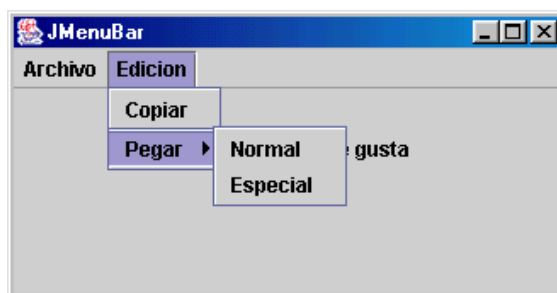


## Clase 13

### JMenuBar

La clase **JMenuBar** permite incluir barras de menú en una ventana. Las barras de menú actúan como contenedores de menús. Se apoya en las clases **JMenu** y **JMenuItem**. Cuando se selecciona un elemento de un menú se genera un evento del tipo **ActionEvent**.



### Ejemplo12.java

```
public class Ejemplo12
{
    public static void main ( String args [ ] )
    {
        Ventana12 miVentana;

        miVentana = new Ventana12 ( );
    }
}
```

### Ventana12.java

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

public class Ventana12 extends JFrame implements ActionListener
{
    private Container    miContenedor;
    private JMenuBar     barraMenuPrincipal;
    private JMenu        subMenuArchivo;
    private JMenu        subMenuEdicion;
    private JMenuItem    abrir;
    private JMenuItem    salir;
    private JMenuItem    copiar;
    private JMenu        pegar;
    private JMenuItem    normal;
    private JMenuItem    especial;
    private JSeparator    separador;
    private JSeparator    separador1;
    private JLabel        titulo;
    private JLabel        salida;

    public Ventana12 ( )
    {
        titulo = new JLabel ( );
    }
}
```

```

barraMenuPrincipal = new JMenuBar ( );
subMenuArchivo     = new JMenu     ( );
subMenuEdicion     = new JMenu     ( );
abrir              = new JMenuItem ( );
salir              = new JMenuItem ( );
copiar             = new JMenuItem ( );
pegar              = new JMenu      ( );
normal             = new JMenuItem ( );
especial           = new JMenuItem ( );
separador          = new JSeparator ( );
separador1         = new JSeparator ( );
salida             = new JLabel     ( );

titulo.setHorizontalAlignment ( SwingConstants.CENTER );
titulo.setText          ( "Elija lo que mas le gusta" );
titulo.setBounds       ( 10, 20, 310, 30 );

subMenuArchivo.setText ( "Archivo" );
subMenuEdicion.setText ( "Edicion" );
abrir.setText          ( "Abrir" );
salir.setText          ( "Salir" );
copiar.setText         ( "Copiar" );
pegar.setText          ( "Pegar" );
normal.setText         ( "Normal" );
especial.setText       ( "Especial" );

barraMenuPrincipal.add ( subMenuArchivo );
barraMenuPrincipal.add ( subMenuEdicion );
pegar.add              ( normal );
pegar.add              ( especial );
subMenuArchivo.add     ( abrir );
subMenuArchivo.add     ( separador );
subMenuArchivo.add     ( salir );
subMenuEdicion.add     ( copiar );
subMenuEdicion.add     ( separador1 );
subMenuEdicion.add     ( pegar );

salida.setBounds       ( 10, 90, 310, 16 );

miContenedor = getContentPane ( );
miContenedor.setLayout ( null );

miContenedor.add ( titulo );
miContenedor.add ( barraMenuPrincipal );
miContenedor.add ( salida );

setTitle      ( "JMenuBar" );
setSize      ( 330, 170 );

setJMenuBar ( barraMenuPrincipal );

show ( );

abrir.addActionListener ( this );
salir.addActionListener ( this );
copiar.addActionListener ( this );
pegar.addActionListener ( this );
normal.addActionListener ( this );
especial.addActionListener ( this );

```

```

    }

    public void actionPerformed((ActionEvent evento) )
    {
        if ( evento.getSource ( ) == salir )
        {
            System.exit ( 0 );
        }

        salida.setText ( "Seleccionó: " + evento.getActionCommand ( ) );
    }
}

```

## JFileChooser

Provee un mecanismo sencillo para que el usuario escoja un archivo. Puede obtenerse una cadena con el nombre del archivo seleccionado o la ruta completa.

## FileFilter

Es una interface que se utiliza para limitar la lista de los archivos que aparecen como disponibles cuando se utiliza el componente JFileChooser.



## Ejemplo13.java

```

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

public class Ejemplo13
{
    public static void main( String args [ ] )
    {
        JFileChooser selector;
        int valor;
        String extensiones [ ];
        Filtro filtro;
    }
}

```

```

extensiones = new String [ 2 ];

extensiones [ 0 ] = "jpg";
extensiones [ 1 ] = "gif";

filtro      = new Filtro ( extensiones, "Imágenes JPG & GIF" );

selector = new JFileChooser      ( "."      );
selector.setFileFilter           ( filtro );
selector.setAcceptAllFileFilterUsed ( true  );

