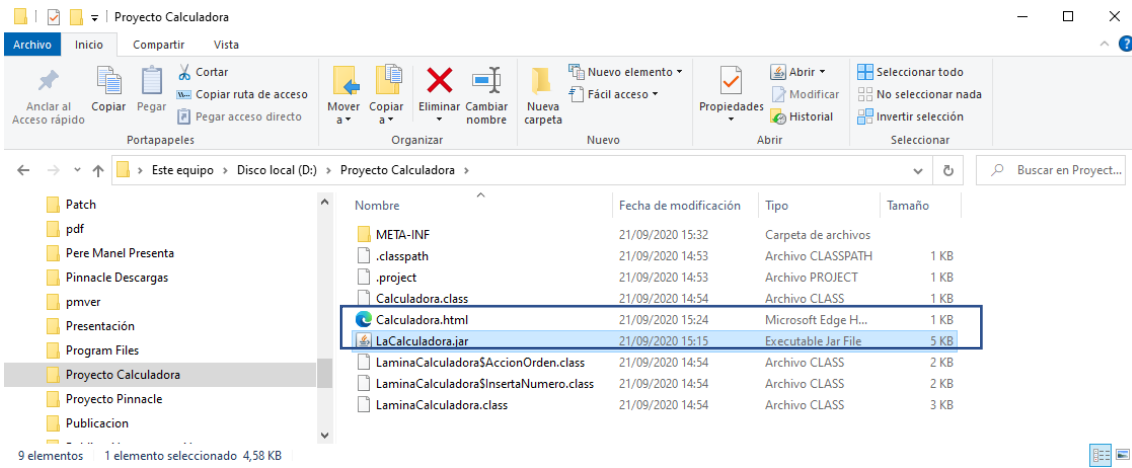
Una vez guardados los cambios lo abrimos con el navegador.



Ya podemos distribuirlo o utilizarlo en el navegador, así como poder realizar su distribución.

El archivo LaCalculadora.jar lo podemos descomprimir para ver los archivos que contiene.

Si lo seleccionamos con el botón derecho y le decimos Extraer aquí veremos los archivos que contiene.



Lo remarcados son los que habían y el resto son los archivos descomprimidos de LaCalculadora.jar.

También se pueden hacer ejecutables con archivos jar.



## Despliegue Aplicaciones. Archivos JAR II. Firmar JAR I (Vídeo 138)

¿Qué es firmar un JAR?

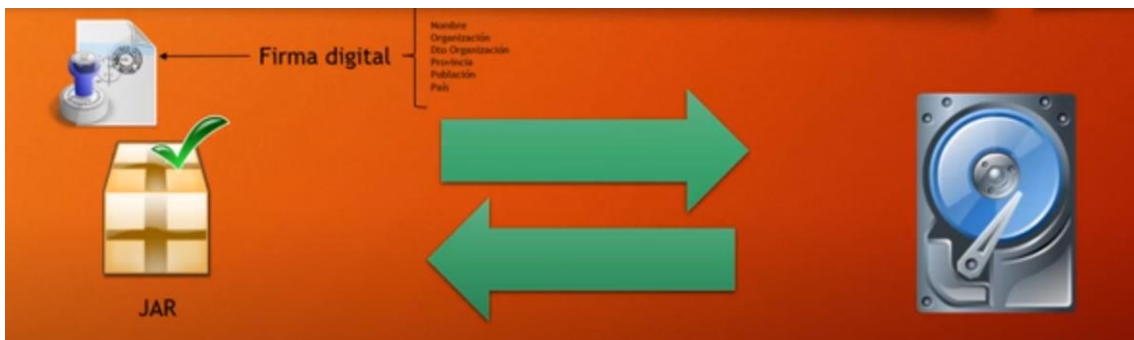


No debemos ejecutar un archivo JAR si no sabemos de su procedencia.



La manera de garantizar la aplicación JAR hay lo que denomina firma digital.

La firma digital lleva Nombre, Organización, Dto. Organización, Provincia, Población y País.



¿Cómo firmar un JAR?

1. Crear un certificado (firma) para la Aplicación (JAR)
    - a. Necesitamos herramienta keytool
  2. Firmar la aplicación (JAR) con el certificado generado
    - a. Necesitamos jarstgen
- ¿Dónde conseguir estas herramientas keytool y jarsigner?
    - Vienen con el JDK (Java Development Kit) de Oracle.

Si vamos al siguiente enlace:

<https://www.oracle.com/technetwork/es/java/javase/downloads/index.html>



Iniciar sesión

País

Llamar

- Java SE
- Soporte para Java SE
- Java Embedded
- Java EE
- Java ME
- Java FX
- Java DB
- Web Tier
- Java Card
- Comunidad

- Resumen
- Descargas
- Documentación
- Comunidad
- Tecnología
- Formación

### Descargas de Java SE



Java Platform (JDK) 8u111 / 8u112



NetBeans con JDK 8

#### Java Platform, Standard Edition

##### Java SE 8u111 / 8u112

Java SE 8u111 incluye importantes soluciones de seguridad. Oracle recomienda que todos los usuarios de Java SE 8 actualicen esta versión. Java SE 8u112 es una actualización con un conjunto de parches, que incluye todas las características adicionales de 8u111 (descritas en las notas de la versión). [Lea más aquí \(en inglés\)](#) ▶

##### Importante cambio planificado para MD5-signed JARs

A partir de las versiones de la revisión crítica de abril, previstas para el 18 de abril de 2017, todas las versiones de JRE tratarán a los JARs firmados con MD5 como no firmado. [Obtenga más información y vea las instrucciones de prueba \(en inglés\)](#). Para obtener más información sobre el soporte del algoritmo criptográfico, por favor chequee este documento: [JRE and JDK Crypto Roadmap \(en inglés\)](#)

- [Instrucciones de instalación \(en inglés\)](#)
- [Notas de la versión \(en inglés\)](#)
- [Licencia de Oracle \(en inglés\)](#)
- [Productos Java SE \(en inglés\)](#)
- [Licencias de terceros \(en inglés\)](#)
- [Configuraciones del sistema certificadas \(en inglés\)](#)

##### Archivos LeaMe

- [Archivo LeaMe JDK \(en inglés\)](#)
- [Archivo LeaMe JRE \(en inglés\)](#)

JDK  
[DOWNLOAD](#)

Servidor JRE  
[DOWNLOAD](#)

JRE  
[DOWNLOAD](#)



Donde podremos descargar el JDK, si lo seleccionamos.

Java SE Development Kit 8u261		
This software is licensed under the Oracle Technology Network License Agreement for Oracle Java SE		
Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	73.4 MB	<a href="#">jdk-8u261-linux-arm32-vfp-hflt.tar.gz</a>
Linux ARM 64 Hard Float ABI	70.3 MB	<a href="#">jdk-8u261-linux-arm64-vfp-hflt.tar.gz</a>
Linux x86 RPM Package	121.92 MB	<a href="#">jdk-8u261-linux-i586.rpm</a>
Linux x86 Compressed Archive	136.81 MB	<a href="#">jdk-8u261-linux-i586.tar.gz</a>
Linux x64 RPM Package	121.53 MB	<a href="#">jdk-8u261-linux-x64.rpm</a>
Linux x64 Compressed Archive	136.48 MB	<a href="#">jdk-8u261-linux-x64.tar.gz</a>
macOS x64	203.94 MB	<a href="#">jdk-8u261-macosx-x64.dmg</a>
Solaris SPARC 64-bit (SVR4 package)	125.77 MB	<a href="#">jdk-8u261-solaris-sparcv9.tar.Z</a>
Solaris SPARC 64-bit	88.72 MB	<a href="#">jdk-8u261-solaris-sparcv9.tar.gz</a>
Solaris x64 (SVR4 package)	134.23 MB	<a href="#">jdk-8u261-solaris-x64.tar.Z</a>
Solaris x64	92.47 MB	<a href="#">jdk-8u261-solaris-x64.tar.gz</a>
Windows x86	154.52 MB	<a href="#">jdk-8u261-windows-i586.exe</a>
Windows x64	166.28 MB	<a href="#">jdk-8u261-windows-x64.exe</a>

