

Ficha de Trabalho nº 2

GUIs com Swing

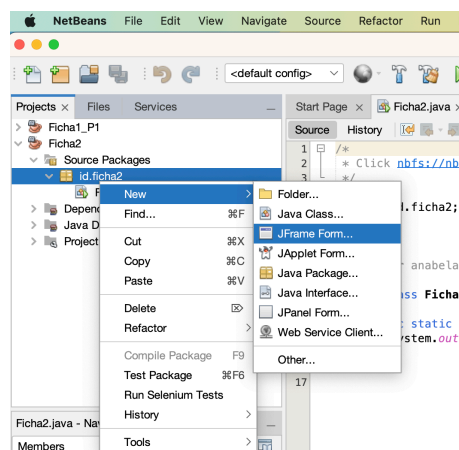
1. Bibliografia

<http://docs.oracle.com/javase/tutorial/uiswing/>

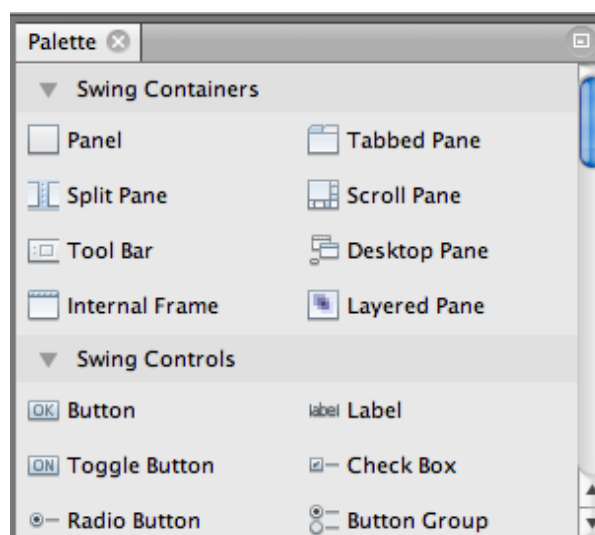
2. Primeiro GUI

2.1 Criar um Projecto no Netbeans

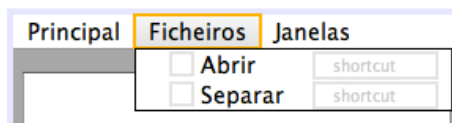
- Inicie o Netbeans e faça: **File - New Project - Java with Maven - Java Application**
- Grave com o nome: **Ficha2**
- Na package **ficha2**, use o botão direito do rato para criar um **JFrame Form** (Frame que vai conter todos os componentes do interface). Dê-lhe o nome de **Frame**



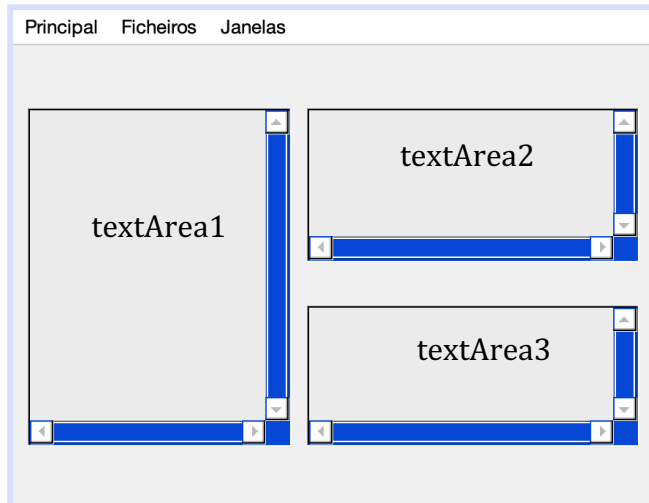
Os componentes que podem ser colocados nesta **Frame** encontram-se na janela **Palette**



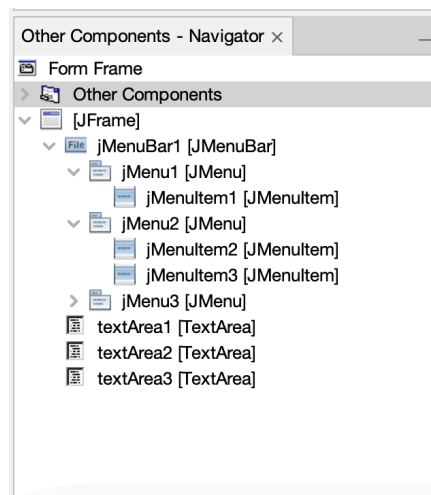
Escolha um objeto da categoria **Swing – Menus – Menu-Bar** e arraste-o para cima da **Frame**. Altere a barra de menus. Acrescente um **Menu** novo (Janelas). No menu Principal acrescente um **Menu Item** (Sair), no menu Ficheiros acrescente dois **Menu Item** (Abrir e Separar):



