Universidad Tecnológica de Pereira

Mision TIC 2022

El futuro digital es de todos

MinTIC

Unidad 5

15 - Tablas y Árboles

Hechos QUE CONECTAN

Universidad Tecnológica de Pereira

Mision TIC 2022

El futuro digital es de todos

MinTIC

JTree - Árboles

https://docs.oracle.com/javase/tutorial/uiswing/components/tree.html

Hechos QUE CONECTAN

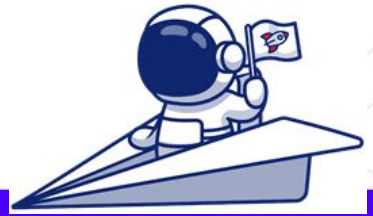
Árboles (JTree)

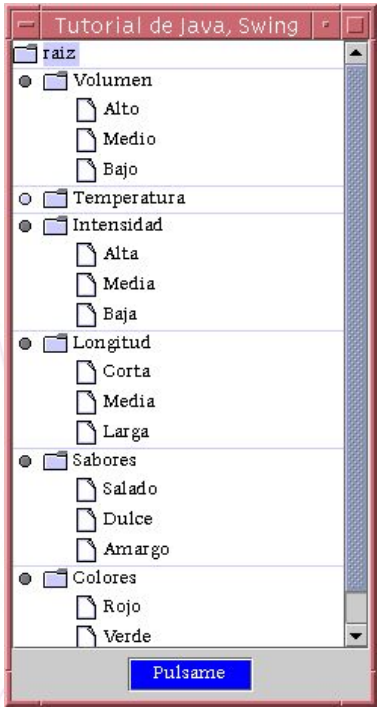
Utilizar en Swing un árbol, como los que se despliegan en muchas de las ventanas de los sistemas operativos al uso, es tan simple como escribir:

```
add(new JTree(new Object[]{ "este","ese","aquel" }));
```

Esto crea un arbolito muy primitivo; sin embargo, el API de Swing para árboles es inmenso, quizá sea uno de los más grandes.

En principio parece que se puede hacer cualquier cosa con árboles, pero lo cierto es que si se van a realizar tareas con un cierto grado de complejidad, es necesario un poco de investigación y experimentación, antes de lograr los resultados deseados.





Ejemplo

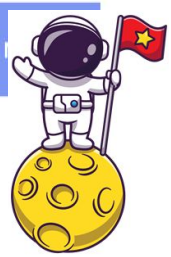
```
import javax.swing.tree.DefaultMutableTreeNode;

public class Rama {
  DefaultMutableTreeNode r;
  public Rama(String datos[]) {
    r = new DefaultMutableTreeNode(datos[0]);
    for( int i=1; i < datos.length; i++ ) {
      r.add( new DefaultMutableTreeNode(datos[i]));
    }
  }

  public DefaultMutableTreeNode node() {
    return( r );
  }
}
```

```
import java.awt.BorderLayout;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;
import javax.swing.JFrame;

public class Main {
  public static void main(String[] args) {
    JFrame frame = new JFrame("Tutorial de Java,
Swing");

frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.getContentPane().add(new MiArbol(),
                                BorderLayout.CENTER);
    frame.setSize(400, 500);
    frame.setLocationRelativeTo(null);
    frame.setVisible(true);
  }
}
```

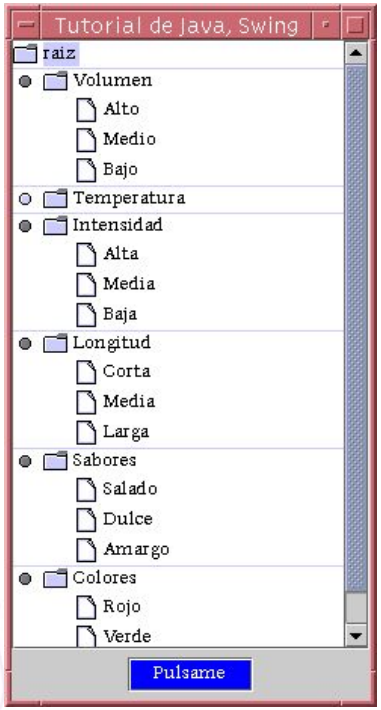
Ejemplo

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

public class MiArbol extends JPanel {
  String datos[][] = {
          { "Colores", "Rojo", "Verde", "Azul" },
          { "Sabores", "Salado", "Dulce", "Amargo" },
          { "Longitud", "Corta", "Media", "Larga" },
          { "Intensidad", "Alta", "Media", "Baja" },
          { "Temperatura", "Alta", "Media", "Baja" },
          { "Volumen", "Alto", "Medio", "Bajo" }
      };

  static int i = 0;
  DefaultMutableTreeNode raiz, rama, seleccion;
  JTree arbol;
  DefaultTreeModel modelo;