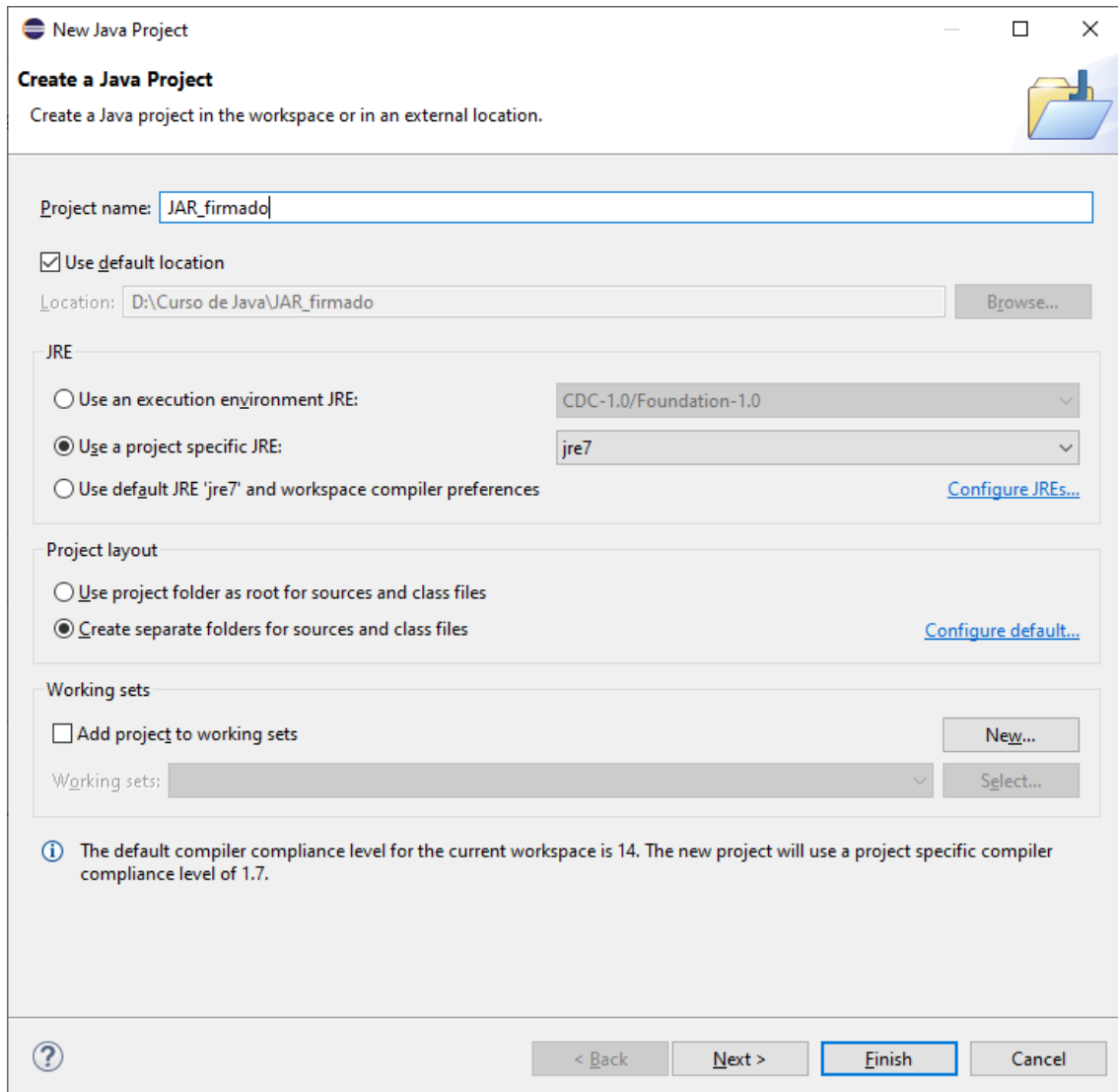


Sí vamos a Archivo de Programas y buscamos la carpeta java encontraremos jdk con su correspondiente versión, si entramos en ella y luego accedemos a la carpeta bin encontraremos estos dos programas.

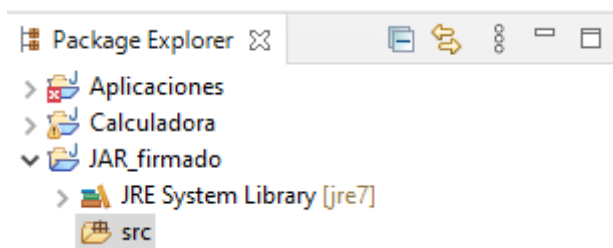
Para eso vamos a realizar el siguiente ejemplo.

Nos vamos a Eclipse.

Creamos un nuevo proyecto llamado JAR\_firmado.



Le damos a Finish.



De nuestro proyecto seleccionamos la carpeta src.

Creamos una nueva clase llamada JAR\_Multimedia.

**New Java Class**

**Java Class**

The use of the default package is discouraged.

Source folder: JAR\_firmado/src Browse...

Package: (default) Browse...

Enclosing type: Browse...

---

Name: Applet\_Multimedia

Modifiers:  public  package  private  protected  
 abstract  final  static

Superclass: java.lang.Object Browse...

Interfaces: Add...  
Remove

Which method stubs would you like to create?

public static void main(String[] args)

Constructors from superclass

Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))

Generate comments

? Finish Cancel

Le damos a Finish.

Vamos a copiar una imagen que puede ser de extensión jpg, png, gif, etc.

Esta la guardaremos en la carpeta bin del proyecto JAR\_firmado.

