

```
// Clase que representa una entidad de la base de datos que vamos a mostrar en nuestra tabla
//Tenemos que cambiar por el array de 2 dimensiones que tenemos
class Player{
    String firstName, LastName, Sport;
    int year;
    boolean vegetarian;

    public Player(String firstName, String LastName, String Sport, int year, boolean vegetarian) {
        this.firstName = firstName;
        this.LastName = LastName;
        this.Sport = Sport;
        this.year = year;
        this.vegetarian = vegetarian;
    }

    public String getFirstName() {
        return firstName;
    }

    public String getLastName() {
        return LastName;
    }

    public String getSport() {
        return Sport;
    }

    public int getYear() {
        return year;
    }

    public boolean isVegetarian() {
        return vegetarian;
    }
}
```

```

// Vista de la tabla.
public class TableSortDemo extends JPanel {
    private boolean DEBUG = false;
    public TableSortDemo() {
        super(new GridLayout(1,0));
        JTable table = new JTable(new MyTableModel());
        table.setPreferredScrollableViewportSize(new Dimension(500, 70));
        table.setFillsViewportHeight(true);
        table.setAutoCreateRowSorter(true);

        //ScrollPane que contiene a la tabla.
        JScrollPane scrollPane = new JScrollPane(table);

        //Añadimos el scrollpane a nuestro panel.
        add(scrollPane);
    }
    //modelo de la tabla
    class MyTableModel extends AbstractTableModel {
        private String[] columnNames = {"First Name",
                                         "Last Name",
                                         "Sport",
                                         "# of Years",
                                         "Vegetarian"};

        private Object[][] data = {
            {"Kathy", "Smith",
             "Snowboarding", new Integer(5), new Boolean(false)},
            {"John", "Doe",
             "Rowing", new Integer(3), new Boolean(true)},
            {"Sue", "Black",
             "Knitting", new Integer(2), new Boolean(false)},
            {"Jane", "White",
             "Speed reading", new Integer(20), new Boolean(true)},
            {"Joe", "Brown",
             "Pool", new Integer(10), new Boolean(false)}
        };
    };
    /* Tenemos que cambiar lo de arriba por esto.
    private Player[] data_players = {
        new Player("Kathy", "Smith",
                   "Snowboarding", new Integer(5), new Boolean(false)),
        new Player("John", "Doe",
                   "Rowing", new Integer(3), new Boolean(true)),
        new Player("Sue", "Black",
                   "Knitting", new Integer(2), new Boolean(false)),
        new Player("Jane", "White",
                   "Speed reading", new Integer(20), new Boolean(true)),
        new Player("Joe", "Brown",
                   "Pool", new Integer(10), new Boolean(false))
    };*/

```

```

private ArrayList<Player> al_players = new ArrayList( Arrays.asList( data_players) );

public int getColumnCount() {
    return columnNames.length;
}

public int getRowCount() {
    return data.length;
}

public String getColumnName(int col) {
    return columnNames[col];
}

public Object getValueAt(int row, int col) {
    return data[row][col];
}

//prueba a comentar este método para ver qué ocurre
public Class getColumnClass(int c) {
    return getValueAt(0, c).getClass();
}

//Implementa este método si tu tabla es editable
public boolean isCellEditable(int row, int col) {
    //Note that the data/cell address is constant,
    //no matter where the cell appears onscreen.
    return (col >= 2);
}

// Método que debemos implementar si los datos de la tabla pueden cambiar
public void setValueAt(Object value, int row, int col) {
    if (DEBUG) {
        System.out.println("Setting value at " + row + "," + col
            + " to " + value
            + " (an instance of "
            + value.getClass() + ")");
    }

    data[row][col] = value;

    if (DEBUG) {
        System.out.println("New value of data:");
        printDebugData();
    }
}

```

```

private void printDebugData() {
    int numRows = getRowCount();
    int numCols = getColumnCount();

    for (int i=0; i < numRows; i++) {
        System.out.print("    row " + i + ":");
        for (int j=0; j < numCols; j++) {
            System.out.print(" " + data[i][j]);
        }
        System.out.println();
    }
    System.out.println("-----");
}

}

/**
 * Create the GUI and show it. For thread safety,
 * this method should be invoked from the
 * event-dispatching thread.
 */
private static void createAndShowGUI() {
    //Create and set up the window.
    JFrame frame = new JFrame("Tabla ALD");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    //Creamos content pane y lo asignamos al frame
    TableSortDemo newContentPane = new TableSortDemo();

    frame.setContentPane(newContentPane);

    //Mostramos la ventana.
    frame.pack();
    frame.setVisible(true);
}

public static void main(String[] args) {
    javax.swing.SwingUtilities.invokeLater(new Runnable() {
        public void run() {
            createAndShowGUI();
        }
    });
}

```