Unió Europea
Fons Social Europeu
L'FSE inverteix en el teu futur

GENERALITAT VALENCIANA

I.E.S. Maria Enríquez
Gandia

UF12 – PROGRAMACIÓ GRÀFICA

- Teoria -

PROGRAMACIÓ
CFGS DAM

Autor:
Àngel Olmos Giner

segons el material de Carlos Cacho i Raquel Torres

a.olmosginer@edu.gva.es

2022/2023

PROGRAMACIÓ GRÀFICA

ÍNDEX DE CONTINGUTS

1. INTRODUCCIÓ

- Una interfície gràfica d'usuari, coneguda també com ***GUI*** (de l'anglés *Graphical User Interface*), és un programa informàtic que proporciona un entorn visual senzill per a permetre la comunicació i interacció de l'usuari amb el programa
- Exemples ? ...

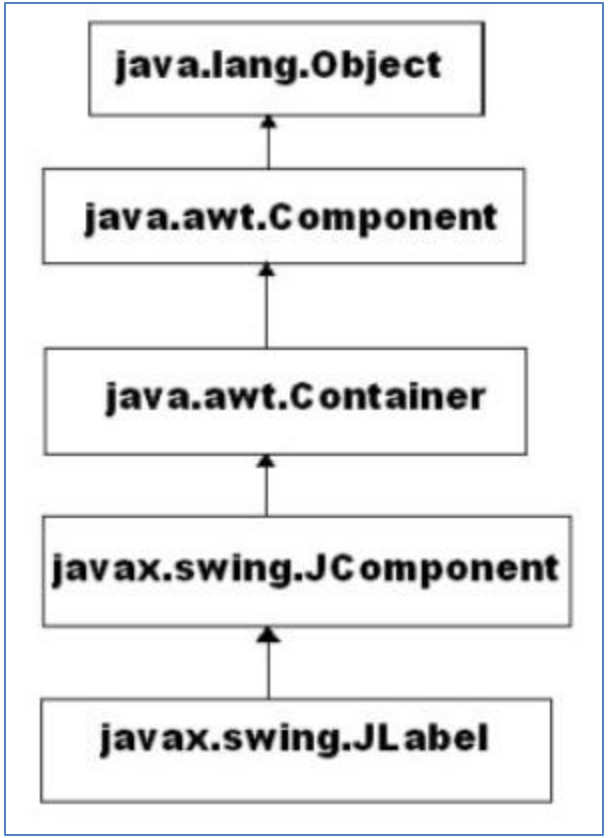
- Per a crear *GUIs* utilitzarem ***Swing***: biblioteca gràfica de Java que inclou *widgets*:
  - ❑ Caixes de text
  - ❑ Botons
  - ❑ Llistes desplegables ...
- *Swing* està basat en el *Abstract Window Toolkit* (**AWT**), que és un kit d'eines de gràfiques, interfícies d'usuari i sistema de finestres

PROGRAMACIÓ GRÀFICA

ÍNDEX DE CONTINGUTS

2. CREAR UNA APLICACIÓ *SWING*

- Tots els elements *Swing* hereten de la classe *javax.swing.JComponent*, la qual hereta de *java.awt.Container* i aquesta, al seu torn, de *java.awt.Component*
- Conéixer aquesta jerarquia ens permetrà conéixer algunes de les característiques comuns.
- Per exemple podem veure la jerarquia d'herència del component *JLabel*, que representa una etiqueta
- *JButton, JPanel, JTextField, JTextArea* ...



2. CREAR UNA APLICACIÓ *SWING*

Els passos a seguir per a crear una GUI són els següents:

a) Crear un projecte
b) Crear una finestra d'aplicació → ***JFrame***
c) Afegir un contenidor → ***JPanel***
d) Afegir components al contenidor
e) Canviar propietats dels components
f) Programar events

2. CREAR UNA APLICACIÓ *SWING*

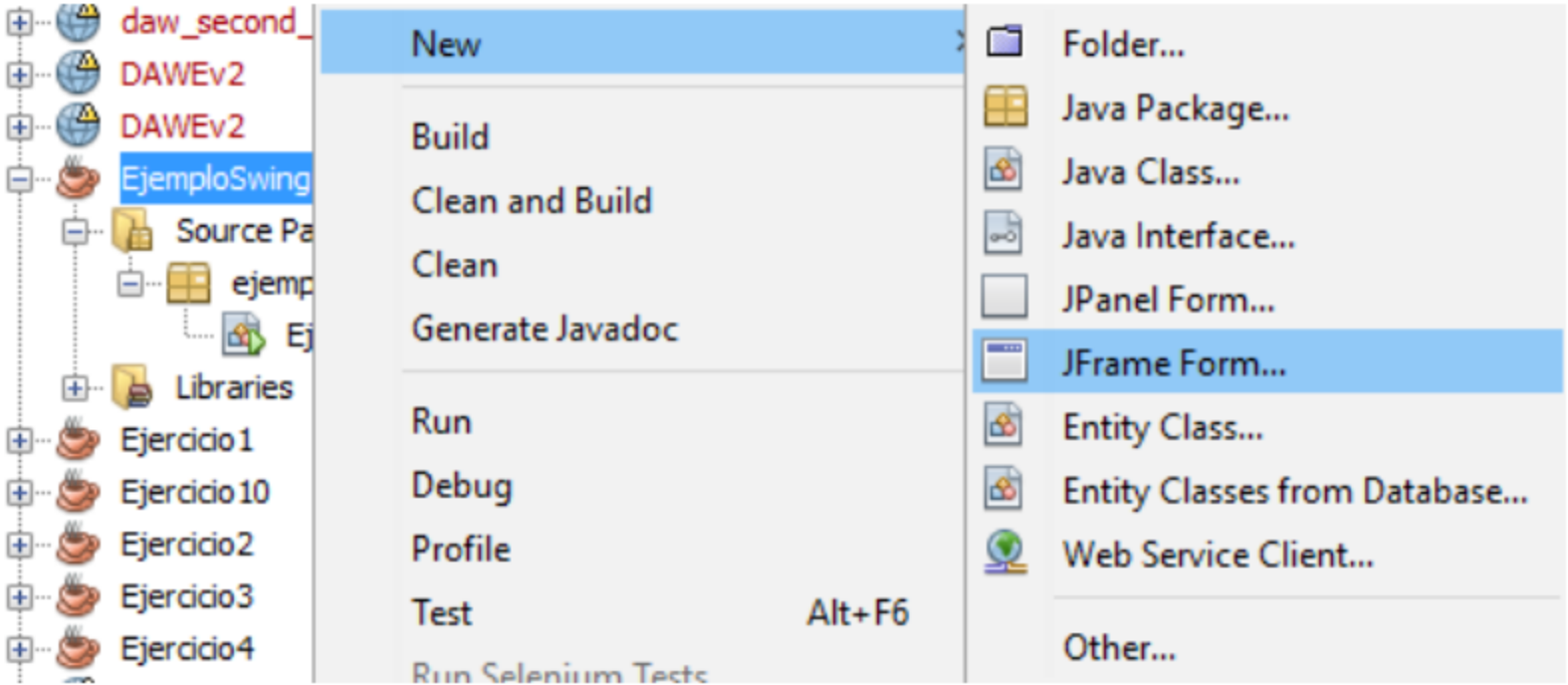
a) Crear un projecte

Igual que ho hem fet fins ara ...

però desmarcant l'opció de "Create Main Class"

2. CREAR UNA APLICACIÓ *SWING*

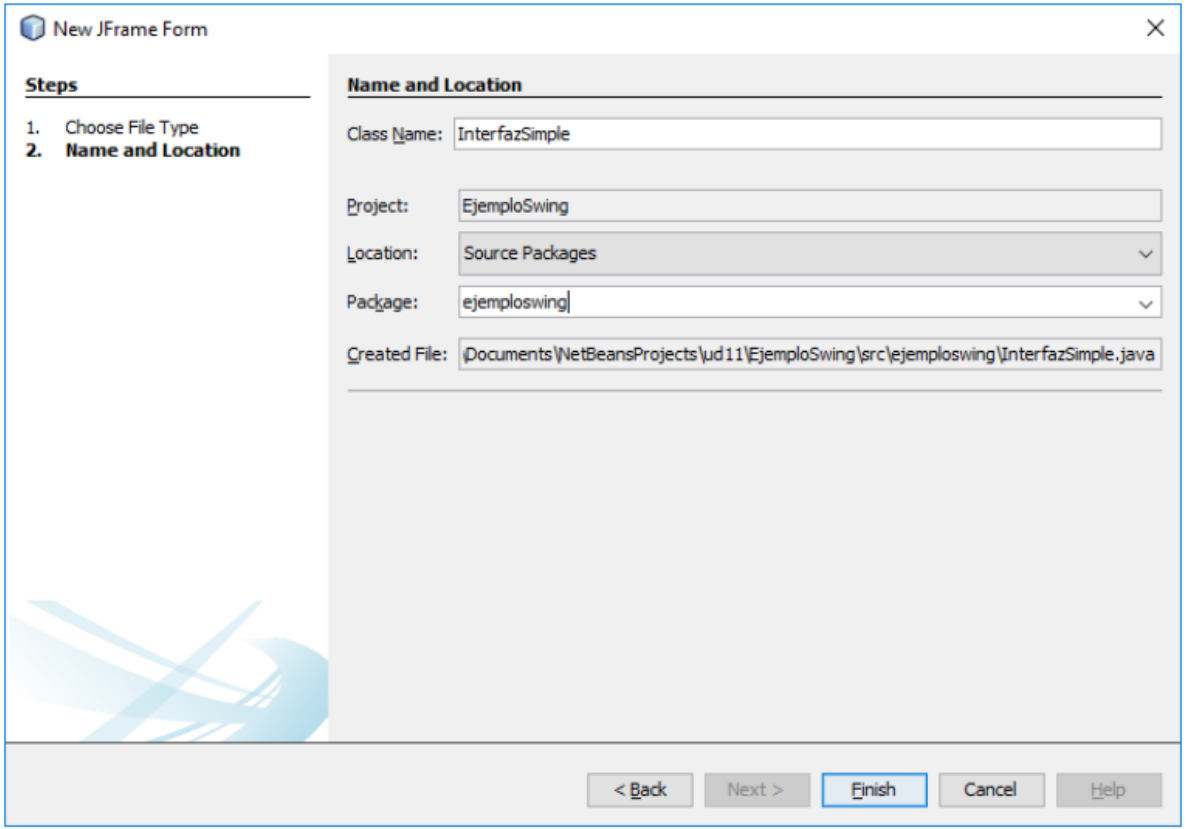
b) Crear una finestra d'aplicació → *JFrame*