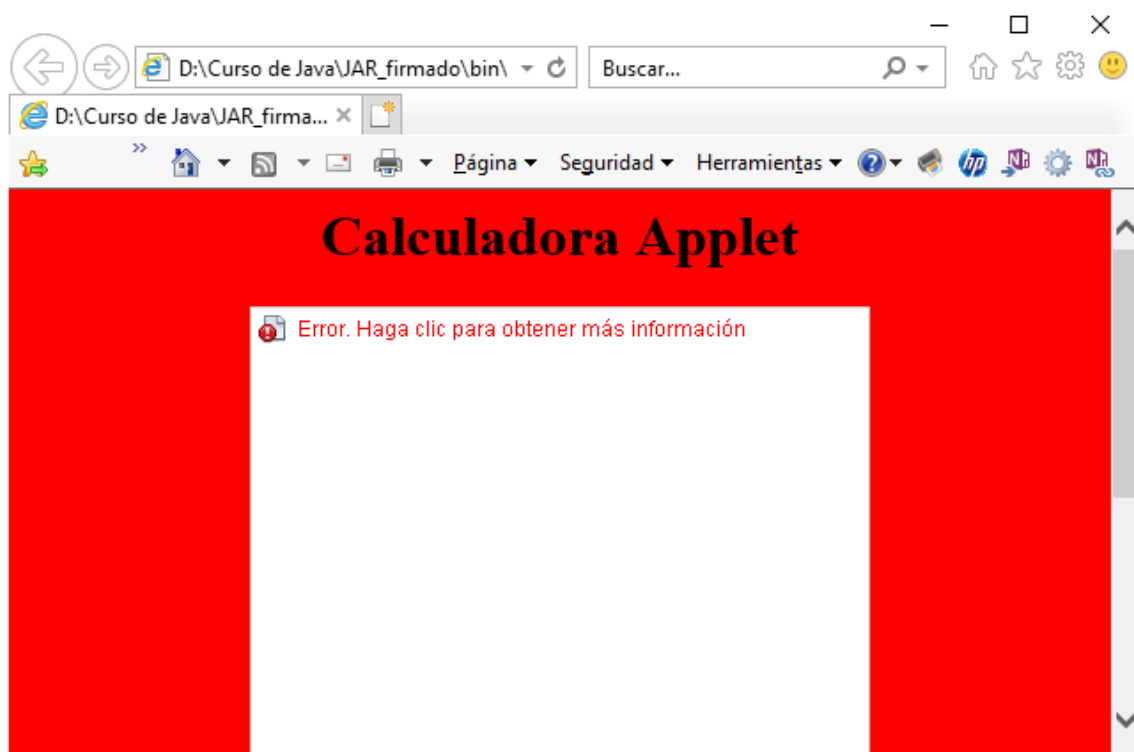
Agregamos un archivo Appelt\_Multimedia.html con el siguiente código:



```
*Applet_Multimedia.html: Bloc de notas
Archivo Edición Formato Ver Ayuda
<html>

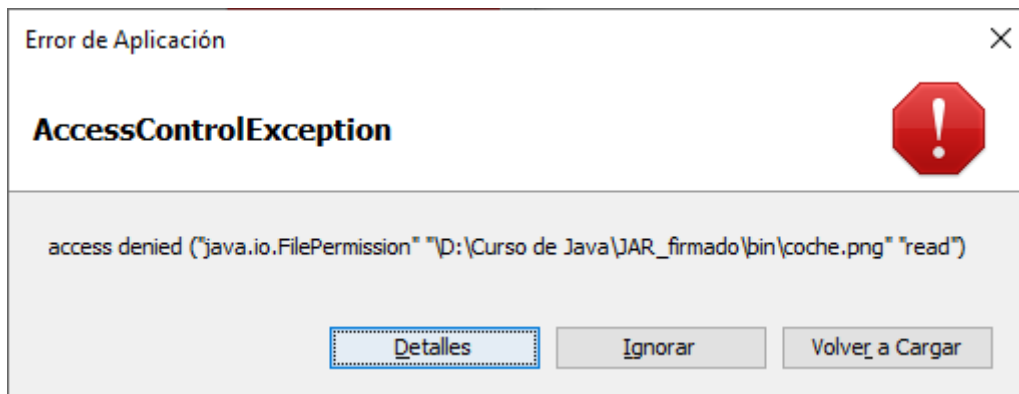
<style>
#miapplet{
margin:auto;
position:relative;
width: 350px;
}
h1{
text-align:center;
}
</style>
<body bgcolor="red">
<h1>Calculadora Applet</h1>
<div id="miapplet">
<applet code="Applet_Multimedia.class" width="350" height="500" >
</applet>
</div>
</body>
</html>
```

Ahora lo intentamos abrir con el navegador.



Sale un error.

Si hacemos un clic con el botón izquierdo verás el siguiente mensaje.

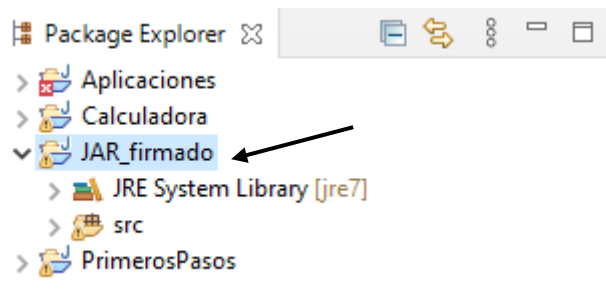


Como solución consiste en empaquetar y firmar, que lo veremos en el siguiente capítulo.