  public MiArbol() {
    setLayout(new BorderLayout());
    raiz = new DefaultMutableTreeNode("raiz");
    arbol = new JTree(raiz);

    add(new JScrollPane(arbol), BorderLayout.CENTER);

    modelo = (DefaultTreeModel) arbol.getModel();
    ...
```

```
    ...
    JButton botonPrueba = new JButton("Cargar información");
    botonPrueba.addActionListener(new ActionListener() {
      public void actionPerformed(ActionEvent evt) {
        if (i < datos.length) {
          rama = new Rama(datos[i++]).node();
          seleccion = (DefaultMutableTreeNode)
arbol.getLastSelectedPathComponent();
          if (seleccion == null) {
            seleccion = raiz;
          }

          modelo.insertNodeInto(rama, seleccion, 0);
        } else {
          JOptionPane.showMessageDialog(MiArbol.this,
                        "No hay más datos para cargar");
        }
      }
    });

    // Cambio del color del botón
    botonPrueba.setBackground(Color.blue);
    botonPrueba.setForeground(Color.white);
    // Se crea un panel para contener al botón
    JPanel panel = new JPanel();
    panel.add(botonPrueba);
    add(panel, BorderLayout.SOUTH);
  }
}
```



Explicación del Ejemplo



En la aplicación, la primera clase, Rama, coge un array de String y genera uno de los nodos el árbol, con la primera cadena como raíz y el resto como sus hojas. Posteriormente, se pueden hacer llamadas al método node() para generar la raíz de esta rama.

La clase JTree contiene un array de String de dos dimensiones, a partir del cual salen las ramas, y un contador para moverse a través del array. Los objetos DefaultMutableTreeNode contienen los nodos, pero la representación física sobre la pantalla está controlada por la clase JTree y su modelo asociado, el DefaultTreeModel. Cuando el JTree se incorpora al Frame, es incrustado automáticamente en un JScrollPane, para proporcionar desplazamiento o scroll automático al árbol.

Universidad Tecnológica de Pereira

Mision TIC 2022

El futuro digital es de todos

MinTIC

JTable - Tablas

https://docs.oracle.com/javase/tutorial/uiswing/components/table.html

Hechos QUE CONECTAN

Tablas (JTable)

Al igual que los árboles, las tablas en Swing son importantes y poderosas. En principio, se crearon para constituir un interfaz ligado a bases de datos a través del Java Database Connectivity (JDBC), y así evitar la complejidad inherente al manejo de los datos, proporcionando mucha flexibilidad al programador.

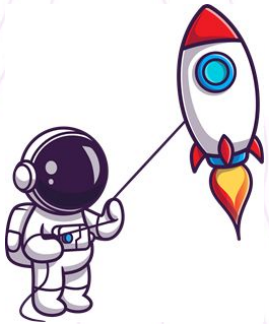
Hay suficientes características como para montar desde una hoja de cálculo básica hasta información para escribir un libro completo. Sin embargo, también es posible crear una JTable relativamente simple si entiende correctamente el funcionamiento.

La **JTable** controla cómo se presentan los datos, siendo el **TableModel** quien controla los datos en sí mismos. Para crear una **JTable** habrá pues que crear un **TableModel** antes, normalmente.

Se puede implementar, para ello, el interface TableModel, pero es mucho más simple heredar de la clase ayuda AbstractTableModel.

Ejemplo

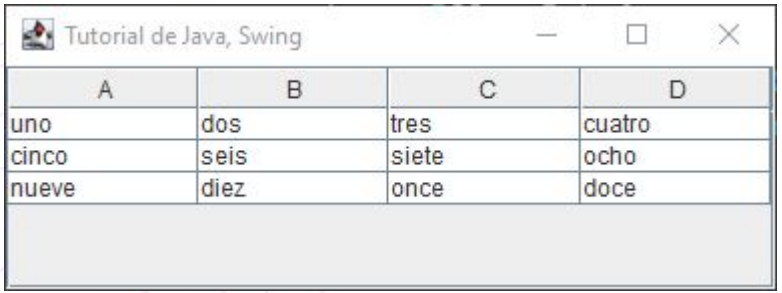
```
public class ModeloDatos extends AbstractTableModel {
  Object datos[][] = {
        { "uno", "dos", "tres", "cuatro" },
        { "cinco", "seis", "siete", "ocho" },
        { "nueve", "diez", "once", "doce" } };

  public ModeloDatos() {
    addTableModelListener(evt -> {
      for (int i = 0; i < datos.length; i++) {
        for (int j = 0; j < datos[0].length; j++)
          System.out.print(datos[i][j] + " ");
        System.out.println();
      }
    });
  }

  @Override
  public int getRowCount() {
    return datos.length;
  }
  ...
```



```
  ...
  @Override
  public int getColumnCount() {
    return datos[0].length;
  }

