

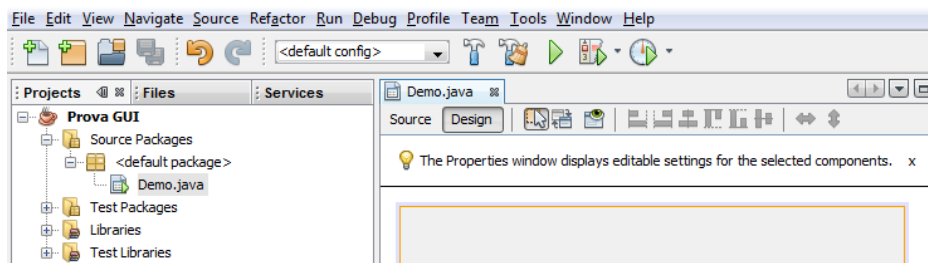
# Ajuda pel Lliurament 4

En aquest document anem a explicar com construir una Interfície Gràfica d'Usuari (GUI) fent servir l'Entorn de Desenvolupament Integrat (IDE) NetBeans. Trobareu també ajuda per utilitzar la classe JDialog i JFileChooser.

## 1. Projecte GUI

Realitzeu el projecte “ProvaGUI” amb l'entorn NetBeans, seguint els següents passos:

1. Creeu un nou projecte “ProvaGUI” sense classe Main (desactivant opció Create Main Class de l'assistent “New Java Application”: “2. Name and Location”).
2. Creeu una classe de tipus JFrame (“JFramer Form”) anomenada “Demo”.
3. L'entorn Netbeans disposa d'eines per facilitar el desenvolupament de l'aplicació gràfica i les organitza de la següent manera
  - Es disposa de dues pestanyes Source/Design
    - La pestanya Source, conté el codi de la classe Demo que hereta de javax.swing.JFrame.  
Hi ha part del codi en gris que no es pot modificar directament sinó que s'haurà de fer mitjançant les eines disponible a la pestanya Design.
    - La pestanya Design, visualitza l'aplicació gràfica generada i conté una sèrie d'eines gràfiques per modificar-la: paleta d'objectes/classes (SWING) i propietats del JFrame.