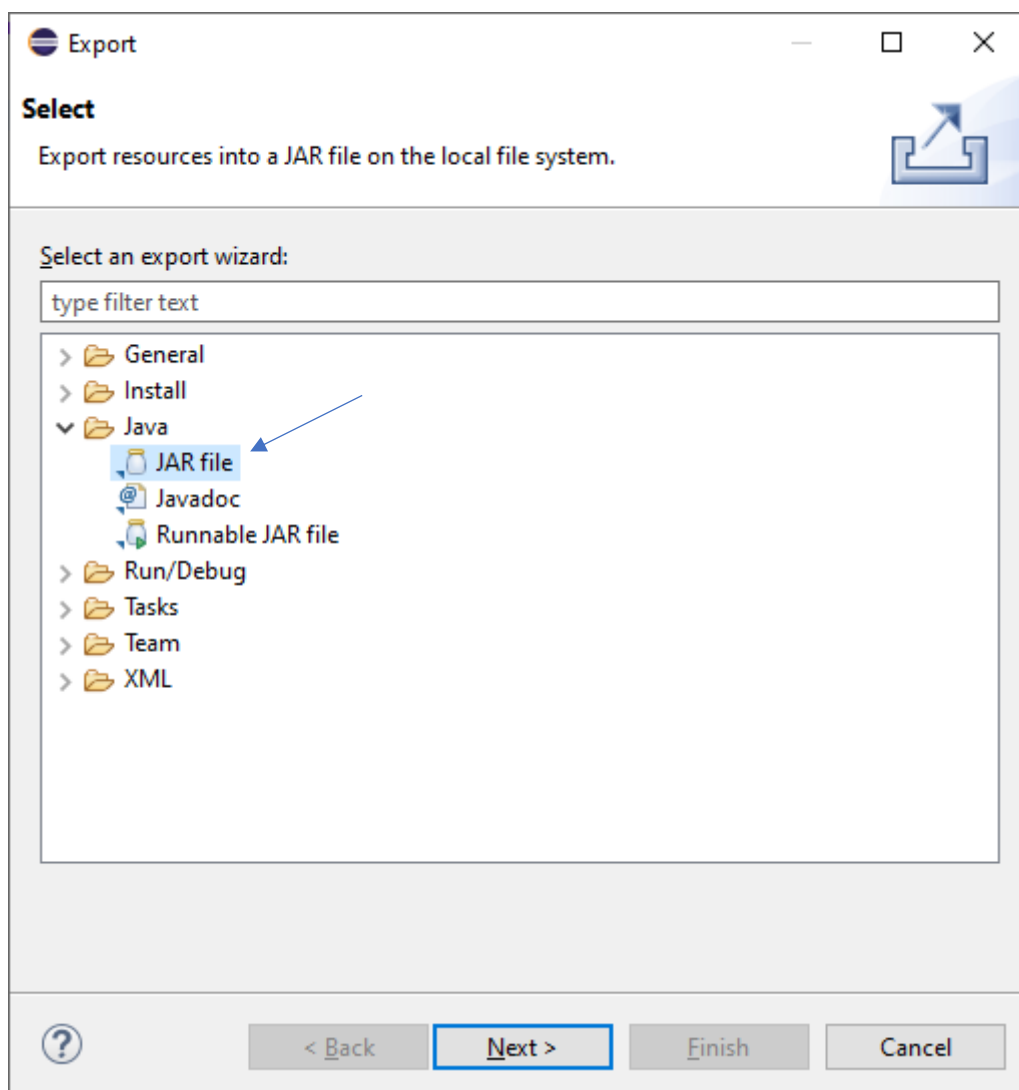


## Despliegue Aplicaciones. Archivos JAR III. Firmar JAR II. (Vídeo 139)

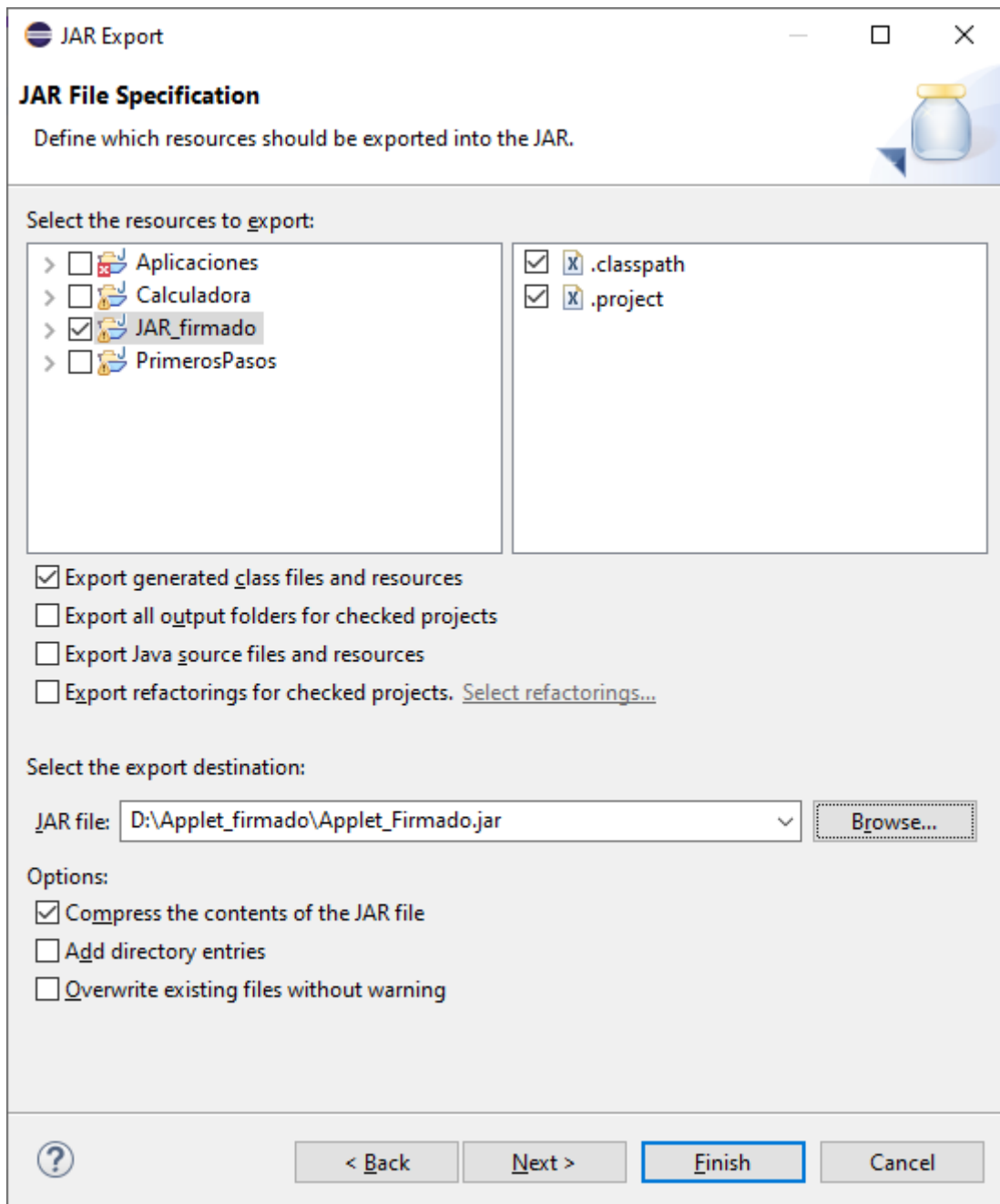
Seleccionamos el proyecto.



Del menú File seleccionamos Export...

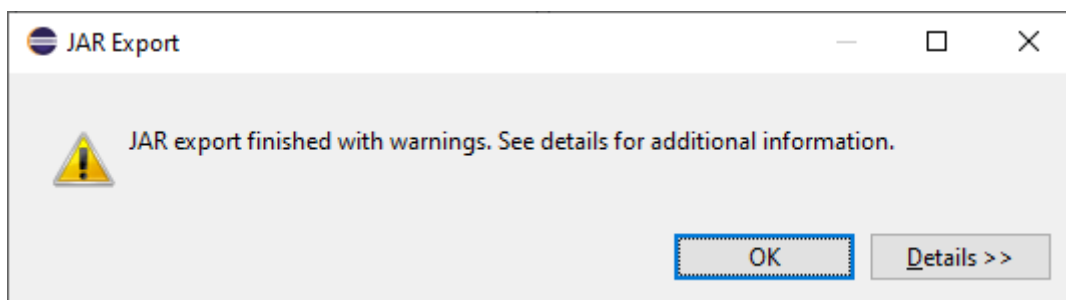


Seleccionamos JAR file seguido del botón Next.



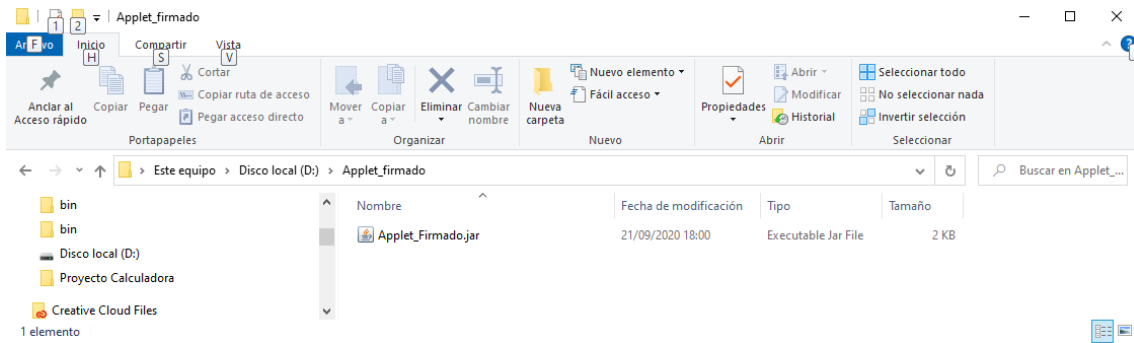
Seleccionamos el proyecto que queremos empaquetar y en el apartado Jar file: donde lo queremos guardar y nombre del archivo jar.

Por último el botón finalizar.

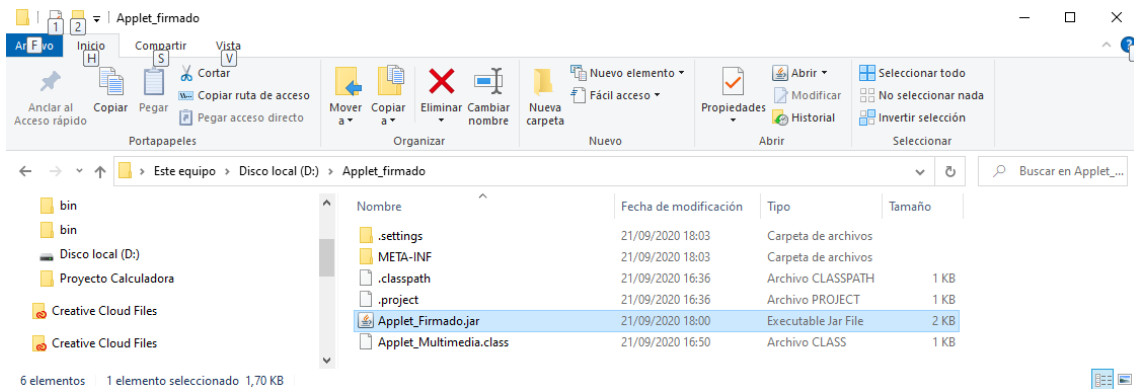


Contestamos con OK.

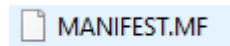
Vamos a comprobar si tenemos el archivo jar.



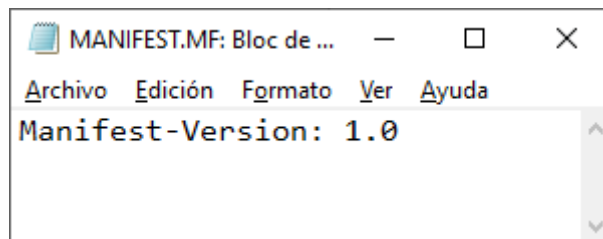
Si lo descomprimes, verás los archivos y directorios que contiene.



Nos vamos a la carpeta META-INF, encontraremos el siguiente archivo.



Si lo abrimos con el bloc de notas veremos la siguiente información.

