```
public JPanel createContentPane() {
  JPanel totalGUI = new JPanel(new BorderLayout(5, 5));
  textArea = new JTextArea(5, 30);
  JToolBar toolBar = new JToolBar("Still draggable");
totalGUI.add(toolBar, BorderLayout.PAGE_START);
  JButton load = new JButton();
  load.setActionCommand("load");
load.setToolTipText("Load a file");
load.addActionListener(this);
  load.setIcon( createImageIcon("open.png") );
  toolBar.add(load);
  JButton save = new JButton();
  save.setActionCommand("save");
save.setToolTipText("Save a file");
save.addActionListener(this);
save.setIcon( createImageIcon("save.png") );
  toolBar.add(save);
public void actionPerformed(ActionEvent e) {
    textArea.append("Clicked on "+e.getActionCommand() +"\n");
```

```
Clicked on save
Clicked on load
Clicked on load
```

```
public JPanel createContentPane() {
  JPanel totalGUI = new JPanel(new BorderLayout(5, 5));
  textArea = new JTextArea(5, 30);
  JToolBar toolBar = new JToolBar("Still draggable");
totalGUI.add(toolBar, BorderLayout.PAGE_START);
  JButton load = new JButton();
  load.setActionCommand("load");
load.setToolTipText("Load a file");
load.addActionListener(this);
  load.setIcon( createImageIcon("open.png") );
  toolBar.add(load);
  JButton save = new JButton();
  save.setActionCommand("save");
save.setToolTipText("Save a file");
save.addActionListener(this);
save.setIcon( createImageIcon("save.png") );
  toolBar.add(save);
public void actionPerformed(ActionEvent e) {
    textArea.append("Clicked on "+e.getActionCommand() +"\n");
```

```
Clicked on save
Clicked on load
Clicked on load
```

```
public JPanel createContentPane() {
  JPanel totalGUI = new JPanel(new BorderLayout(5, 5));
  textArea = new JTextArea(5, 30);
  JToolBar toolBar = new JToolBar("Still draggable"); totalGUI.add(toolBar, BorderLayout.PAGE_START);
  JButton load = new JButton();
  load.setActionCommand("load");
load.setToolTipText("Load a file");
load.addActionListener(this);
   load.setIcon( createImageIcon("open.png") );
  toolBar.add(load);
 JButton save = new JButton();
save.setActionCommand("save");
save.setToolTipText("Save a file");
save.addActionListener(this);
save.setIcon( createImageIcon("save.png") );
  toolBar.add(save);
public void actionPerformed(ActionEvent e) {
    textArea.append("Clicked on "+e.getActionCommand() +"\n");
```

```
Clicked on save
Clicked on load
Clicked on load
```

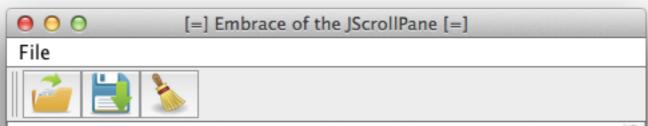
```
public JPanel createContentPane() {
  JPanel totalGUI = new JPanel(new BorderLayout(5, 5));
  textArea = new JTextArea(5, 30);
  JToolBar toolBar = new JToolBar("Still draggable");
totalGUI.add(toolBar, BorderLayout.PAGE_START);
  JButton load = new JButton();
  load.setActionCommand("load");
load.setToolTipText("Load a file");
load.addActionListener(this);
  load.setIcon( createImageIcon("open.png") );
  toolBar.add(load);
  JButton save = new JButton();
  save.setActionCommand("save");
save.setToolTipText("Save a file");
save.addActionListener(this);
save.setIcon( createImageIcon("save.png") );
  toolBar.add(save);
public void actionPerformed(ActionEvent e) {
    textArea.append("Clicked on "+e.getActionCommand() +"\n");
```

```
Clicked on save
Clicked on load
Clicked on load
```

```
public JPanel createContentPane() {
                                                                                                    [=] ToolBar [=]
  JPanel totalGUI = new JPanel(new BorderLayout(5, 5));
  textArea = new JTextArea(5, 30);
                                                                                        Clicked on load
                                                                                        Clicked on load
  JToolBar toolBar = new JToolBar("Still draggable");
totalGUI.add(toolBar, BorderLayout.PAGE_START);
  JButton load = new JButton();
 load.setActionCommand("load");
load.setToolTipText("Load a file");
load.addActionListener(this);
                                                                            setActionCommand es muy útil
                                                                            cuando un oyente escucha a
  load.setIcon( createImageIcon("open.png") );
                                                                            varios componentes. Podemos
  toolBar.add(load);
                                                                            usarlo en todas partes (menus,
 JButton save = new JButton();
save.setActionCommand("save");
save.setToolTipText("Save a file");
save.addActionListener(this);
save.setIcon( createImageIcon("save.png") );
                                                                            button, etc.).
  toolBar.add(save);
public void actionPerformed(ActionEvent e) {
    textArea.append("Clicked on "+e.getActionCommand() +"\n");
```

```
public JPanel createContentPane() {
                                                                                                     [=] ToolBar [=]
  JPanel totalGUI = new JPanel(new BorderLayout(5, 5));
  textArea = new JTextArea(5, 30);
                                                                                         Clicked on load
                                                                                         Clicked on load
  JToolBar toolBar = new JToolBar("Still draggable");
totalGUI.add(toolBar, BorderLayout.PAGE_START);
  JButton load = new JButton();
 load.setActionCommand("load");
load.setToolTipText("Load a file");
load.addActionListener(this);
                                                                             setActionCommand es muy útil
                                                                             cuando un oyente escucha a
  load.setIcon( createImageIcon("open.png") );
                                                                             varios componentes. Podemos
  toolBar.add(load);
                                                                             usarlo en todas partes (menus,
 JButton save = new JButton();
save.setActionCommand("save");
save.setToolTipText("Save a file");
save.addActionListener(this);
save.setIcon( createImageIcon("save.png") );
                                                                             button, etc.).
                                                                                  Se puede añadir (casi) todo a
                                                                                  un JToolBar
  toolBar.add(save);
                                                                                            [=] ToolBar [=]
                                                                                     Times Roman
public void actionPerformed(ActionEvent e) {
    textArea.append("Clicked on "+e.getActic Clicked on load Clicked on save
                                                                      Clicked on save
see: examples.swing.misc.ToolBarExample
```

Ejemplo: un Editor de Texto



The Swing library makes heavy use of the Model/View/Controller software design pattern,[6] which conceptually decouples the data being viewed from the user interface controls through which it is viewed. Because of this, most Swing components have associated models (which are specified in terms of Java interfaces), and the programmers can use various default implementations or provide their own. The framework provides default implementations of model interfaces for all of its concrete components. The typical use of the Swing framework does not require the creation of custom models, as the framework provides a set of default implementations that are transparently, by default, associated with the corresponding JComponent child class in the Swing library. In general, only complex components, such as tables, trees and sometimes lists, may require the custom model implementations around the application-specific data structures. To get a good sense of the potential that the Swing architecture makes possible, consider the hypothetical situation where custom models for tables and lists are urranners over DAO and/or FIR consises