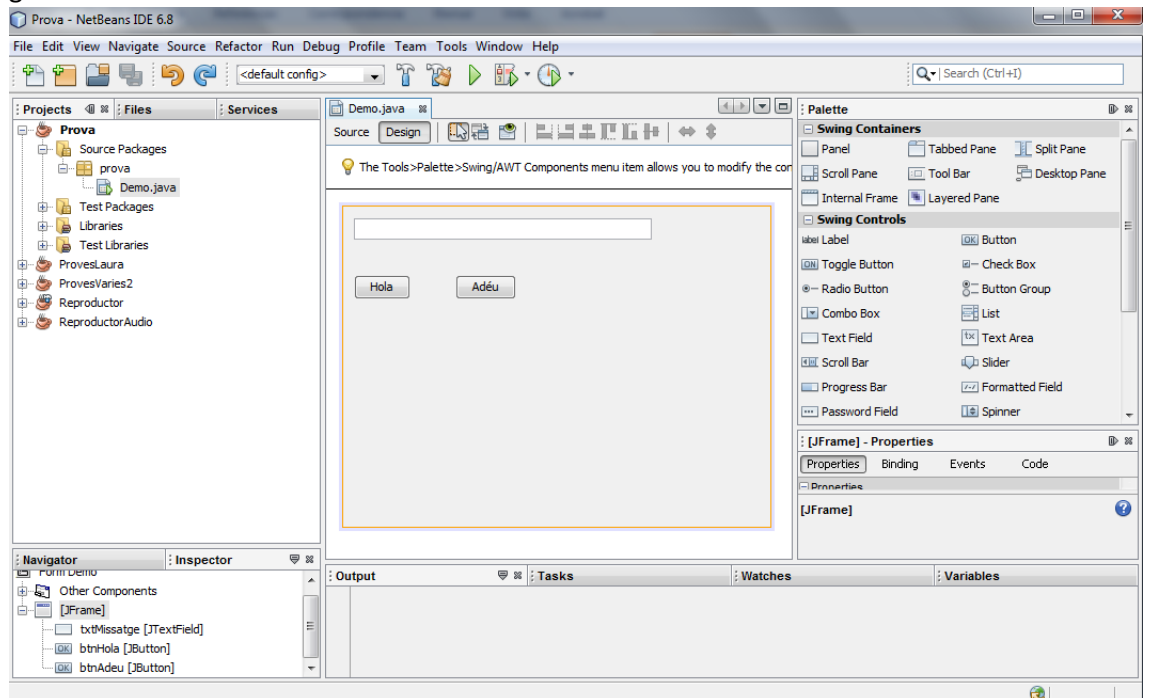


4. Podeu crear qualsevol tipus de camps (JLabels, JButtons, ButtonGroups, JTrees, JTextFields, ComboBoxes ) arrastrant amb el ratolí des de la paleta fins al JFrame
5. Creeu un camp tipus text per mostrar text i dos botons
  - Tindreu un objecte de tipus JTextField i dos objectes de tipus JButton
6. El NetBeans (constructor de GUI) automàticament controla la posició i alineament correcte dins del JFrame.
7. Podeu realitzar diferents accions sobre aquests controls editant directament les seves propietats (amb menú del botó dret del ratolí o en la finestra de la dreta: “properties”)
8. Canvieu els noms de les components
  - Canvieu el text que contenen els botons:
    - Canvieu de jButton1 i jButton2 a Hola i Adéu
  - Elimineu el text que conté el camp de text:
    - jTextField1

## Programació 2.

Grau d'Enginyeria Informàtica. Facultat de Matemàtiques. UB

- D'aquesta manera obtenim la següent configuració de l'aplicació gràfica:



- La pròpia finestra tindrà un títol que es pot afegir.
- Aneu a la pestanya Source i fixeu-vos en com ha canviat el codi fins ara.

### 9. Renombrreu les variables

- Aneu a la finestra de la dreta: "code" i canvieu els noms de les variables a algun nom amb més significat per a l'aplicació (recordeu que per conveni els noms de les variables comencen amb minúscula)
  - i. De jTextField1 a txtMissatge
  - ii. De jButton1 i jButton2 a btnHola i btnAdéu.

### 10. Podeu afegir funcions a cada un dels botons (assignació d'events)

- Us col·loqueu sobre el botó "Hola"
- Aneu a la finestra de la dreta: "Events" i seleccioneu actionPerformed o aneu al menú del boto dret del ratolí: Events → Action → actionPerformed.
- El IDE crearà automàticament un "ActionListener" per al botó "Hola" i genera un mètode de manipulació.
- El IDE obrirà la finestra del codi font i es situarà on has d'implementar l'acció que vols que es realitzi quan el botó es pressiona (amb el ratolí o amb el teclat). El teu codi continuarà el mètode:

```
private void  
btnHolaActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
}
```

que queda preparat per afegir codi.

- Afegiu el codi per a que s'escrigui el text corresponent al botó (Hola) dins del camp de text txtMissatge (quan premeu el botó btnHola)

```
private void  
btnHolaActionPerformed(java.awt.event.ActionEvent evt) {  
    txtMissatge.setText(btnHola.getText());  
}
```

## Programació 2.

Grau d'Enginyeria Informàtica. Facultat de Matemàtiques. UB

---

11. Feu el mateix amb el botó “Adéu”
12. Creeu un control check box
  - Canvieu el text de JCheckBox1 a Botons Actius
  - Canvieu el nom de la variable a chkBotonsActius
13. Per tal de configurar els botons actius o desactius es pot afegir al constructor de la classe “Demo” les següents línies:

```
chkBotonsActius.setSelected(true);
txtMissatge.setEnabled(true);
```
14. Doneu-li la següent funcionalitat: quan està activat els botons “Hola” i “Adéu” s’activen i quan està desactivat els botons es desactiven
  - Aneu a la finestra de la dreta: “Events” i seleccioneu stateChanged:

```
chkBotonsActiusStateChanged.
```