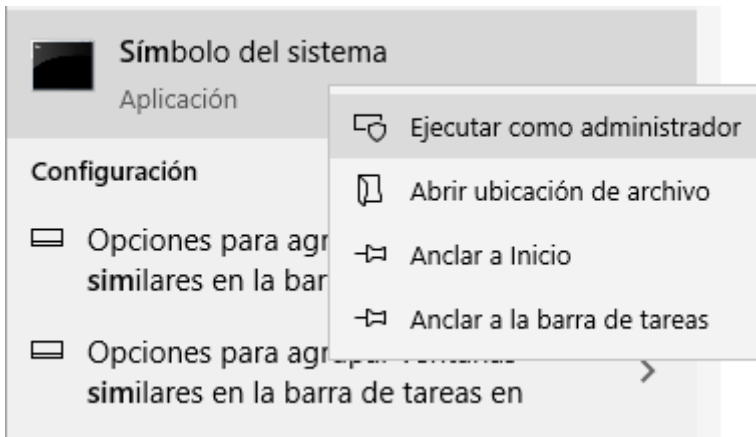


A continuación elimino todos los archivos y dejo el jar.

Ahora vamos a firmar el archivo jar.

Primero tenemos que generar la clave con la herramienta keytool y luego firmaremos el archivo con jarsigner.

Estas operaciones las realizaremos desde la consola, lo tenemos que ejecutar como administrador.



```
Administrador: Símbolo del sistema
Microsoft Windows [Versión 10.0.19041.508]
(c) 2020 Microsoft Corporation. Todos los derechos reservados.

C:\WINDOWS\system32>cd ..
C:\Windows>cd ..
C:\>cd Program files
C:\Program Files>cd Java
C:\Program Files\Java>cd jdk1.8.0_261
C:\Program Files\Java\jdk1.8.0_261>cd bin
C:\Program Files\Java\jdk1.8.0_261\bin>
```

```
Administrador: Símbolo del sistema - keytool -genkey -alias pevedugo -validity 150 -v
C:\Program Files\Java\jdk1.8.0_261\bin>keytool -genkey -alias pevedugo -validity 150 -v
Introduzca la contraseña del almacén de claves:
Volver a escribir la contraseña nueva:
¿Cuáles son su nombre y su apellido?
[Unknown]: Pere Manel
¿Cuál es el nombre de su unidad de organización?
[Unknown]: Dto. de Formación
¿Cuál es el nombre de su organización?
[Unknown]: INFORMENAT
¿Cuál es el nombre de su ciudad o localidad?
[Unknown]: Barcelona
¿Cuál es el nombre de su estado o provincia?
[Unknown]: Barcelona
¿Cuál es el código de país de dos letras de la unidad?
[Unknown]: ES
¿Es correcto CN=Pere Manel, OU=Dto. de Formaci#n, O=INFORMENAT, L=Barcelona, ST=Barcelona, C=ES?
[no]:
```

Nos pregunta si son correctos, le diremos que si.

```
Administrador: Símbolo del sistema - keytool -genkey -alias pevedugo -validity 150 -v
Introduzca la contraseña del almacén de claves:
Volver a escribir la contraseña nueva:
¿Cuáles son su nombre y su apellido?
[Unknown]: Pere Manel
¿Cuál es el nombre de su unidad de organización?
[Unknown]: Dto. de Formación
¿Cuál es el nombre de su organización?
[Unknown]: INFORMENAT
¿Cuál es el nombre de su ciudad o localidad?
[Unknown]: Barcelona
¿Cuál es el nombre de su estado o provincia?
[Unknown]: Barcelona
¿Cuál es el código de país de dos letras de la unidad?
[Unknown]: ES
¿Es correcto CN=Pere Manel, OU=Dto. de Formaciøn, O=INFORMENAT, L=Barcelona, ST=Barcelona, C=ES?
[no]: si

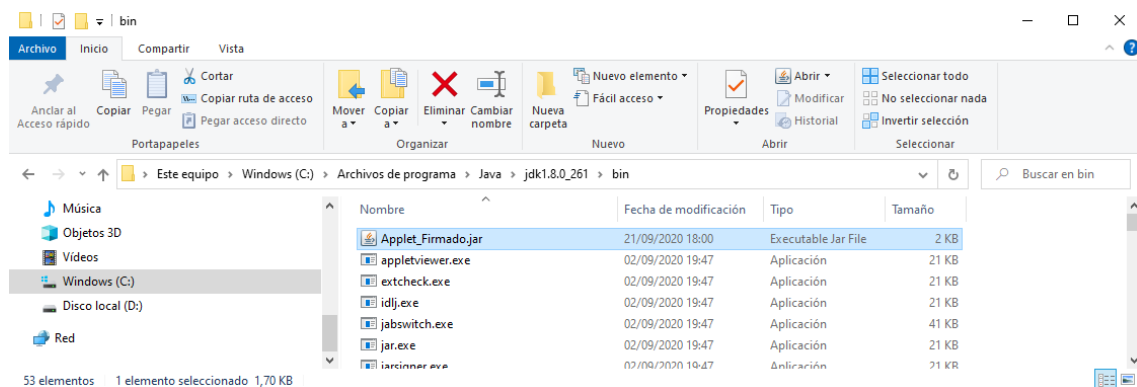
Generando par de claves DSA de 2.048 bits para certificado autofirmado (SHA256withDSA) con una
validez de 150 días
para: CN=Pere Manel, OU=Dto. de Formaciøn, O=INFORMENAT, L=Barcelona, ST=Barcelona, C=
ES
Introduzca la contraseña de clave para <pevedugo>
(INTRO si es la misma contraseña que la del almacén de claves):
```

Me pregunta que vuelva a introducir la contraseña dos veces.

```
Administrador: Símbolo del sistema
[no]: si

Generando par de claves DSA de 2.048 bits para certificado autofirmado (SHA256withDSA) con una
validez de 150 días
para: CN=Pere Manel, OU=Dto. de Formaciøn, O=INFORMENAT, L=Barcelona, ST=Barcelona, C=
ES
```

El archivo Applet\_Firmado.jar lo movemos al directorio donde están los programas.



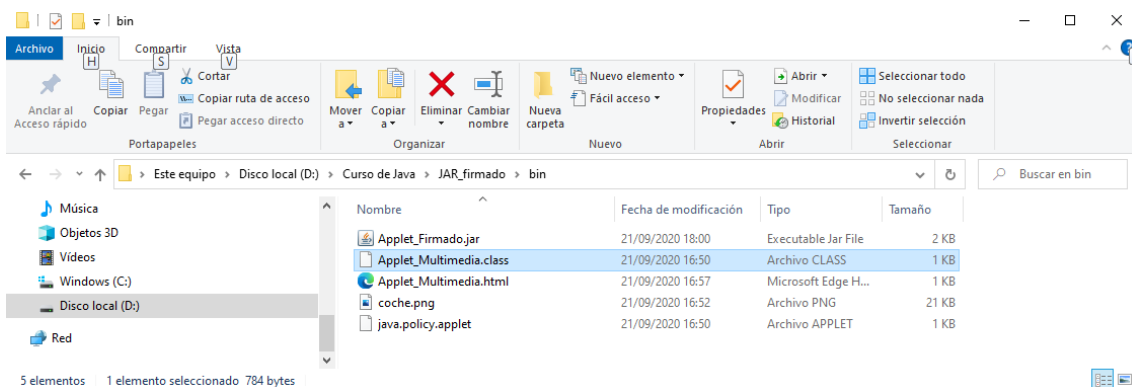
```
Administrador: Símbolo del sistema
C:\Program Files\Java\jdk1.8.0_261\bin>jarsigner.exe Applet_Firmado.jar peverdugo -verbose
Enter Passphrase for keystore:
updating: META-INF/MANIFEST.MF
  adding: META-INF/PEVERDUG.SF
  adding: META-INF/PEVERDUG.DSA
signing: Applet_Multimedia.class
signing: .classpath
signing: .project
signing: .settings/org.eclipse.jdt.core.prefs
>>> Signer
X.509, CN=Pere Manel, OU=Formacion, O=Informenat, L=Sentmenat, ST=Barcelona, C=ES
[trusted certificate]

jar signed.

Warning:
The signer's certificate is self-signed.

C:\Program Files\Java\jdk1.8.0_261\bin>
```

Ahora el archivo lo movemos a la carpeta que esta anteriormente.