// Si desea el cuadro de abrir archivo:
// valor = selector.showOpenDialog ( null );
valor    = selector.showSaveDialog ( null );

if( valor == JFileChooser.APPROVE_OPTION )
{
    JOptionPane.showMessageDialog
        ( null, selector.getSelectedFile ( ).getName ( ) + "\n" +
          selector.getSelectedFile ( ).getPath ( ) );
}
else
{
    JOptionPane.showMessageDialog
        ( null, "No seleccionó ningún archivo" );
}
}
}

```

## Filtro.java

```

import java.io.File;
import javax.swing.filechooser.*;

public class Filtro extends FileFilter
{
    String extension [ ];
    String descripcion;

    public Filtro ( String extension [ ], String descripcion )
    {
        this.extension    = extension;
        this.descripcion   = descripcion;
    }

    public boolean accept ( File archivo )
    {
        int i;

        if ( archivo != null )
        {
            if ( archivo.isDirectory ( ) )
            {
                return true;
            }

            for ( i = 0; i < extension.length; i++ )

```

```

    {
        if ( getExtension ( archivo ).equals ( extension [ i ] ) == true )
        {
            return true;
        }
    }

    return false;
}

public String getDescription ( )
{
    return descripcion;
}

private String getExtension ( File archivo )
{
    String nombreArchivo;
    int i;

    if ( archivo != null )
    {
        nombreArchivo = archivo.getName ( );

        i = nombreArchivo.lastIndexOf ( '.' );

        if ( i > 0 && i < nombreArchivo.length ( ) - 1 )
        {
            return nombreArchivo.substring ( i + 1 ).toLowerCase ( );
        }
    }

    return "";
}
}

```

## JColorChooser

Es un componente que provee un panel de control diseñado para permitir al usuario seleccionar un color en diferentes representaciones. El formato más común es el formato RGB.



## Ejemplo14.java

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

public class Ejemplo14
{
    public static void main ( String args [ ] )
    {
        Color color, colorNuevo;

        color = new Color ( 200, 200, 50 );

        JOptionPane.showMessageDialog ( null, "Color actual: " + color );

        colorNuevo = JColorChooser.showDialog ( null, "Selección de color", color );

        JOptionPane.showMessageDialog ( null, "Nuevo color: " + colorNuevo );
    }
}
```

## JTabbedPane

Es un componente que permite representar un grupo de carpetas con pestañas. Cada carpeta tiene un título y cuando el usuario selecciona una carpeta, se muestra su contenido. En este componente solo puede estar seleccionada una carpeta al mismo tiempo.



## Ejemplo15.java

```
public class Ejemplo15
{
    public static void main ( String args [ ] )
    {
        Ventana15 miVentana;
```

```

        miVentana = new Ventana15 ( );
    }
}

```

## Ventana15.java

```

import javax.swing.*;

public class Ventana15 extends JFrame
{
    public Ventana15 ( )
    {
        JTabbedPane fichas;
        PanelCiudades ciudades;
        PanelColores colores;
        PanelSabores sabores;

        fichas = new JTabbedPane ( );
        ciudades = new PanelCiudades ( );
        colores = new PanelColores ( );
        sabores = new PanelSabores ( );

        fichas.addTab ( "Ciudades", ciudades );
        fichas.addTab ( "Colores", colores );
        fichas.addTab ( "Sabores", sabores );

        getContentPane ( ).add ( fichas );

        setTitle      ( "JTabbedPane" );
        setSize       ( 330, 170 );
        show ( );
    }
}

```

## PanelCiudades.java

```

import javax.swing.*;

public class PanelCiudades extends JPanel
{
    public PanelCiudades ( )
    {
        JButton b1;
        JButton b2;
        JButton b3;
        JButton b4;

        b1 = new JButton ( "Nueva York" );
        b2 = new JButton ( "Londres" );
        b3 = new JButton ( "Hong Kong" );
        b4 = new JButton ( "Tokio" );

        add ( b1 );
        add ( b2 );
        add ( b3 );
        add ( b4 );
    }
}

```

```
}
```

## PanelColores.java

```
import javax.swing.*;

class PanelColores extends JPanel
{
    public PanelColores ( )
    {
        JCheckBox opcion1;
        JCheckBox opcion2;
        JCheckBox opcion3;
        JCheckBox opcion4;
        JCheckBox opcion5;

        opcion1 = new JCheckBox ( "Rojo"      );
        opcion2 = new JCheckBox ( "Amarillo"  );
        opcion3 = new JCheckBox ( "Azul"      );
        opcion4 = new JCheckBox ( "Verde"     );
        opcion5 = new JCheckBox ( "Negro"     );

        add ( opcion1 );
        add ( opcion2 );
        add ( opcion3 );
        add ( opcion4 );
        add ( opcion5 );
    }
}
```

## PanelSabores.java

```
import javax.swing.*;

public class PanelSabores extends JPanel
{
    public PanelSabores ( )
    {
        JComboBox lista;

        lista = new JComboBox ( );

        lista.addItem ( "Vanilla" );
        lista.addItem ( "Chocolate" );
        lista.addItem ( "Fresa" );
        lista.addItem ( "Mandarina" );
        lista.addItem ( "Limón" );

        add ( lista );
    }
}
```

## Taller

En el ejercicio del componente JTabbedPane hace falta el procesamiento de los eventos, en el cual se debe imprimir una etiqueta cuyo texto corresponda con el elemento seleccionado. Complete el ejercicio.