Es crea automàticament el mètode “Listener”:

```
private void
chkBotonsActiusStateChanged(javax.swing.event.ChangeEvent evt)
```

on podeu afegir el codi per activar els botons “Hola” i “Adéu”:

```
btnHola.setEnabled(chkBotonsActius.isSelected());
btnAdeu.setEnabled(chkBotonsActius.isSelected());
```

## 2. Projecte GUI (Part 2)

---

15. Creeu un nou botó “Mostrar Llista”
  - Canvieu el text de jButton1 a “Mostrar Llista”
  - Canvieu el nom de la variable a btnMostrar
16. Per la gestió de llistes, creeu un nou control arrastrant des de la paleta un objecte de tipus JList. Fixeu-vos en la finestra Inspector (situada a baix a l’esquerra) que el camp de l’objecte llista sempre va dins d’un objecte tipus JScrollPane
  - Canvieu el nom de la variable a lstLlista1

17. Afegiu un atribut a la classe Demo:

```
ArrayList<A> llista = null;
```

Per fer-ho necessitareu l’import:

```
import java.util.ArrayList;
```

On A és la classe:

```
class A {
    int x;
    int y;

    public A(int x, int y){
        this.x=x;
        this.y=y;
    }
    @Override
    public String toString(){
        return "x= " + x + "\ny= " + y;
    }
}
```

18. Afegiu el mètode de suport afegirDadesTest() al constructor de la classe Demo.

```
private void afegirDadesTest() {
```

## Programació 2.

Grau d'Enginyeria Informàtica. Facultat de Matemàtiques. UB

---

```
llista = new ArrayList<A>();  
llista.add(new A(1,2));  
llista.add(new A(3,4));  
llista.add(new A(5,6));  
}
```

19. Afegiu funcionalitat quan es premi el botó “Mostrar Llista” (com ja sabeu, des de menu→ Events→Action→actionPerformed)

Feu que quan es premi el botó es mostri la llista al lstLlista1 afegint el mètode omplirLlista():

```
private void  
btnMostrarActionPerformed(java.awt.event.ActionEvent evt) {  
    omplirLlista();  
}
```

Definit com segueix:

```
void omplirLlista() {  
    DefaultListModel model = new DefaultListModel();  
    model.clear();  
    for(A item: llista) {  
        model.addElement(item);  
    }  
    lstLlista1.setModel(model);  
}
```

Recordeu que se necessita un import:

```
import javax.swing.DefaultListModel;
```

o alternativament:

```
import javax.swing.*;
```

20. Afegiu un botó per eliminar elements de la llista

- Creeu un botó: objecte de tipus JButton
- Canvieu el text que contenen els botons: de jButton1 a Eliminar
- Canvieu el nom de la variable a btnEliminar

21. Feu el botó eliminar inactiu per defecte i que s'activi quan seleccionem algun element de la llista

- Afegiu al constructor de la classe Demo: btnEliminar.setEnabled(false);

22. Feu que aquest botó s'activi quan seleccionem algun element de la llista

- Sobre la llista aneu al menú del boto dret del ratolí: Events → ListSelection → valueChanged
- El IDE obrirà la finestra del codi font i es situarà al mètode on has d'implementar l'acció que vols que es realitzi quan es selecciona algun element:

```
private void  
lstLlista1ValueChanged(javax.swing.event.ListSelectionEvent evt)  
{  
    // TODO add your handling code here:  
}
```

- Afegeix el següent codi:

```
btnEliminar.setEnabled(!lstLlista1.isSelectionEmpty());
```

23. Afegiu al botó la funcionalitat d'eliminar l'element de la llista seleccionat

- Aneu a “Events” i seleccioneu actionPerformed. Llavors, es crearà automàticament el mètode:

```
private void  
btnEliminarActionPerformed(java.awt.event.ActionEvent evt) {
```