2. CREAR UNA APLICACIÓ *SWING*

b) Crear una finestra d'aplicació → *JFrame*

- Escriviu per exemple *InterfazSimple* com a nom de la classe
- Afegiu *"el nom del vostre projecte"* com a nom del *package*



2. CREAR UNA APLICACIÓ *SWING*

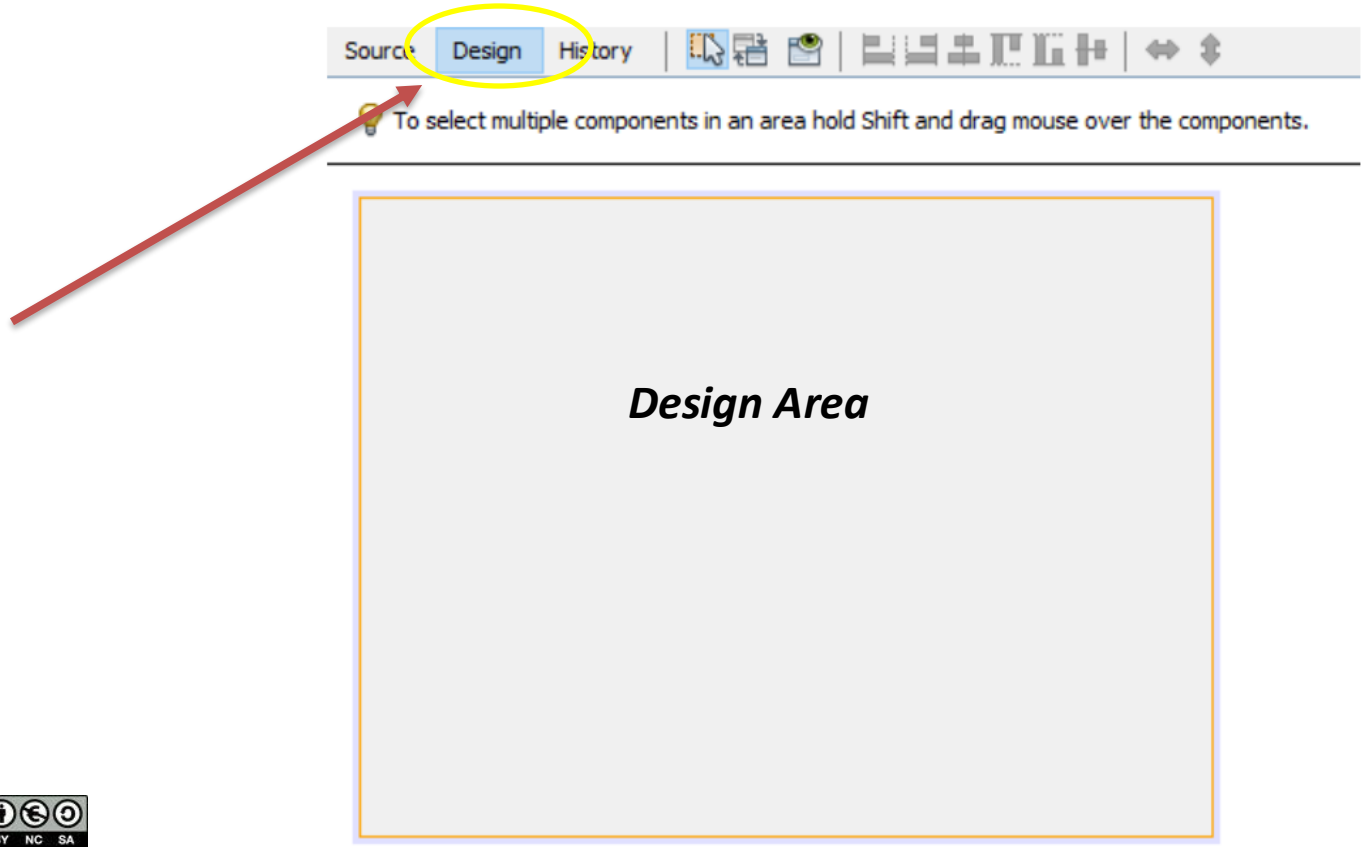
b) Crear una finestra d'aplicació → *JFrame*

En l'entorn *NetBeans* podem distingir:

1. ***Design Area***: Finestra principal per a crear i modificar la GUI
2. ***Navigator***: Mostra la jerarquia de components de la nostra aplicació
3. ***Palette***: Llista de tots els components que podem afegir a la nostra aplicació gràfica
4. ***Properties***: Mostra les propietats del component seleccionat