  @Override
  public Object getValueAt(int rowIndex, int columnIndex) {
    return datos[rowIndex][columnIndex];
  }
  @Override
  public void setValueAt(Object aValue, int rowIndex, int
columnIndex) {
    datos[rowIndex][columnIndex] = aValue;
    fireTableDataChanged();
  }
  @Override
  public boolean isCellEditable(int rowIndex, int
columnIndex) {
    return true;
  }
}
```

Ejemplo

```
public class MiTabla extends JPanel {

  public MiTabla() {
    setLayout(new BorderLayout());
    JTable tabla = new JTable(new ModeloDatos());
    JScrollPane panel = new JScrollPane(tabla);
    add(panel, BorderLayout.CENTER);
  }

  public static void main(String[] args) {
    JFrame frame = new JFrame("Tutorial de Java, Swing");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.getContentPane().add(new MiTabla(), BorderLayout.CENTER);
    frame.setSize(400, 150);
    frame.setLocationRelativeTo(null);
    frame.setVisible(true);
  }
}
```

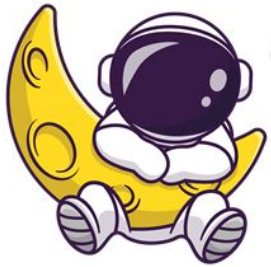


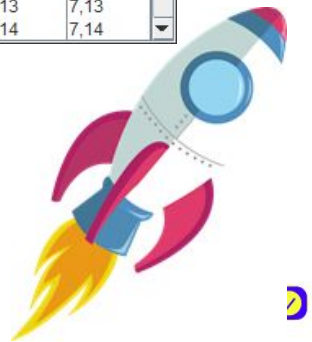
Tablas (JTable)

Las tablas pueden soportar un comportamiento más complejo. En el siguiente ejemplo, se utiliza un método para cargar un array de ocho columnas por un ciento de filas y, posteriormente, la tabla es configurada para que muestre solamente las líneas verticales y permita la selección simultánea de la fila y columna en que se encuentre la celda marcada.

Tutorial de Java, Swing

Col: 0	Col: 1	Col: 2	Col: 3	Col: 4	Col: 5	Col: 6	Col: 7
0,0	1,0	2,0	3,0	4,0	5,0	6,0	7,0
0,1	1,1	2,1	3,1	4,1	5,1	6,1	7,1
0,2	1,2	2,2	3,2	4,2	5,2	6,2	7,2
0,3	1,3	2,3	3,3	4,3	5,3	6,3	7,3
0,4	1,4	2,4	3,4	4,4	5,4	6,4	7,4
0,5	1,5	2,5	3,5	4,5	5,5	6,5	7,5
0,6	1,6	2,6	3,6	4,6	5,6	6,6	7,6
0,7	1,7	2,7	3,7	4,7	5,7	6,7	7,7
0,8	1,8	2,8	3,8	4,8	5,8	6,8	7,8
0,9	1,9	2,9	3,9	4,9	5,9	6,9	7,9
0,10	1,10	2,10	3,10	4,10	5,10	6,10	7,10
0,11	1,11	2,11	3,11	4,11	5,11	6,11	7,11
0,12	1,12	2,12	3,12	4,12	5,12	6,12	7,12
0,13	1,13	2,13	3,13	4,13	5,13	6,13	7,13
0,14	1,14	2,14	3,14	4,14	5,14	6,14	7,14





Ejemplo 2

```
public class MiTabla2 extends JPanel {
  private String titles[];
  private String data[][];

  public MiTabla2() {
    setLayout(new BorderLayout());
    generarDatos();
    var tabla = new JTable(data, titles);
    tabla.setShowHorizontalLines(false);
    tabla.setRowSelectionAllowed(true);
    tabla.setColumnSelectionAllowed(true);
    tabla.setSelectionForeground(Color.white);
    tabla.setSelectionBackground(Color.red);
    add(new JScrollPane(tabla), BorderLayout.CENTER);
  }

  public void generarDatos() {
    titles = new String[8];
     for (int i = 0; i < 8; i++) {
        titles[i] = "Col: " + i;
     }
     ...
```

```
    ...
    data = new String[100][8];
    for (int iY = 0; iY < 100; iY++) {
      for (int iX = 0; iX < 8; iX++) {
        data[iY][iX] = "" + iX + "," + iY;
      }
    }
  }

  public static void main(String args[]) {
    JFrame ventana = new JFrame("Tutorial de Java, Swing");
    ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    ventana.getContentPane().add(new MiTabla2(), BorderLayout.CENTER);
    ventana.setSize(500, 300);
    ventana.setLocationRelativeTo(null);
    ventana.setVisible(true);
  }
}
```

Para la próxima sesión...

- Revisar el código fuente de ejemplos (carpeta tabla y arbol) en el repositorio: https://github.com/cesardiaz-utp/MisionTIC2022-Ciclo2-Unidad5-IntroGUI
- Revisar el funcionamiento de la aplicación que se encuenta en https://github.com/cesardiaz-utp/MisionTIC2022-Ciclo2-Unidad5-Swing