## Programació 2.

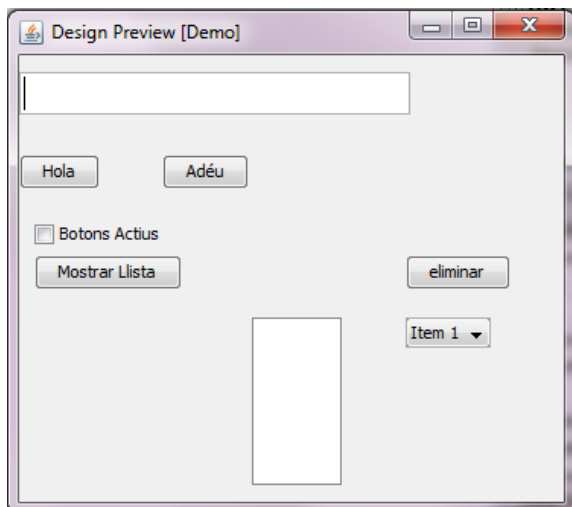
Grau d'Enginyeria Informàtica. Facultat de Matemàtiques. UB

```
        // TODO add your handling code here:
    }
```

on heu d'afegir el codi per eliminar l'element de la llista seleccionat:

```
for(Object item: lstLlista1.getSelectedValuesList()){
    A elementSeleccionat = (A) item;
    llista.remove(elementSeleccionat);
}
omplirLlista();
```

A la següent figura en teniu l'estat final de l'aplicació:



Us adjuntem el codi d'aquesta aplicació per a que tingueu un exemple.

## 3. Exercicis

1. Afegiu un control Combo Box i mostreu la mateixa llista utilitzant el botó "Mostrar Llista". Feu que el botó "Eliminar" elimini els elements seleccionats del Combo Box.

## Referències

- <http://java.sun.com/docs/books/tutorial/uiswing/>
- Guia per començar: <http://netbeans.org/kb/trails/matisse.html>. Els capítols més interessants:
  - Introducció per construir GUI amb NetBeans:  
<http://netbeans.org/kb/docs/java/gui-functionality.html>
  - Pas a pas del disseny d'un GUI amb Swing en l'entorn NetBeans:  
<http://netbeans.org/kb/docs/java/quickstart-gui.html>

### 4. JDialog

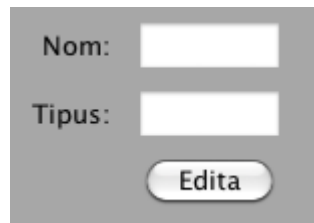
---

Les finestres de dialog són subfinestres independents de la finestra principal de l'aplicació, però gestionades per aquesta. A continuació crearem una aplicació que omple uns camps de text agafant les dades d'un JDialog.

- 1- Creem la classe Arxiu que representarà un Arxiu:

```
public class Arxiu {  
    private String nom;  
    private String tipus;  
  
    public String getNom() {  
        return nom;  
    }  
    public void setNom(String nom) {  
        this.nom = nom;  
    }  
    public String getTipus() {  
        return tipus;  
    }  
    public void setTipus(String tipus) {  
        this.tipus = tipus;  
    }  
}
```

- 2- Creem un JFrame From, anomenat GUIDemo, amb dos camps de text (*txtNom* i *txtTipus*) i un botó (*btnEdita*)
- Editem *txtNom* i *txtTipus* per a que no siguin editables (*click dret sobre el quadre de text->propiedades i desmarquem "enabled"*)



- La classe contindrà una instància d'Arxiu que es crea al constructor:

```
Arxiu arxiu;  
  
public GUIDemo() {  
    initComponents();  
    arxiu=new Arxiu();  
}
```

- 3- Creem una finestra de diàleg. *Clic botó dret sobre el nom del paquet->nuevo->JDialog Form*. Si no apareix l'opció aneu a "Otros" i Swing GUI Forms per cercar el *JDialog Form*.
- Posem com a nom *DadesDialog*
  - Eliminem el mètode *main* de *DadesDialog*