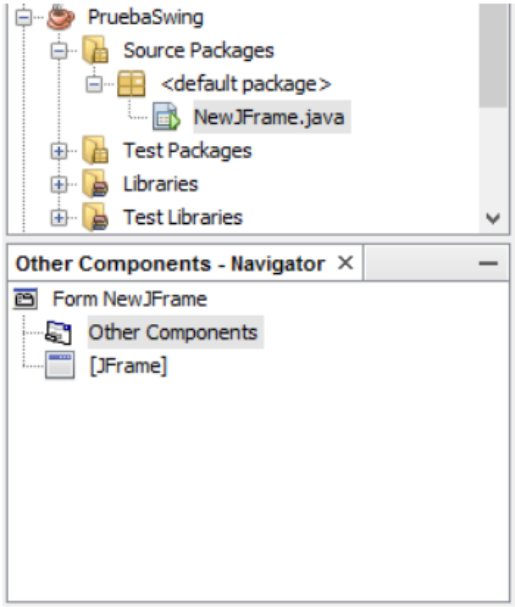
2. CREAR UNA APLICACIÓ *SWING*

b) Crear una finestra d'aplicació → *JFrame*



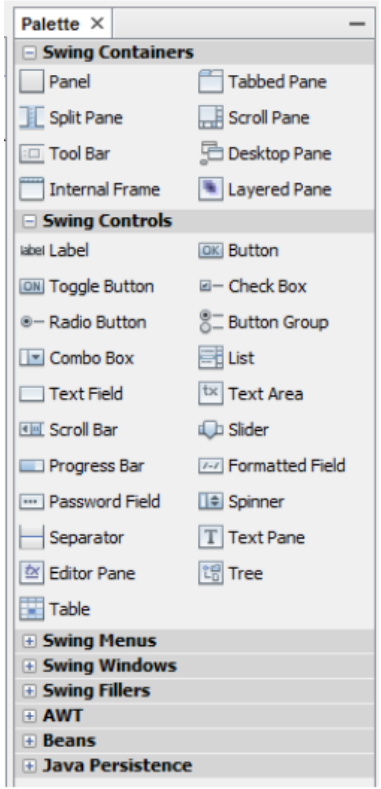
2. CREAR UNA APLICACIÓ *SWING*

b) Crear una finestra d'aplicació → *JFrame*



Navigator

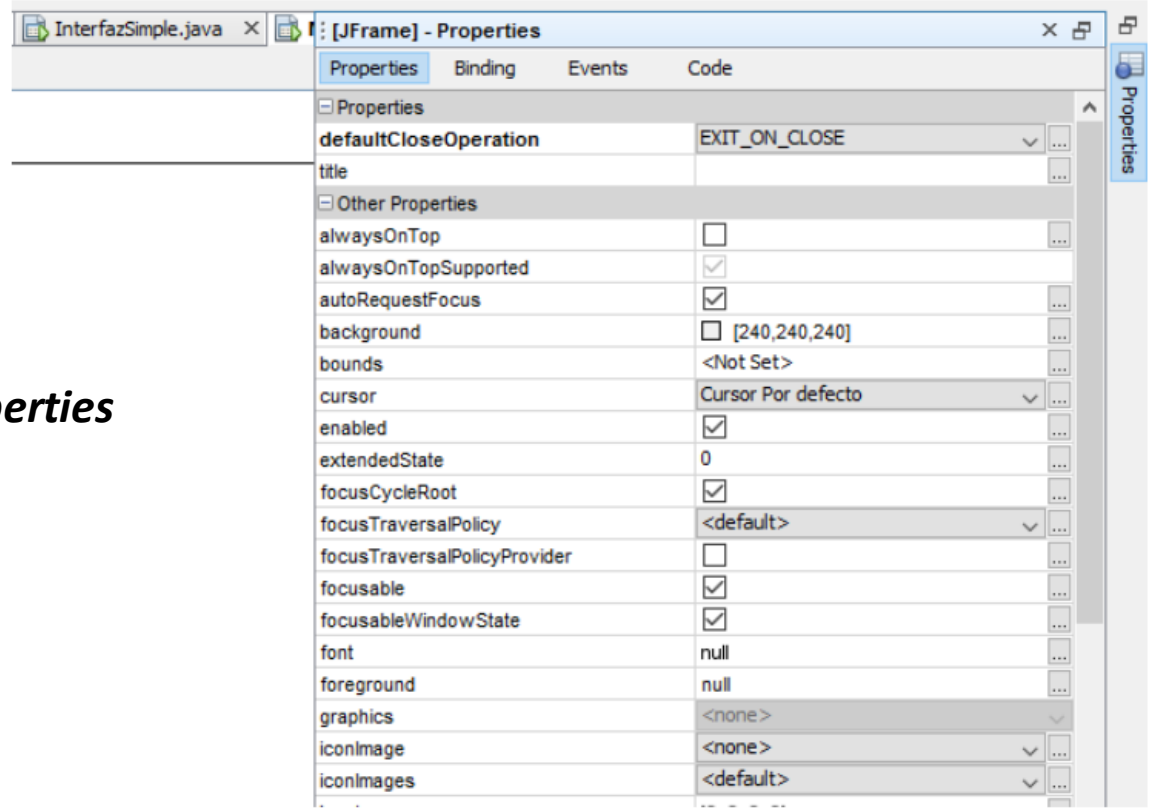
Palette



2. CREAR UNA APLICACIÓ *SWING*

b) Crear una finestra d'aplicació → *JFrame*

Properties



2. CREAR UNA APLICACIÓ *SWING*

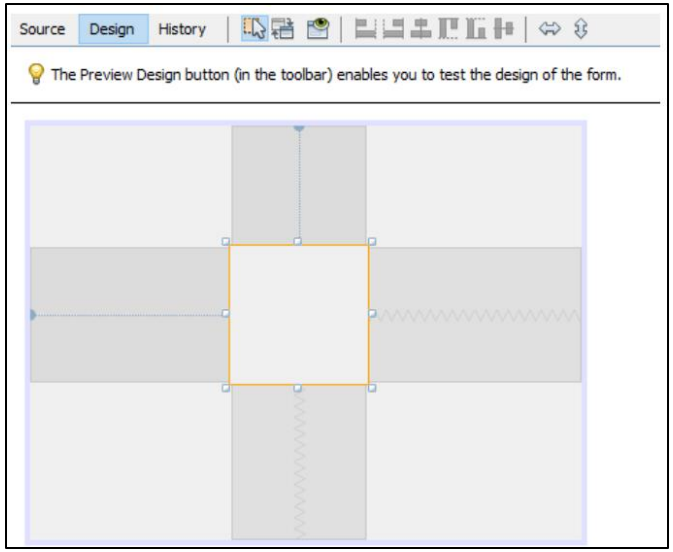
c) Afegir un contenidor → *JPanel*

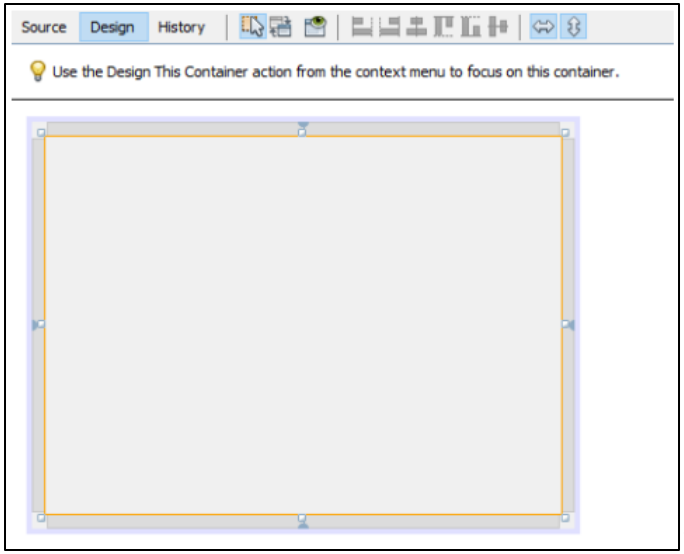
- Un contenidor és un **element no visual per a organitzar els components** de la nostra aplicació, és a dir, els botons, text, menús, imatges, etc.
- En *Swing* existeixen molts tipus de contenidors. Les seues diferencies radiquen en la forma en la qual manegen els components que tenen dins.
- Per exemple, el *JTabbedPane* és un contenidor amb pestanyes on cadascuna està associada a un component
- El *JSplitPane* és un contenidor que es comporta com un panell dividit en dos
- Nosaltres utilitzarem el **contenidor** de propòsit general ***JPanel***

2. CREAR UNA APLICACIÓ *SWING*

c) Afegir un contenidor → *JPanel*

Arrossegueu el *JPanel* dins del contenidor *JFrame* i ho ajusteu a les vores del *JFrame*



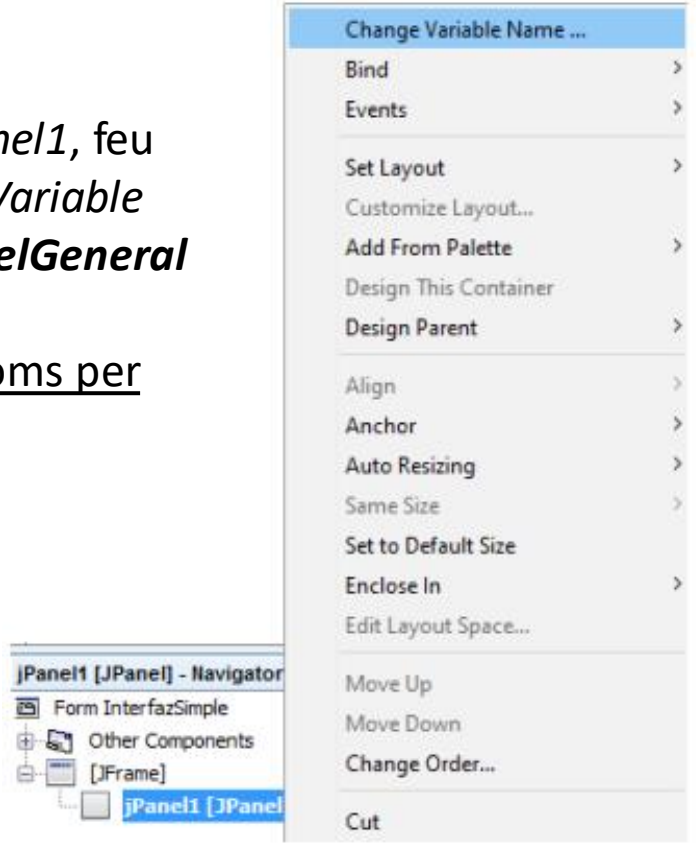


2. CREAR UNA APLICACIÓ *SWING*

c) Afegir un contenidor → *JPanel*

En el panell *Navegator* seleccioneu *jPanel1*, feu clic-dret i seleccioneu l'opció "*Change Variable Name...*" per a **canviar-li el nom a *jPanelGeneral***

És molt recomanable canviar tots els noms per defecte de les variables i anomenar els components amb un nom descriptiu



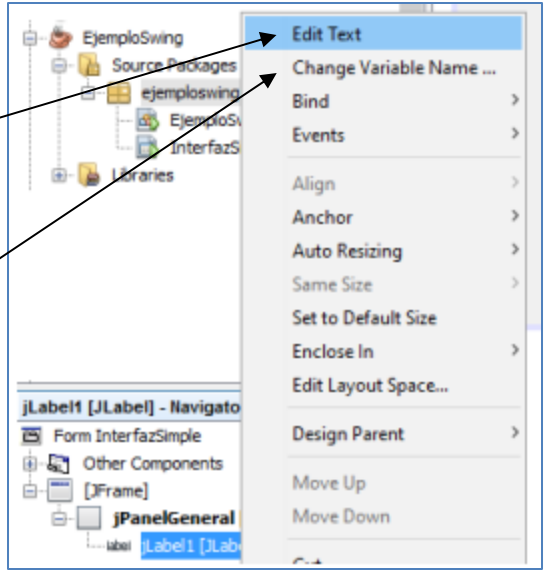
2. CREAR UNA APLICACIÓ *SWING*

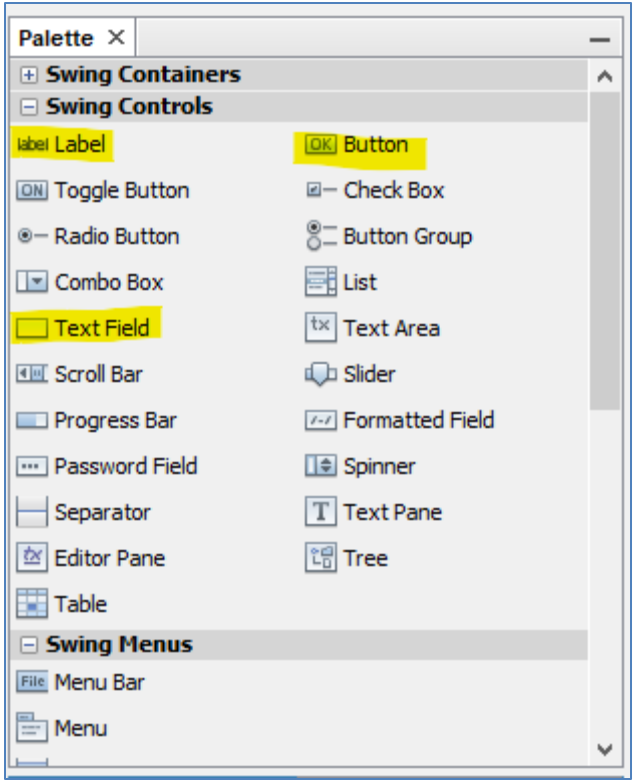
d) Afegir components al contenidor

Ara només queda afegir els components:

1. Una etiqueta (*JLabel*)
2. Un camp de text (*JTextField*)
3. Un botó (*JButton*)

Modifiqueu el text que mostren **i el nom de la seua variable** amb els menús contextuals

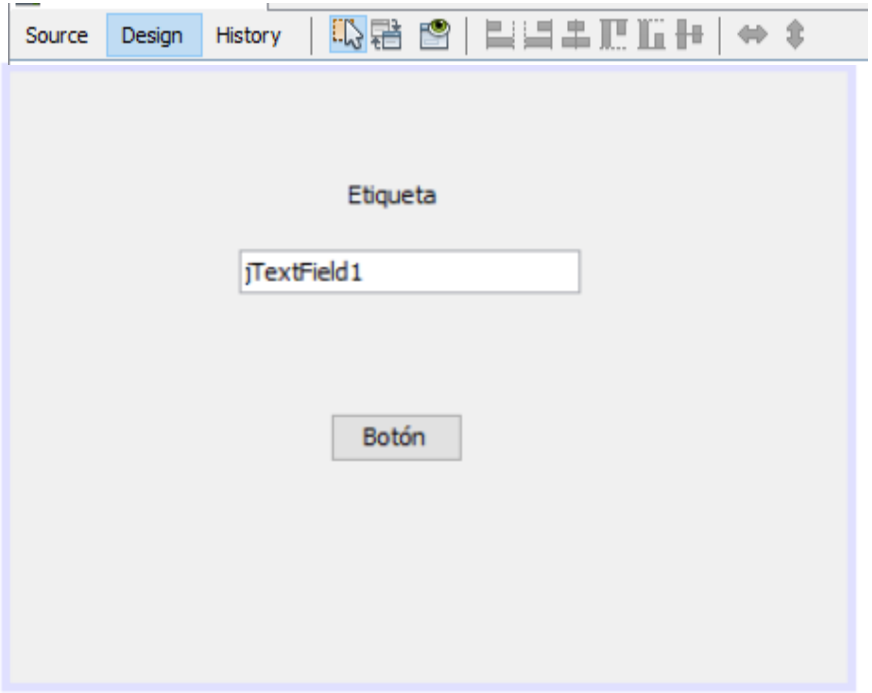




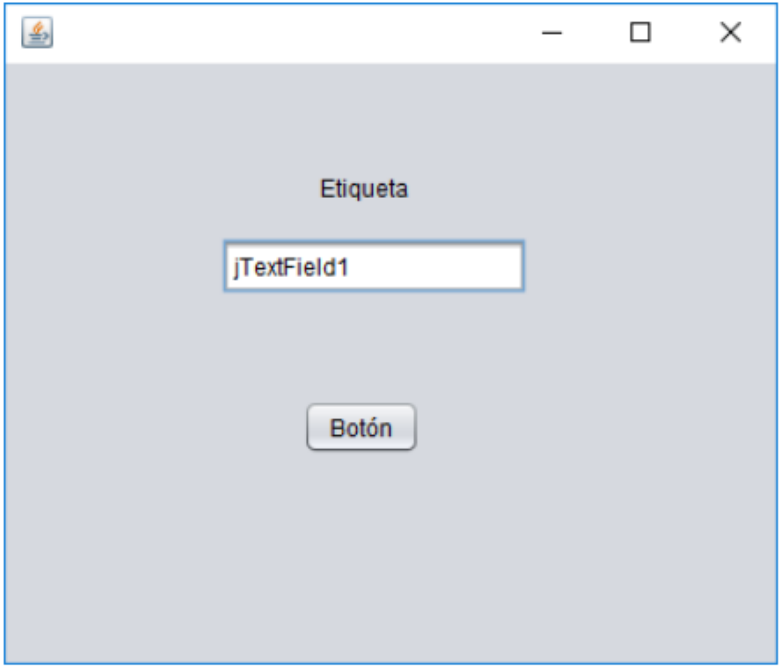
2. CREAR UNA APLICACIÓ *SWING*

d) Afegir components al contenidor

Disseny final



Quan executem el programa



PROGRAMACIÓ GRÀFICA

ÍNDEX DE CONTINGUTS

1. INTRODUCCIÓ
2. CREAR UNA APLICACIÓ *SWING*
3. ESDEVENIMENTS (EVENT)
4. COMPONENTS

3. ESDEVENIMENTS (*EVENT*)

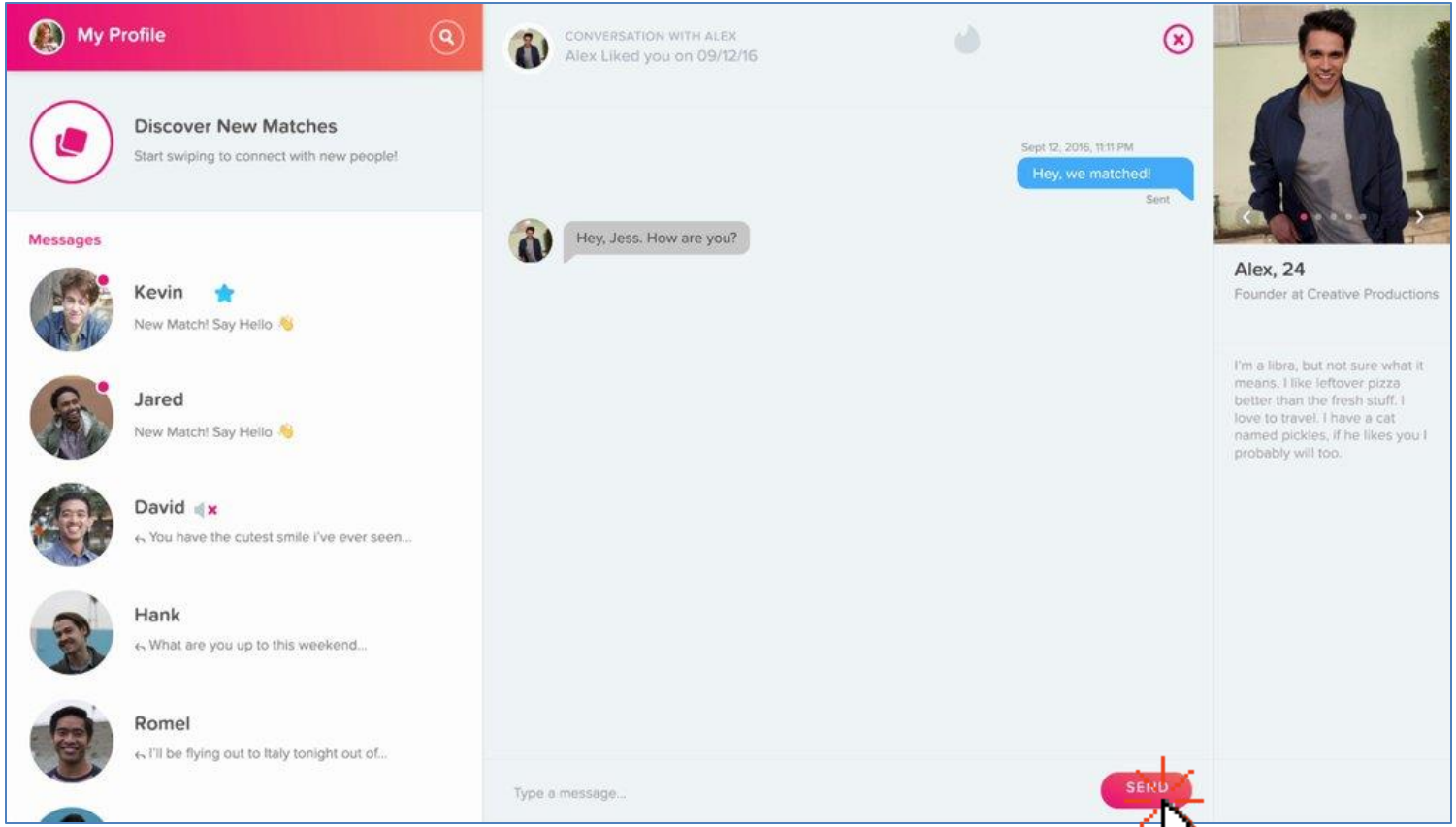
- Tots els sistemes operatius estan constantment atesos els esdeveniments generats pels usuaris

- Aquests esdeveniments poden ser **prémer una tecla del teclat, moure el ratolí, fer clic esquerre, clic dret, moure la roda del ratolí, etc.**

- Java distingeix entre simplement prémer el ratolí en un lloc qualsevol o fer-lo, per exemple, en un botó

- Quan es produeix un esdeveniment el sistema operatiu notifica a l'aplicació involucrada i aquesta decideix què fer amb aquest esdeveniment → s'executa un mètode associat a l'esdeveniment

3. ESDEVENIMENTS (*EVENT)*



3. ESDEVENIMENTS (*EVENT)*

El model de Java es basa en la **delegació d'esdeveniments**:

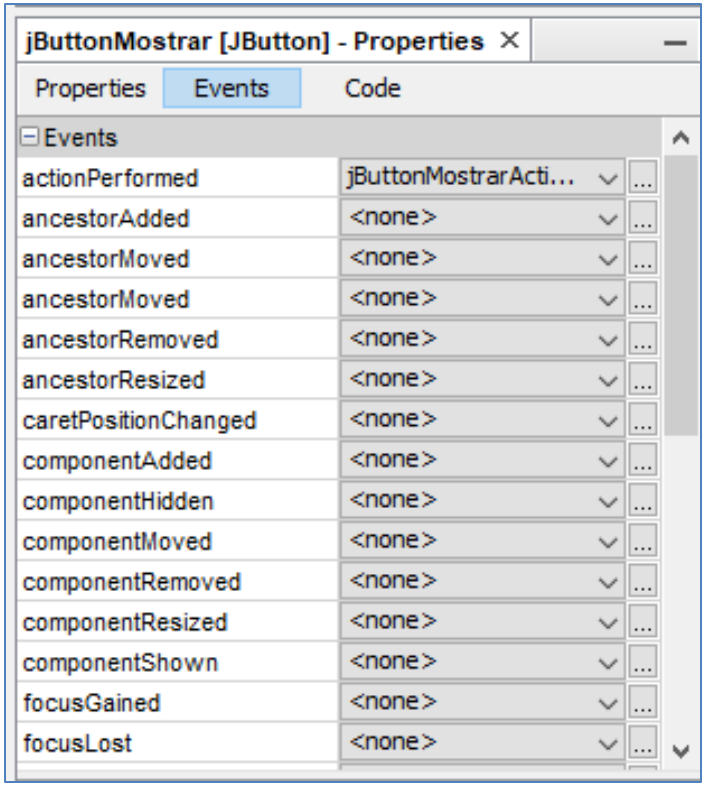
1. L'esdeveniment es produeix en un determinat component, per exemple un *JButton*, que serà la “font de l'esdeveniment” (*ActionListener*)
2. A continuació l'esdeveniment es transmet a un “manejador d'esdeveniments” (*EventListener*) assignat al component *JButton*
3. L'objecte que escolta els esdeveniments és el que s'encarregarà de respondre a ells adequadament

Aquesta separació de codi entre generació de l'esdeveniment i actuació respecte a ell simplifica el codi i facilita la labor de programació

3. ESDEVENIMENTS (*EVENT*)

Com crear un esdeveniment

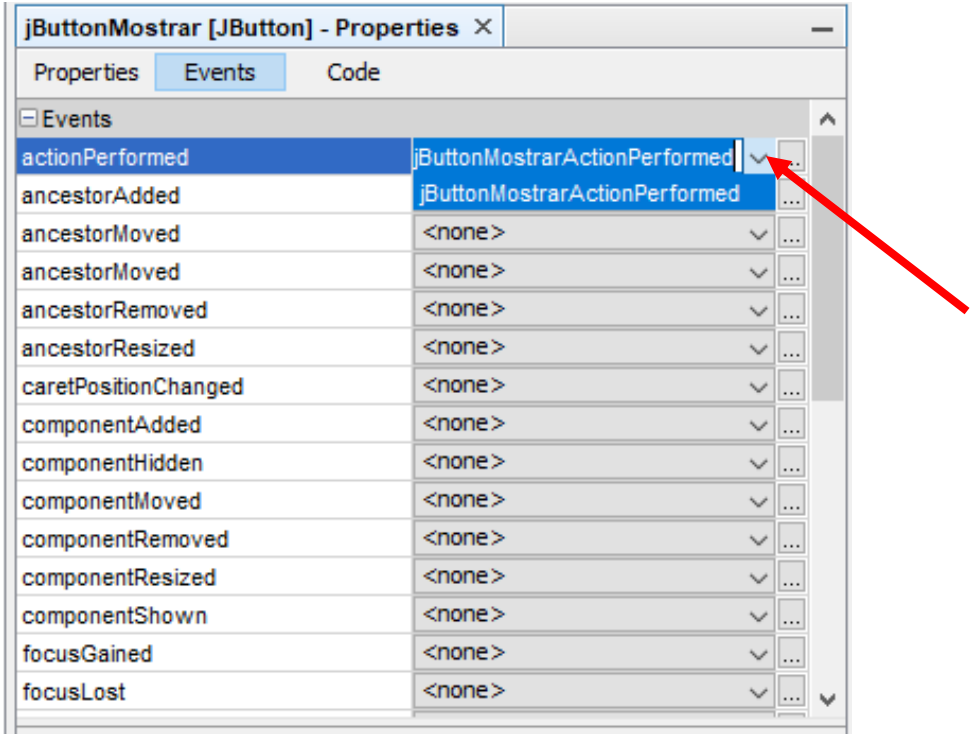
Al costat de les propietats del component seleccionat, està l'opció "*Events*" en la qual apareixen tots els diferents esdeveniments que es poden produir en el component seleccionat, com per exemple:
actionPerformed
keyPressed
keyReleased
mouseClicked
mouseMoved ...



3. ESDEVENIMENTS (*EVENT*)

Com crear un esdeveniment

Crear un manejador d'esdeveniments genèric (*actionPerformed*) és molt senzill.
Només cal desplegar l'acció al costat de *actionPerformed*



3. ESDEVENIMENTS (*EVENT*)

Com crear un esdeveniment

En fer-ho, *NetBeans* farà dues coses en el codi:
1. Crear un mètode per a manejar l'esdeveniment
2. Afegir al component un *ActionListener* associat al mètode anterior

Quan es pressione el botó s'executarà el mètode jButtonMostrarActionPerformed

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

```
jButtonMostrar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButtonMostrarActionPerformed(evt);
    }
```

3. ESDEVENIMENTS (*EVENT)*

Com crear un esdeveniment

Exemple

Volem que quan es pressione el botó aparega en una finestra el missatge amb el text que l'usuari haja introduït en el camp de text.

Per a fer-ho, introduïm el següent codi en el mètode manejador

JOptionPane.showMessageDialog(this, jTextFieldTexto.getText());

```
97
98      private void jButtonBotonActionPerformed(java.awt.event.ActionEvent evt) {
99          // TODO add your handling code here:
100         JOptionPane.showMessageDialog(this,jTextFieldTexto.getText());
101     }
```

PROGRAMACIÓ GRÀFICA

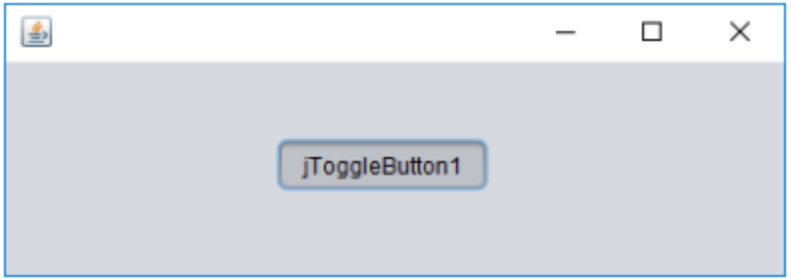
ÍNDEX DE CONTINGUTS

1. INTRODUCCIÓ
2. CREAR UNA APLICACIÓ *SWING*
3. ESDEVENIMENTS (EVENT)
4. COMPONENTS

4. COMPONENTS

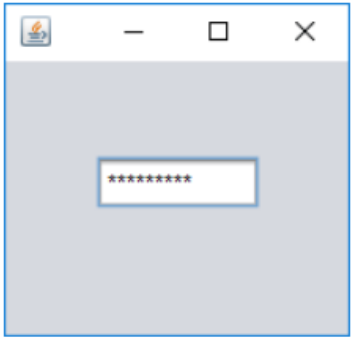
- Ja hem vist alguns dels components bàsics: etiqueta (*JLabel*), camp de text (*JTextField*) i botó (*JButton*)
- Però hi ha d'altres amb les seues particularitats dins el mateix tipus de components

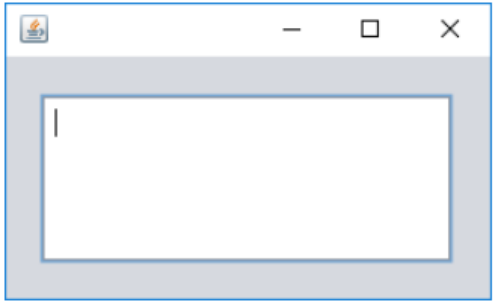
JToggleButton



4. COMPONENTS

JPasswordField i JTextArea





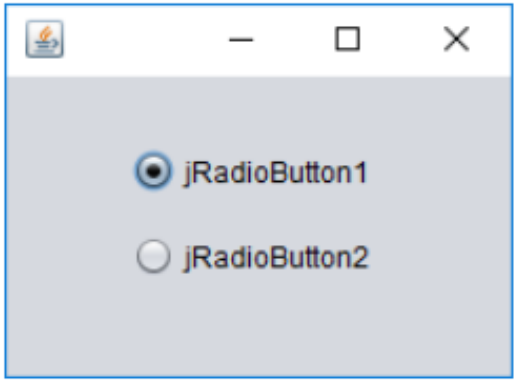
JCheckBox i JRadioButton

ButtonGroup

- Si es volen utilitzar diversos botons *JRadioButton* de manera que solament pot haver-hi un seleccionat, cal crear un *ButtonGroup* que continga els *JRadioButton*
- Per a crear un grup de botons afegirem el component *ButtonGroup* (no és visible)
- i després, des del panell de propietats de cada botó, podem seleccionar el grup al qual pertanyen

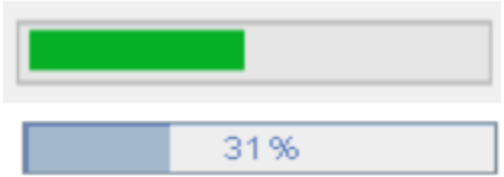
DIY

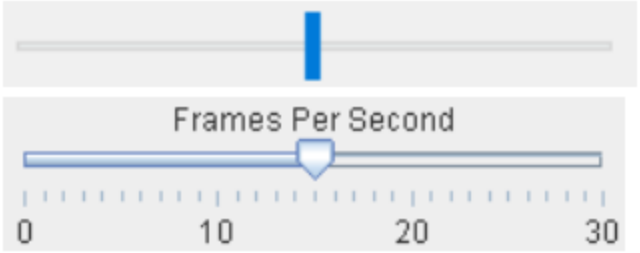
Inseriu un *ButtonGroup* en el vostre panell i assigneu-lo a dos *JRadioButton*.
Comproveu després que només un dels 2 radiobuttons pot estar seleccionat