Podemos eliminar Applet\_Multimedia.class. Ahora vamos a modificar el fichero html.

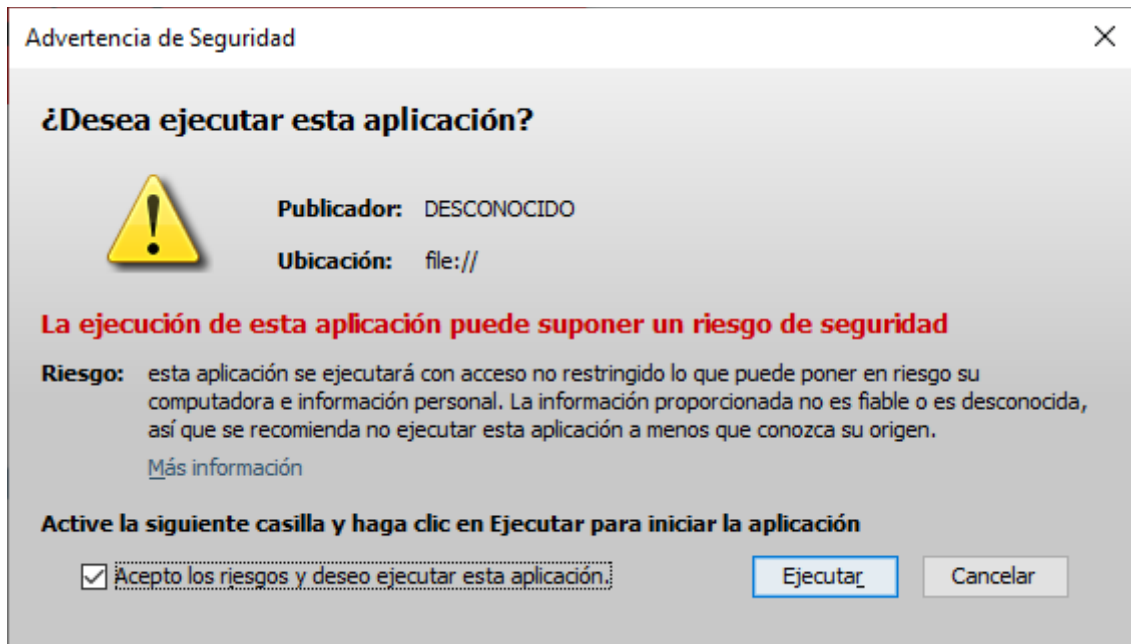
```
Applet_Multimedia.html: Bloc de notas
Archivo Edición Formato Ver Ayuda
<html>

<style>
#miapplet{
margin:auto;
position:relative;
width: 350px;
}
h1{
text-align:center;
}
</style>
<body bgcolor="red">
<h1>Calculadora Applet</h1>
<div id="miapplet">
<applet code="Applet_Multimedia.class" archive="Applet_Firmado.jar" width="350" height="500"
>
</applet>
</div>
</body>
</html>
```



Guardamos los cambios.

Aparece esta ventana que nos pide permiso para ejecutar.



Aceptamos y ejecutamos.

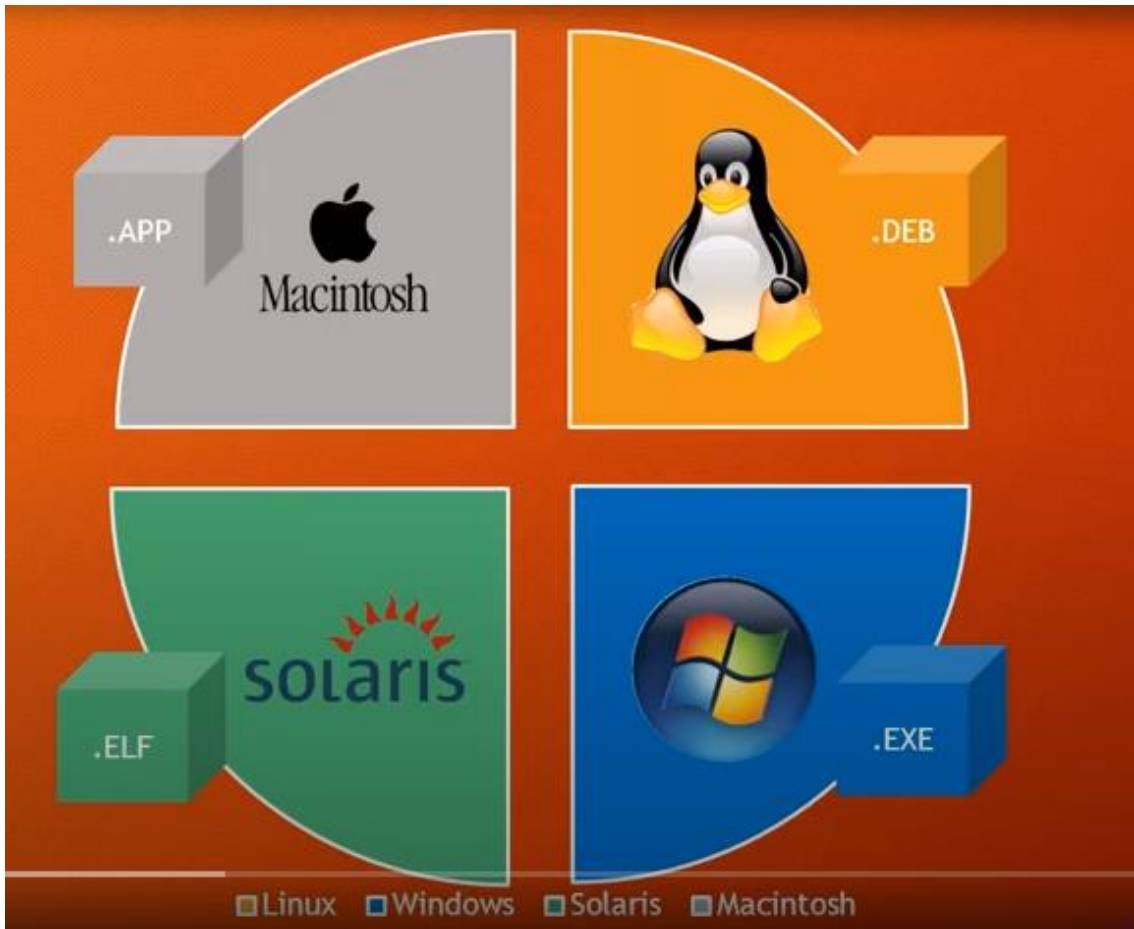
Este será el resultado en el navegador.





The image shows the cover of a course titled "CURSO JAVA 139". The background is a dark orange-red gradient with a faint world map. At the top left is a small Java logo icon. The main title "CURSO JAVA" is in large white letters, with "139" in a yellow box to its right. Below the title, the text "APPLETS Y APLICACIONES" is followed by "Archivos JAR III" and "Firmar JAR II". At the bottom left is the Eclipse logo, and at the bottom right is the Java logo with a coffee cup icon.

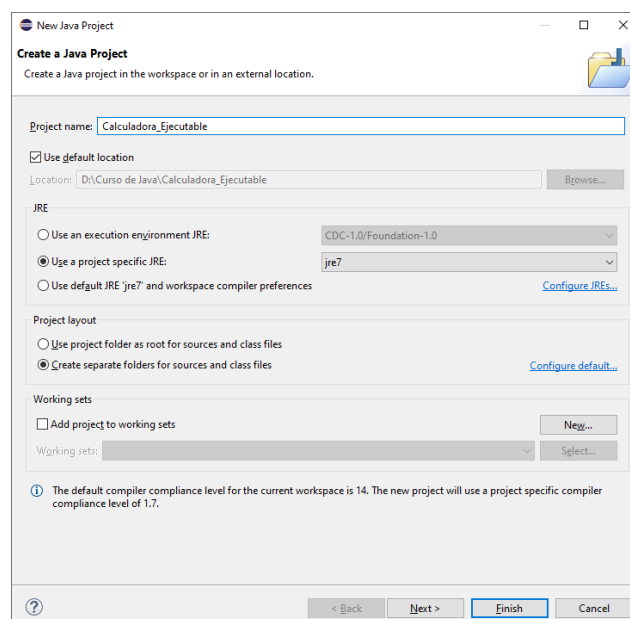
## Despliegue Aplicaciones. Archivos JAR IV. Ejecutables. (Vídeo 140) Ejecutables JAR.