## Programació 2.

Grau d'Enginyeria Informàtica. Facultat de Matemàtiques. UB

---

- c. Insertem un camp de text (*txtNom*) i dos botons (*btnCancela* i *btnAccepta*)



- d. Inserim un comboBox amb dos elements seleccionables “Video” i “Audio”  
e. La classe contindrà un objecte de tipus Arxiu, anomenem-lo file, que rebrà per paràmetre  
f. Creem els events per als botons que contindran el següent codi

**btnCancelaActionPerformed:**

```
//Tanca la finestra de dialeg  
this.dispose();
```

**btnAcceptaActionPerformed:**

```
//Actualitza les dades del fitxer  
file.setNom(txtNom.getText());  
file.setTipus((String) cmbTipus.getSelectedItem());  
//Tanca la finestra de dialeg  
this.dispose();
```

- 4- Creem l’event per al botó editar del JFrame que s’encarregarà de crear el selector de fitxer i actualitzar els camps de textt:

**btnEditaActionPerformed:**

```
//Creem la finestra de dialog  
DadesDialog dialog = new DadesDialog(this,true,arxiu);  
//Titol de la finestra  
dialog.setTitle("Informació d'arxiu");  
//Mostra la finestra  
dialog.pack();  
dialog.setVisible(true);  
//Actualitza els camps de text  
txtNom.setText(arxiu.getNom());  
txtTipus.setText(arxiu.getTipus());
```

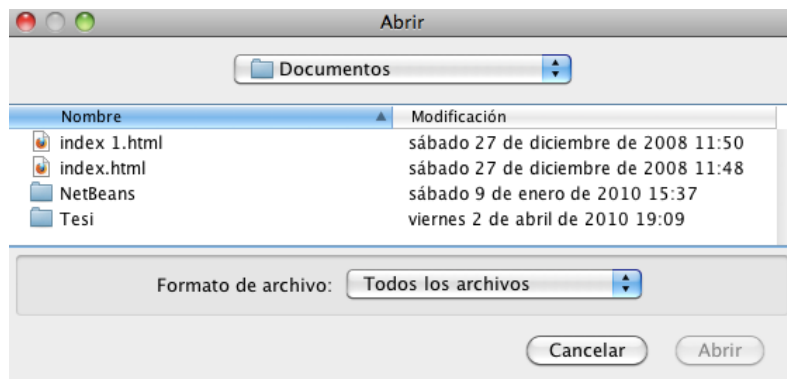
## 5. JFileChooser

---

Els selectors de fitxers proveeixen una GUI per navegar pel sistema de fitxers. A Java la classe JFileChooser ens permet crear una finestra de diàleg que conté aquest selector de fitxers com la que es mostra a continuació:

## Programació 2.

Grau d'Enginyeria Informàtica. Facultat de Matemàtiques. UB



A continuació veurem un exemple de selecció de fitxers mitjançant JFileChooser, en el que un camp de text s'actualitza en canviar la selecció del fitxer.

1- Creeu un JDialog amb un boto (*btnSelecciona*) i un camp de text(*txtNomFitxer*)



2 – Creem l'event del botó selecciona

**btnSeleccionaActionPerformed:**

```
File fitxer;
//Creació del selector de fitxer
JFileChooser seleccio = new JFileChooser();
//Mostrem la finestra de dialóg
//Resultat emmagazema una constant que indica si s'ha
//seleccionat o no un fitxer
int resultat = seleccio.showOpenDialog(this);
//Assegurem que hi hagi un fitxer seleccionat
if (resultat == JFileChooser.APPROVE_OPTION) {
    //Obtenim el fitxer
    fitxer=seleccio.getSelectedFile();
    //Posem la ruta del fitxer al quadre de text
    txtNomFitxer.setText(fitxer.toString());
}
```

## Referències

[1] <http://java.sun.com/docs/books/tutorial/uiswing/components/dialog.html>

[2] <http://java.sun.com/docs/books/tutorial/uiswing/components/filechooser.html>