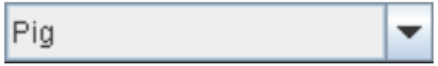
4. COMPONENTS

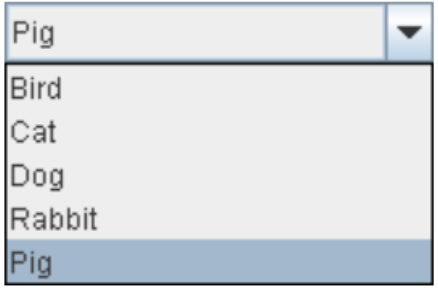
JProgressBar i JSlider





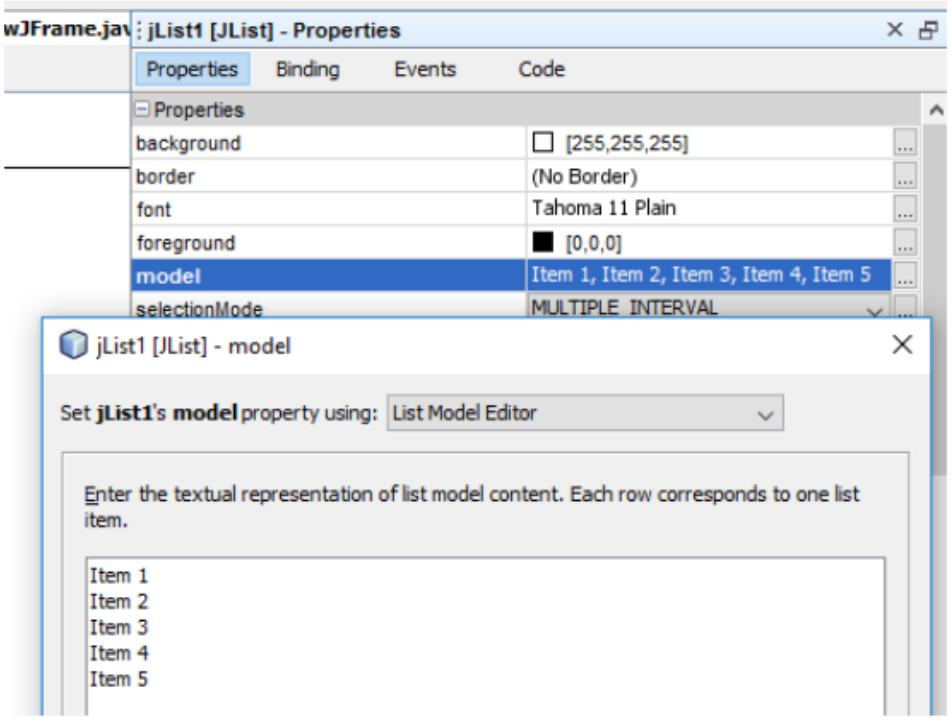
JComboBox





4. COMPONENTS

JList



4. COMPONENTS

JMenuBar, JMenu i JMenuItem

Els menús permeten crear llistes d'opcions i submenús amb més opcions que l'usuari pot utilitzar per a interactuar amb l'aplicació. Existeixen dos tipus principals:

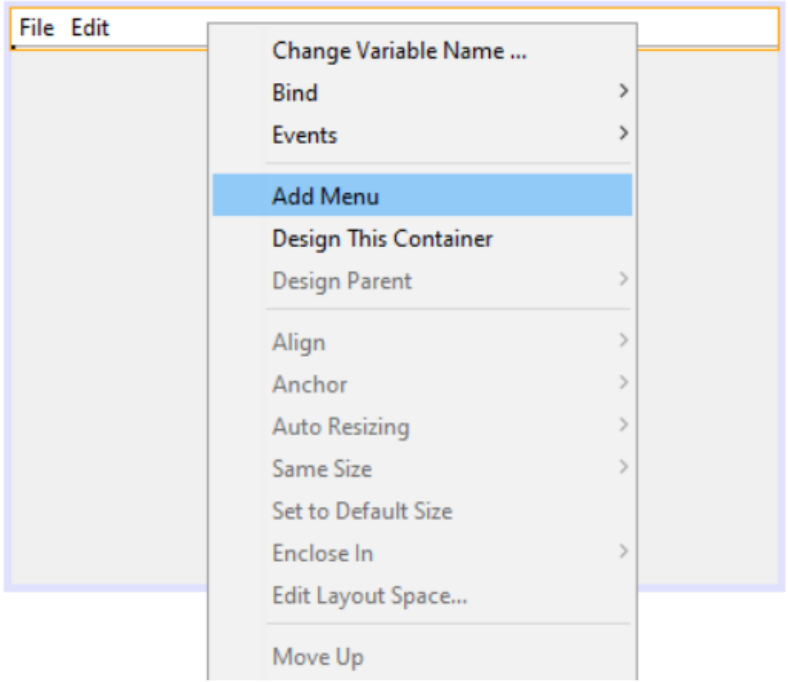
- **Barra de menú**: menú estàndard amb ítems i que pot contenir altres menús
- **Menú desplegable**: menú que es desplega quan l'usuari actua sobre ell

Exemple:
Primer, hem d'inserir en la nostra aplicació el component *JMenuBar*

JMenuBar, JMenu i JMenuItem

Exemple:

Podem afegir més menús si punxem sobre la barra de menú amb el botó dret i després en “Add Menu”



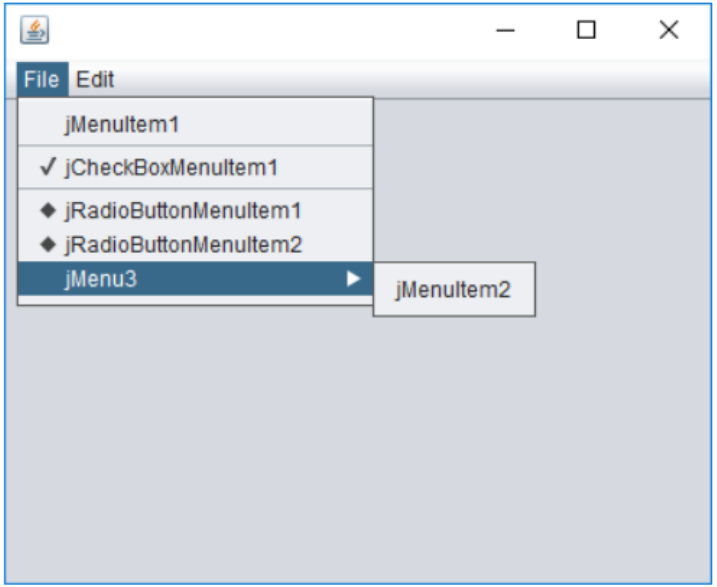
JMenuBar, JMenu i JMenuItem

Exemple:

A continuació inserim un component *JMenuItem* dins del menú *File*

Fem el mateix amb el component *JCheckBoxMenuItem* i *jRadioButtonMenuItem*

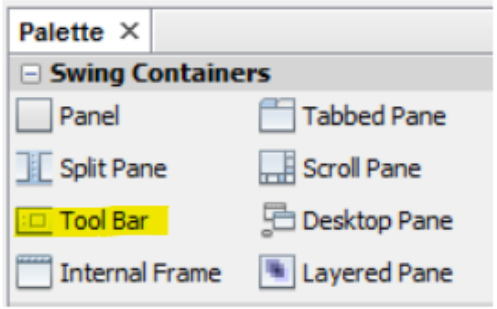
Finalment introduïm un component *JMenu* i dins d'ell un *JMenuItem*



JMenuBar, JMenu i JMenuItem

Exemple:

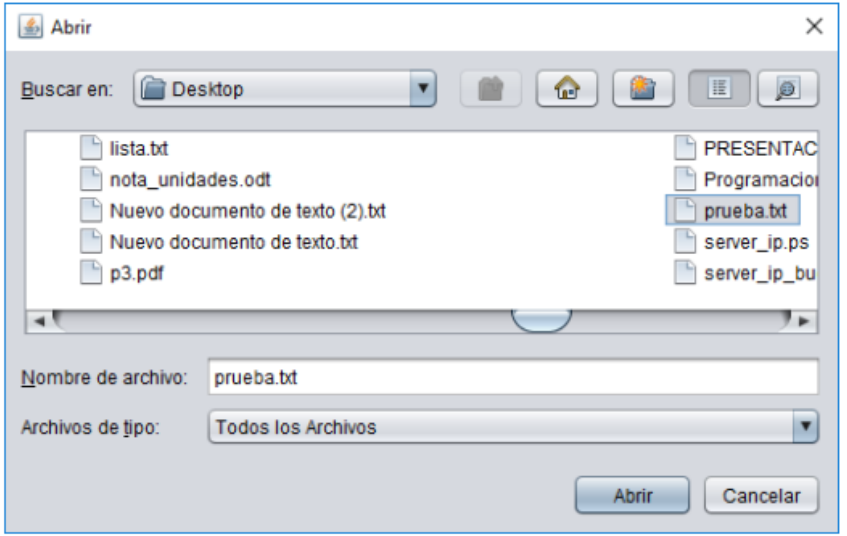
Per acabar inserirem la típica barra d'eines amb icones. Per a això seleccionem i arrosseguem l'element *JToolBar* del grup de contenidors



Només falta inserir botons dins de la barra, per a això seleccionem un botó i l'arrosseguem dins de la barra.