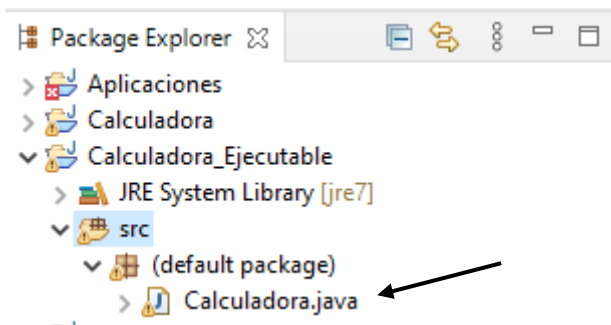
Vamos a trabajar con la calculadora.

Cargaremos la calculadora que hicimos en un principio, la que no es Jar.

Lo primero que vamos a realizar es crear un proyecto nuevo llamado Calculadora\_Ejecutable.



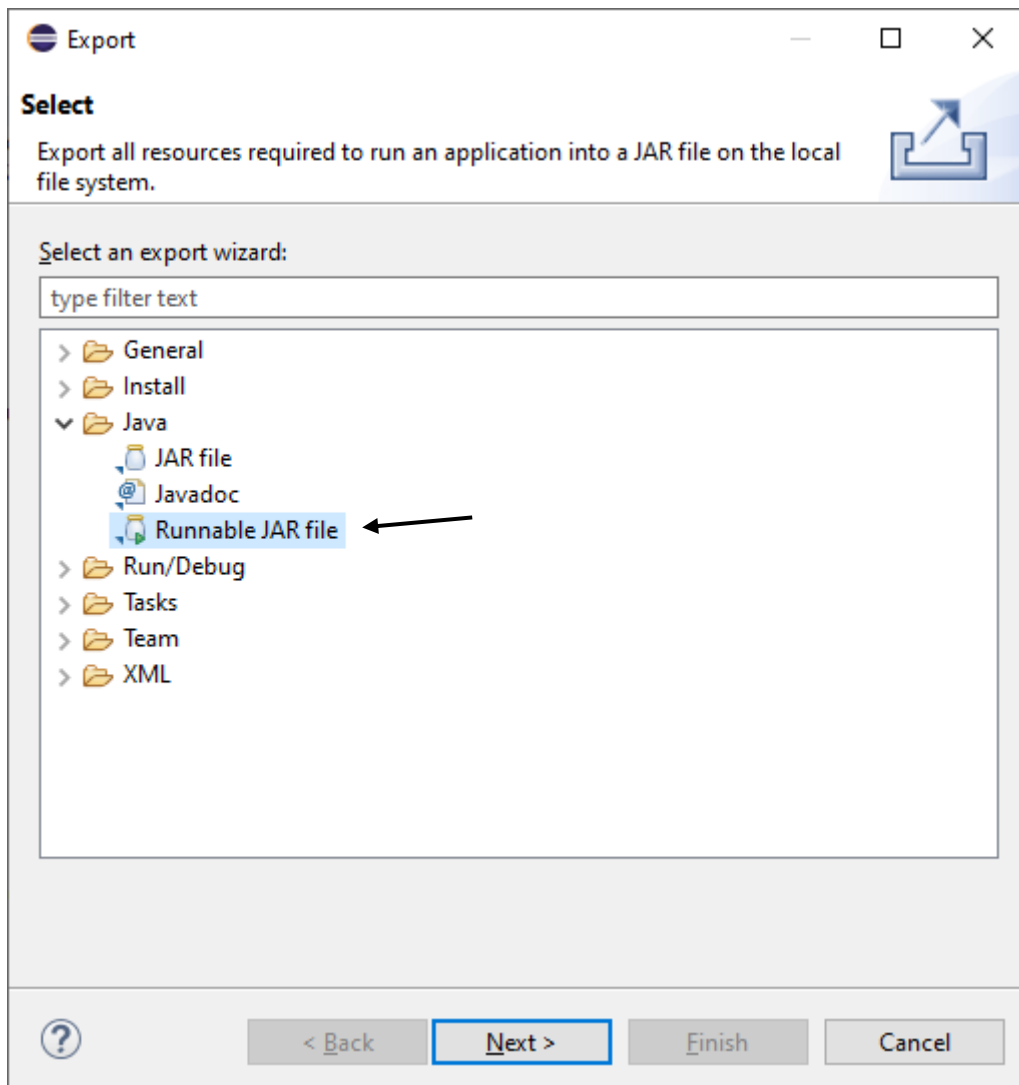
Copiamos la calculadora y la pegamos en la carpeta src del nuevo proyecto.



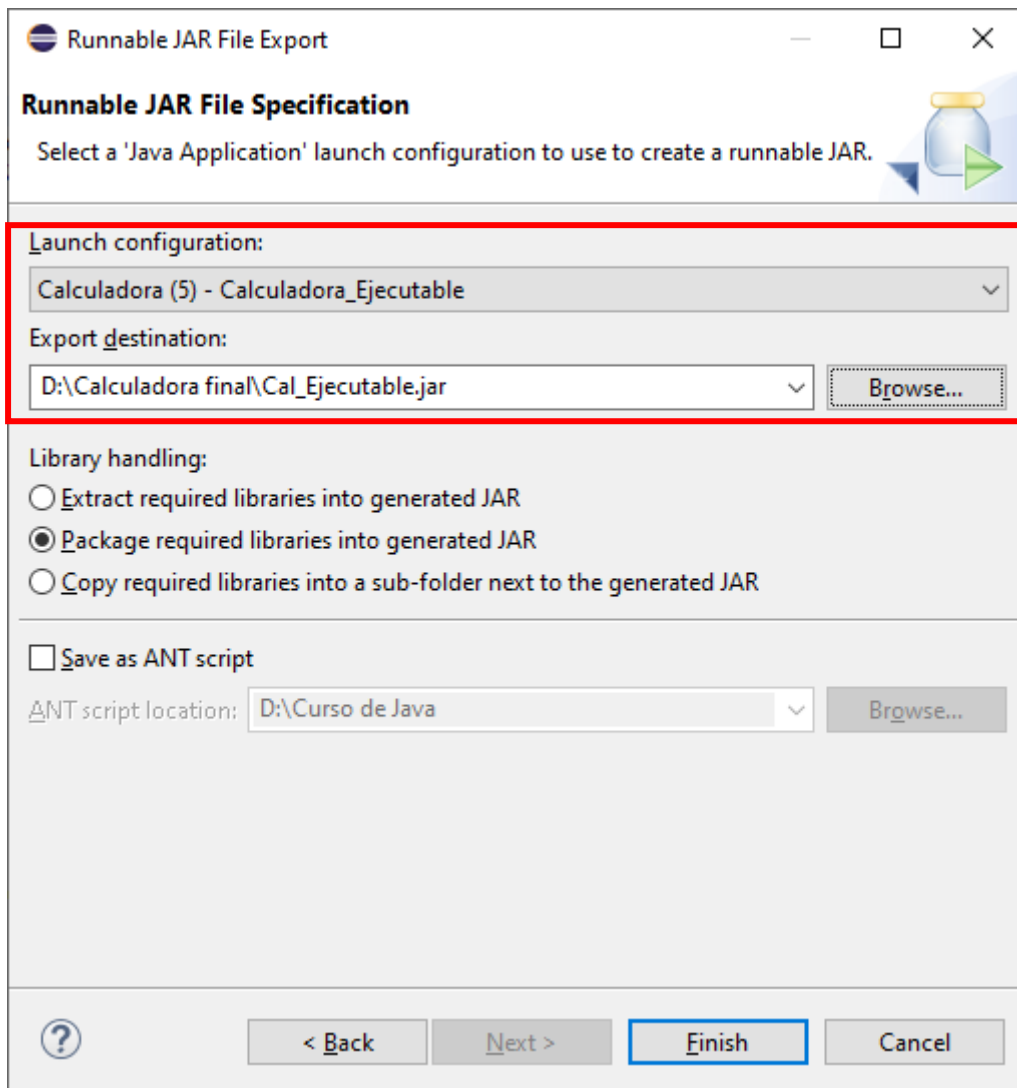
Si algún comentario diera error lo eliminamos.

