

Ordenar y filtrar JTableModel

```
public VentanaPrincipal() {  
    initComponents();  
  
    AlumnoTableModel atm = new AlumnoTableModel(logicaNegocio.getListaAlumnos());  
    this.jTable1.setModel(atm);  
    //Para poder ordenar el table model creo el tablerowsorter  
    TableRowSorter<AlumnoTableModel> sorter = new TableRowSorter<>(atm);  
    //añado el tablerowsorter a mi jTable  
    this.jTable1.setRowSorter(sorter);  
}
```

Ordenada por defecto (Columna, tipo de orden)

```
AlumnoTableModel atm = new AlumnoTableModel(logicaNegocio.getListAlumnos());  
this.jTable1.setModel(atm);  
//Para poder ordenar el table model creo el tablerowsorter  
TableRowSorter<AlumnoTableModel> sorter = new TableRowSorter<>(atm);  
//añado el tablerowsorter a mi jTable  
this.jTable1.setRowSorter(sorter);
```

```
[ List<SortKey> sortKeys = new ArrayList<>();  
  sortKeys.add(new SortKey(0, SortOrder.ASCENDING));  
  sorter.setSortKeys(sortKeys);  
]
```

Problemas con elemento seleccionado

GeetselectedRow no funciona como queremos, ya que selecciona el número de fila que relaciona con la vista, no con el listado que tenemos guardado.



método que convierte el índice de la vista a nuestro índice

ConvertRowIndexToModel

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    int seleccionado = this.jTable1.convertRowIndexToModel(jTable1.getSelectedRow());  
    this.jLabel1.setText(logicaNegocio.getListAlumnos().get(seleccionado).getNombre());  
}
```

Filtrar la tabla

Title 1	Title 2	Title 3	Title 4

Aceptar

jLabel1

Filtrar

setRowFilter

```
List<SortKey> sortKeys = new ArrayList<>();  
sortKeys.add(new SortKey(0,SortOrder.DESENDING));  
sorter.setSortKeys(sortKeys);  
sorter.set
```

}	○ setComparator(int i, Comparator<?> cmptr)	void
	○ setMaxSortKeys(int i)	void
private void	○ setModel(TableModelAlumnos m)	void
{	○ setRowFilter(RowFilter<? super TableModelAlumnos, ? super Integer> rf)	void
	○ setSortKeys(List<? extends SortKey> list)	void
	○ setSortable(int i, boolean bln)	void
TableModelExample	○ setSortsOnUpdates(boolean bln)	void
	○ setStringConverter(TableStringConverter tsc)	void



[javax.swing.DefaultRowSorter](#)

Ponemos a la clase TableRowSorter

```
public class VentanaPrincipal extends javax.swing.JFrame {  
  
    private LogicaNegocio logicaNegocio = new LogicaNegocio();  
    private TableRowSorter<AlumnoTableModel> sorter ;  
    ...  
}
```


RowFilter no se construye con un New (es una factoría) Tiene sus propios métodos, utilizaremos regexFilter

```
private void jButtonFiltrarActionPerformed(java.awt.event.ActionEvent evt) {  
    RowFilter<TableModelAlumnos,Integer> rf = RowFilter.  
}
```

```
/**  
 * @param args the command line arguments  
 */  
public static void main(String args[]) {  
    /* Set the Nimbus look and feel */  
    Look and feel setting code (optional)
```

- andFilter(Iterable<? extends RowFilter<? su
- dateFilter(ComparisonType ct, Date date, in
- notFilter(RowFilter<M, I> rf)
- numberFilter(ComparisonType ct, Number num1
- orFilter(Iterable<? extends RowFilter<? su
- regexFilter(String string, int... ints)
- class
- ComparisonType
- Entry<M, I>

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    RowFilter<AlumnoTableModel,Integer> rf= RowFilter.regexFilter(this.jTextField1.getText(),0);  
    sorter.setRowFilter(rf);  
}
```