Selector de fitxers *JFileChooser*

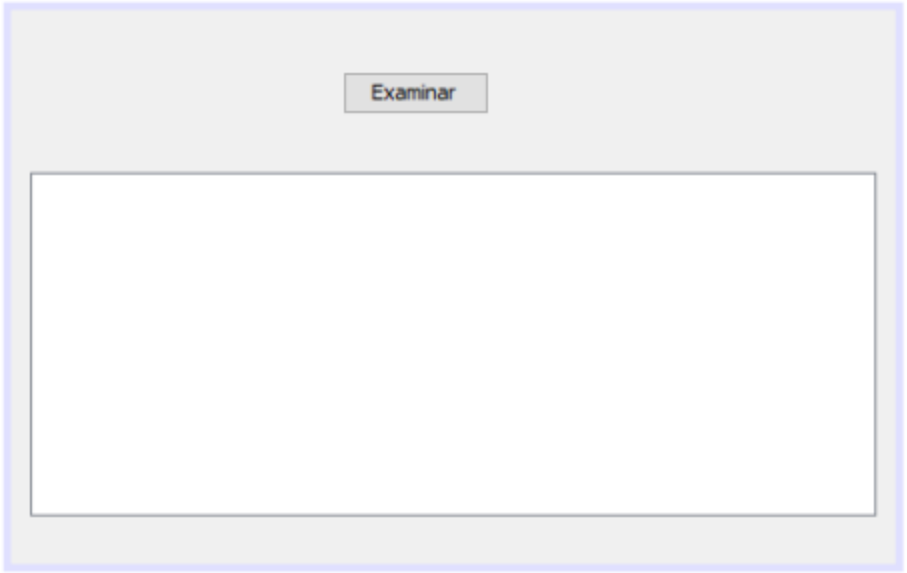
- Ens permet mostrar a l'usuari una finestra emergent de selecció de fitxer
- Per a utilitzar-ho simplement ho haurem de seleccionar i arrossegar a la finestra d'edició, **no dins del JFrame**
- És un element que una vegada inserit no és visible, apareixerà quan l'activem, però sí que apareix en el panell *Navigator*

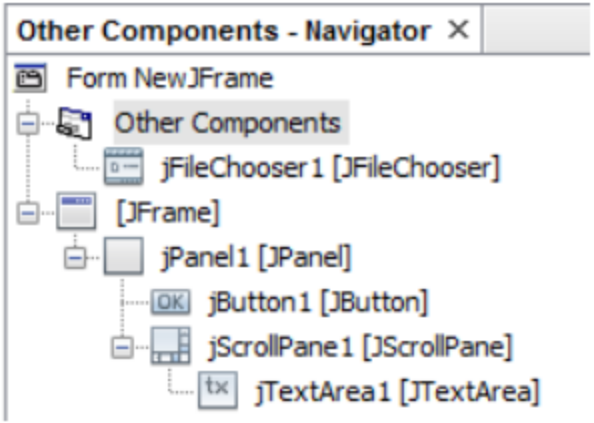


Tingues cura no ho inserisques dins del *JFrame*, llavors sí que apareixerà però no funcionarà correctament.

Selector de fitxers *JFileChooser*

Exemple: Mostrar el contingut d'un arxiu
Inserim el selector de fitxer, el panell, el botó i l'àrea de text (dins un *JScrollPane*)





Selector de fitxers *JFileChooser*

Exemple: Mostrar el contingut d'un arxiu

A continuació, inserim el codi necessari en l'esdeveniment del botó

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    int seleccion = jFileChooser1.showOpenDialog(this); // mostramos el selector

    if(seleccion == JFileChooser.APPROVE_OPTION) // Si no ha habido problemas
    {
        File fichero = jFileChooser1.getSelectedFile(); // Guardamos el fichero seleccionado en una variable del tipo File

        try
        {
            // Utilizamos la clase BufferedReader para leer el fichero
            BufferedReader in = new BufferedReader(new FileReader(fichero.getAbsolutePath()));

            String frase;

            frase = in.readLine(); // Leemos una liena del fichero

            while(frase != null) // Mientras leamos algo
            {
                jTextArea1.setText(jTextArea1.getText() + frase + "\n"); // Actualizamos el área de texto

                frase = in.readLine();
            }

            in.close(); // Al finalizar cerramos la lectura del fichero.
        }
        catch(IOException e)
        {
            JOptionPane.showMessageDialog(this,"Error al leer el fichero.");
        }
    }
}
```

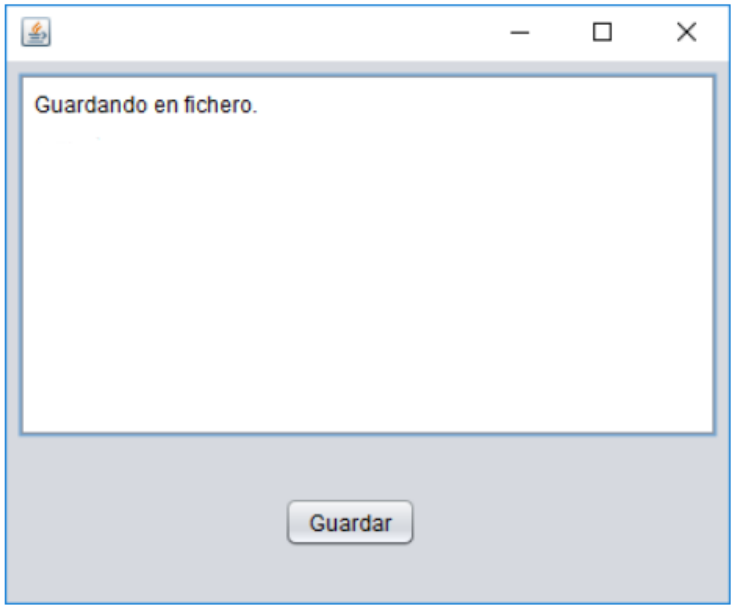
Cal destacar que la variable selecció pot ser igual a:

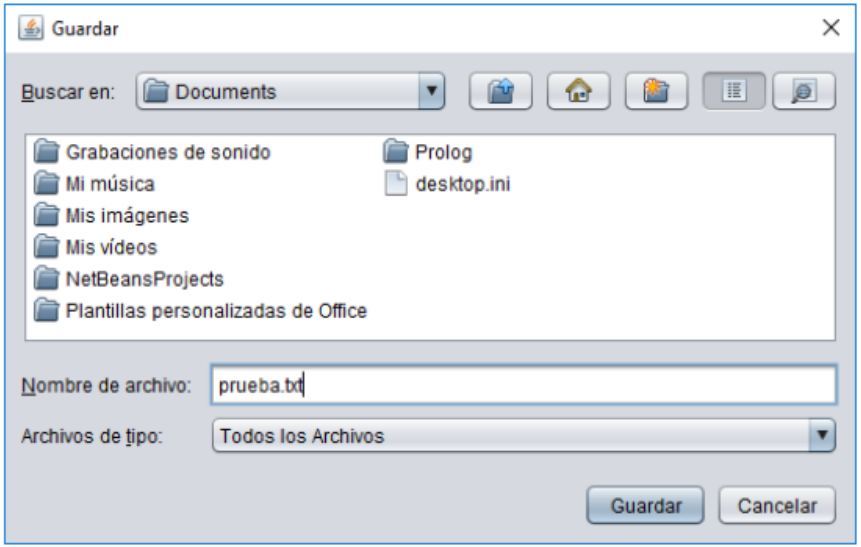
- *JFileChooser.CANCELL_OPTION* → Si l'usuari li ha donat al botó cancel·lar.
- *JFileChooser.APPROVE_OPTION* → Si l'usuari li ha donat al botó acceptar.
- *JFileChooser.ERROR_OPTION* → Si ha ocorregut algun error

4. COMPONENTS

Selector de fitxers *JFileChooser*

DIY: Guardem el contingut d'una caixa de text en un fitxer





4. COMPONENTS

Quadres de diàlegs *JOptionPane*

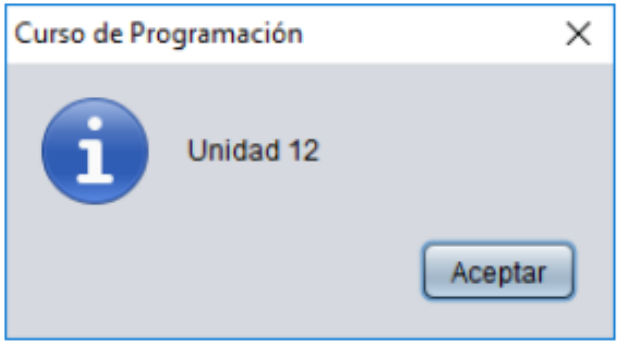
- Un quadre de diàlegs és una finestra que s'obri des d'una aplicació amb l'objectiu d'interactuar amb l'usuari
- Per a crear un diàleg, simple i estàndard, s'utilitza *JOptionPane*
- Hi ha diferents **tipus** de diàlegs: *showMessageDialog, showConfirmDialog* ...
- Per a tots ells els diferents tipus d'opció i les diferents icones són:

Tipus d'opció
- DEFAULT_OPTION
- YES_NO_OPTION
- YES_NO_CANCELL_OPTION
- OK_CANCELL_OPTION

Icones
- ERROR_MESSAGE
- INFORMATION_MESSAGE
- WARNING_MESSAGE
- QUESTION_MESSAGE
- PLAIN_MESSAGE (Sense icona)

Quadres de diàlegs *JOptionPane*

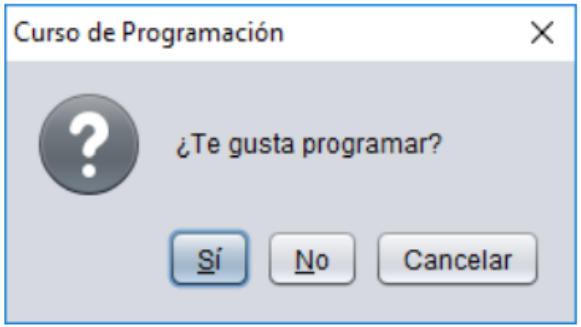
showMessageDialog → Mostra un diàleg amb un botó etiquetat "OK" o "ACCEPTAR"