Le damos al play y vemos que funciona correctamente.

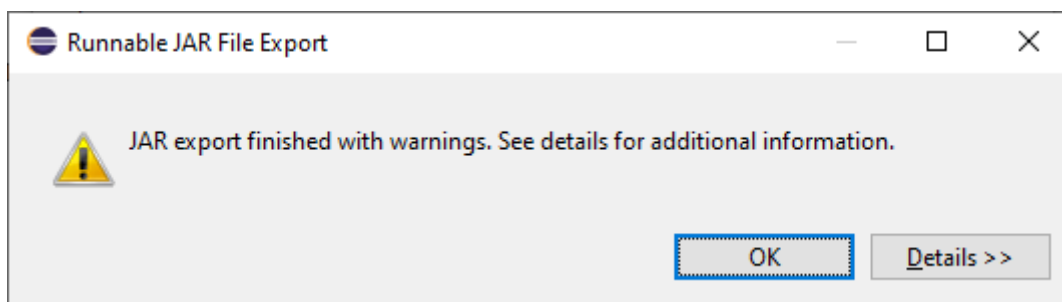
Ahora seleccionamos el menú File y de este Export...



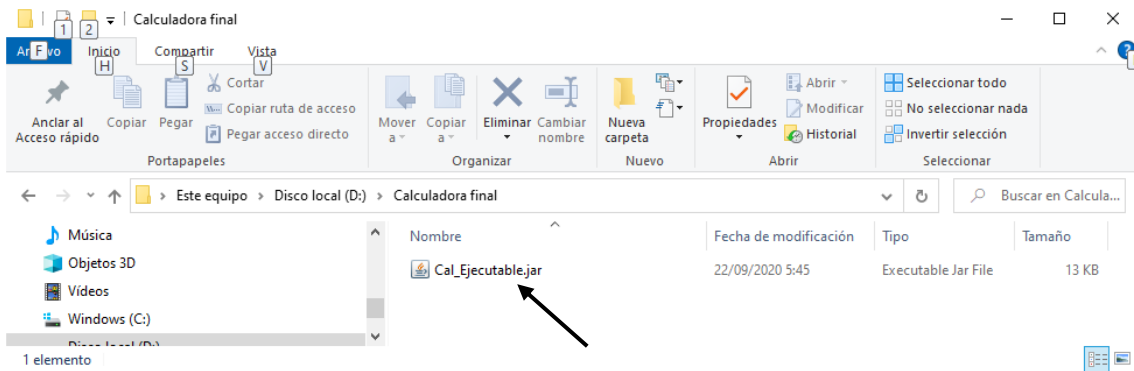
Seleccionamos Runnable JAR file el botón Next.



Buscamos la calculadora que guardamos en un nuevo proyecto llamado Calculadora\_Ejecutable. Y le decimos donde tiene que guardarse y con qué nombre. Seguido del botón Finish.

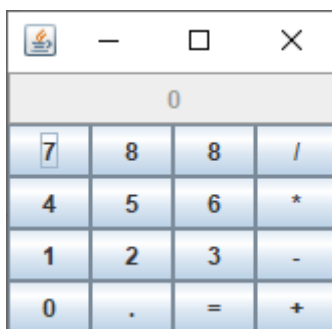


Nos dice que la operación ha tenido éxito pero hay algunos mensaje. Le damos a OK.



Comprobamos que ya lo tenemos.

Si hacemos doble clic sobre él, este ya se ejecuta.



La aplicación la podemos distribuir ya que puede funcionar en todos los sistemas operativos si tienen instalada la máquina virtual de java.

A la hora de empaquetar un proyecto los recursos externos, archivos que adjuntan al proyecto como puede ser una imagen no se van a empaquetar.

```

class LaminaRecursos extends JPanel{
    public LaminaRecursos(){
        setLayout(new BorderLayout());
        zonaTexto=new JTextArea();
        add(new JScrollPane(zonaTexto),BorderLayout.CENTER);
        URL ruta_boton=LaminaRecursos.class.getResource("Verde.gif");
        JButton boton_acerca_de=new JButton("Dale!", new ImageIcon(ruta_boton));
        boton_acerca_de.addActionListener(new Accion_Boton());
        add(boton_acerca_de, BorderLayout.SOUTH);
    }
}

```

Con este código se pueden almacenar recursos externos al empaquetar un JAR.

Lo ejecutamos para que haga el class y lo empaquetamos de nuevo.

Si lo ejecutamos no ocurre nada.

Abrimos el Jar con WinRAR y en la ventana de WinRAR arrastramos el archivo, en este caso la imagen.

Aceptamos y cerramos WinRAR.

Vamos donde tenemos el ejecutable le damos doble clic y este funciona.



## Contenido

Aplicaciones gráficas. Barra de herramientas I (Vídeo 111) .....	1
Aplicaciones gráficas. Barra de herramientas II. (Vídeo 112) .....	4
Aplicaciones gráficas. Barras de herramientas III. (Vídeo 113).....	8
Aplicaciones gráficas. Barras de herramientas IV. (Vídeo 114) .....	11
Aplicaciones gráficas. Disposiciones avanzadas I. Box. (Vídeo 115) .....	17
Aplicaciones gráficas. Disposiciones avanzadas II. Spring. (Vídeo 116).....	20
Aplicaciones gráficas. Disposiciones avanzadas III. Spring II (Video 117). .....	23
Aplicaciones gráficas. Disposiciones avanzadas IV. Disposiciones libres I. (Vídeo 118) .....	26
Aplicaciones gráficas. Disposiciones avanzadas V. Disposiciones libres II. (Vídeo 119) .....	30
Aplicaciones gráficas. Disposiciones avanzadas VI. Disposiciones libres III. (Vídeo 120) .....	35
Aplicaciones gráficas. Ventanas emergentes. Cuadros de diálogo I (Vídeo 121) .....	39
Aplicaciones gráficas. Ventanas emergentes. Cuadros diálogo II. (Vídeo 122) .....	43
Ejercicio Práctico I (Vídeo 123).....	45
Ejercicio Práctico II. (Vídeo 124).....	48
Ejercicio Práctico III. (Vídeo 125).....	50
Ejercicio Práctico IV. (Vídeo 126) .....	56
Ejercicio Práctico V. (Vídeo 127) .....	59
Ejercicio Práctico VI. (Vídeo 128) .....	61
Ejercicio Práctico VII. (Vídeo 129) .....	64
Ejercicio Práctico VIII. (Vídeo 130) .....	67
Ejercicio Práctico IX. (Vídeo 131).....	71
Despliegue Aplicaciones. Applets I. (Vídeo 132) .....	79
Despliegue Aplicaciones. Applet II. (Vídeo 133).....	92
Despliegue Aplicaciones. Applets III. Ventanas emergentes. (Vídeo 134).....	97
Despliegue Aplicaciones. Applets IV. Paso de parámetros. (Vídeo 135).....	100
Despliegue Aplicaciones. Applets V. (Vídeo 136).....	110
Despliegue Aplicaciones. Archivos JAR I. (Vídeo 137).....	115
Despliegue Aplicaciones. Archivos JAR II. Firmar JAR I (Vídeo 138).....	124
Despliegue Aplicaciones. Archivos JAR III. Firmar JAR II. (Vídeo 139).....	130
Despliegue Aplicaciones. Archivos JAR IV. Ejecutables. (Vídeo 140) .....	138