```
JOptionPane.showMessageDialog(this,          // Objecte actual
                              "Unitat 12", // Text del missatge
                              "Curs de Programació", // Títol
                              JOptionPane.INFORMATION_MESSAGE); // Icona
```

Quadres de diàlegs *JOptionPane*

showConfirmDialog → Mostra un diàleg amb tres botons, etiquetats amb "SI" , "NO" i "Cancel·lar". Retorna un enter amb l'opció premuda (0,1,2)

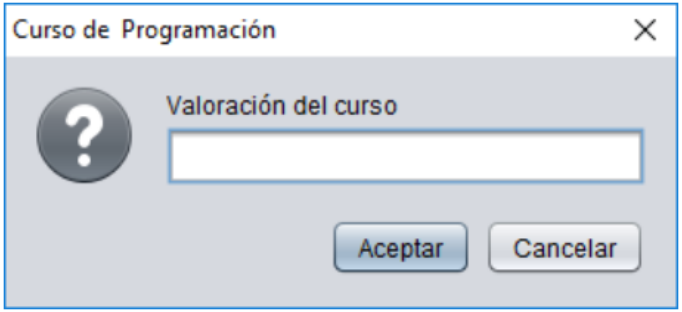



```
int opcion = JOptionPane.showConfirmDialog(this,  // Objecte actual
                                           "T'agrada programar?",  // Text del missatge
                                           "Curs de Programació",  // Títol
                                           JOptionPane.YES_NO_CANCELL_OPTION); // Icona
```

4. COMPONENTS

Quadres de diàlegs *JOptionPane*

showInputDialog → Mostra un camp de text perquè l'usuari teclege en ell o un *ComboBox* no editable, des del qual l'usuari pot triar una d'entre diverses cadenes. Retorna l'opció triada

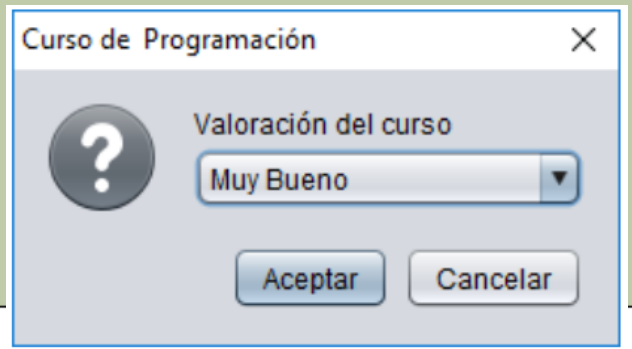



```
String valoracion = JOptionPane.showInputDialog(this,  // Objecte actual
                                                 "Valoració del curs",  // Text del missatge
                                                 "Curs de Programació ",  // Títol
                                                 JOptionPane.QUESTION_MESSAGE); // Icona
```

Quadres de diàlegs *JOptionPane*

showInputDialog → Mostra un camp de text perquè l'usuari teclege en ell o un *ComboBox* no editable, des del qual l'usuari pot triar una d'entre diverses cadenes. Retorna l'opció triada

```
String [ ] valors = {"Molt Bo", "Bo", "Dolent", "Molt Dolent"};
String opc = (String) JOptionPane.showInputDialog(this,  // Objecte actual
                                                  "Valoració del curs",  // Text del missatge
                                                  "Curs de Programació ",  // Títol
                                                  JOptionPane.QUESTION_MESSAGE,  // Icona
                                                  null,   // Paràmetre no utilitzat
                                                  valors,  // Vector de valors
                                                  valors[0]);  // Valor a mostrar per defecte
```

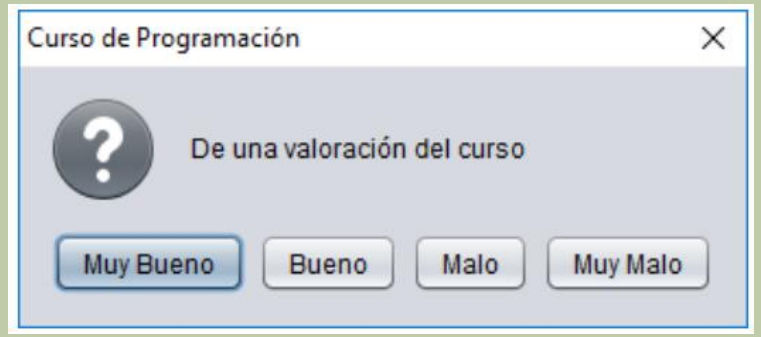


4. COMPONENTS

Quadres de diàlegs *JOptionPane*

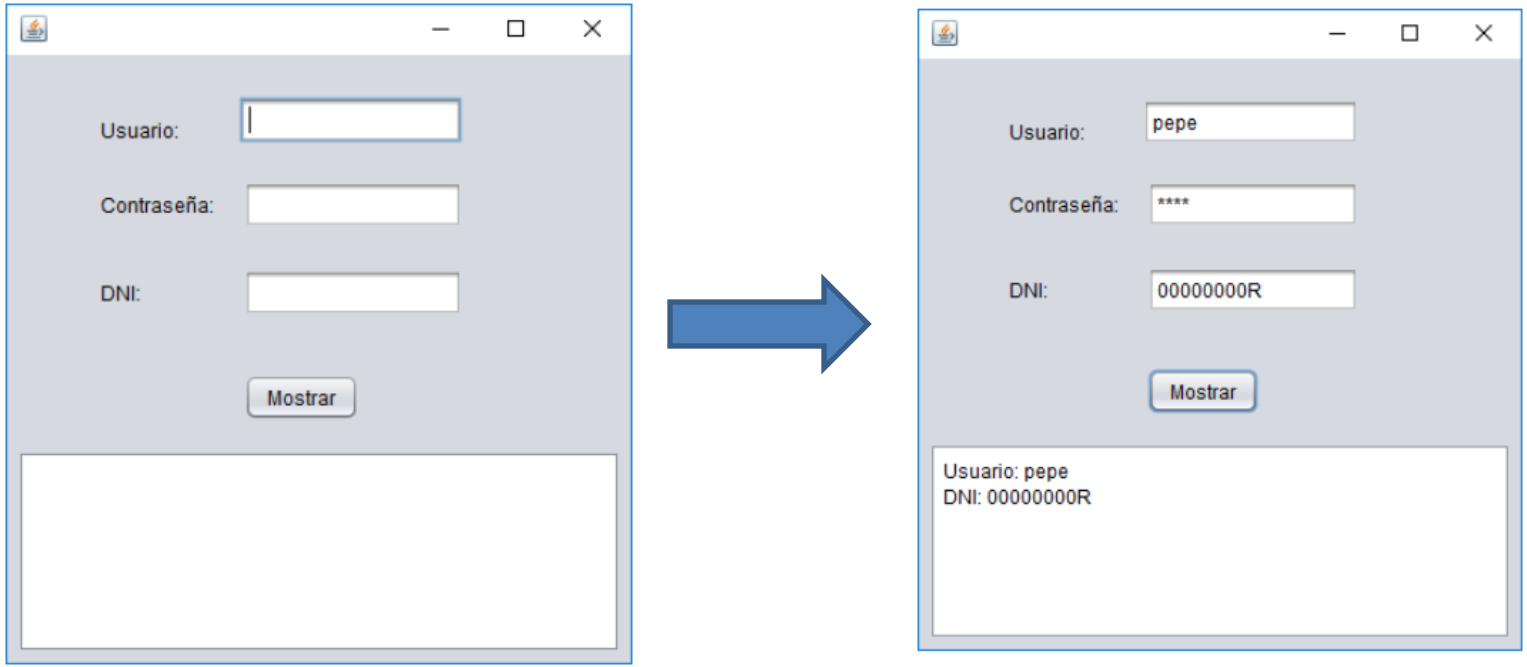
showOptionDialog → Amb aquest mètode podem canviar el text que apareix en els botons dels diàlegs estàndard i realitzar qualsevol tipus de personalització

```
String [ ] valors = {"Molt Bo", "Bo", "Dolent", "Molt Dolent"};
int opc = JOptionPane.showOptionDialog(this,  // Objecte actual
                                       "D'una valoració del curs",  // Text del missatge
                                       "Curs de Programació",  // Títol
                                       JOptionPane.YES_NO_CANCELL_OPTION,  // Opció
                                       JOptionPane.QUESTION_MESSAGE,  // Icona
                                       null,   // Paràmetre no utilitzat
                                       valors,  // Vector de valors
                                       valors[0]);  // Valor a mostrar per defecte
```



4. COMPONENTS

EXEMPLE: Crearem l'aplicació que en prémer en el botó "Mostrar" es mostre en el *textarea* l'usuari i el dni introduït (**és important que canvieu el nom per defecte de les variables**)



Fer Exercicis

PROGRAMACIÓ GRÀFICA

Autor:
Àngel Olmos Giner
segons el material de Carlos Cacho i Raquel Torres