Adicione três elementos do tipo **Text Area** à sua Frame:



A estrutura e o nome dos componentes que vão sendo adicionados pode ser visualizada/alterada no **Navigator**:



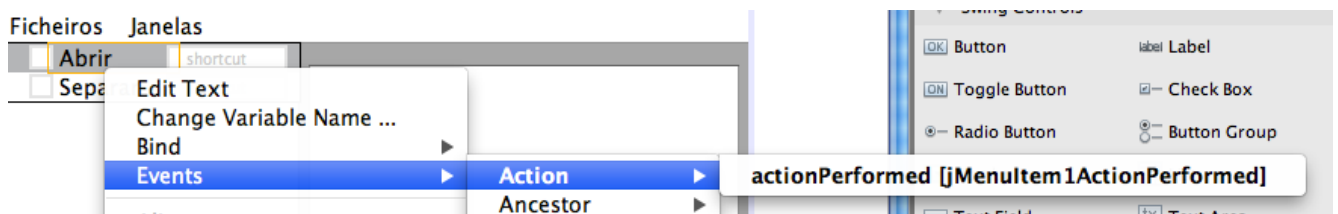
No ficheiro **Ficha2.java**, implemente uma função que leia o ficheiro **alunos.txt** e coloque todo o seu conteúdo numa String (use o método *append* da variável *StringBuilder* para ir juntando as várias linhas lidas):

```
static String ler_ficheiro(String nomeF){
    String linha;
    StringBuilder texto = new StringBuilder(); //permite concatenar várias String
    Scanner input = new Scanner(new FileInputStream(nomeF));

    while ((input.hasNextLine())) {
        ..... //COMPLETAR
        texto.append(linha).append("\n");
    }
    input.close();
    return texto.toString(); //converter StringBuilder para String
}
```

Vamos associar esta função à opção do menu **Ficheiros-Abrir** do seu interface.

Posicione-se no ficheiro **Frame.java** e na opção **Abrir** do menu **Ficheiro** e escolha, com o botão direito do rato, a opção **Events – Action – actionPerformed**:



Na função que é criada deve chamar a função **ler_ficheiro** que implementou em **Ficha2.java** e o texto devolvido pela função deve ser colocado na **Text Area** de nome **textArea1** usando o método **setText** (corrija os erros com ajuda do Netbeans):

```
private void jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt) {
    String texto;
    texto=Ficha2.ler_ficheiro("alunos.txt");
    textArea1.setText(texto);
}
```

Corrija os erros usando um *block Try – Catch*

Associe agora a ação de sair da aplicação à opção **Principal-Sair**



Menu SAIR: A instrução para sair da aplicação que deve colocar no corpo da função é: `System.exit(0);`

Na função *main* coloque o código para executar a janela **Frame**:

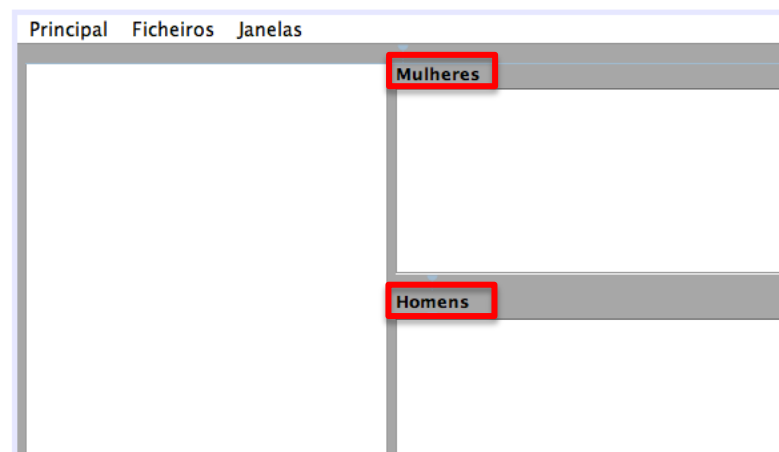
```
Frame app = new Frame();
app.setVisible(true);
```

Corra (Run) a sua aplicação e teste as duas ações programadas nas opções dos menus:
Ficheiros-Abrir e **Principal-Sair**

No ficheiro **Ficha2.java**, implemente uma função que separe o ficheiro **alunos.txt** em dois ficheiros **homens.txt** e **mulheres.txt** (ver Ficha1)

```
static void separa_ficheiro(String nomeF){
    String linha;
    String []campos;
    Scanner input = new Scanner(new FileInputStream(nomeF));
    BufferedWriter mulheres = new BufferedWriter(new FileWriter("mulheres.txt"));
    BufferedWriter homens = new BufferedWriter(new FileWriter("homens.txt"));
    while ((input.hasNextLine())) {
        ...
    }
    mulheres.close();
    homens.close();
    input.close();
}
```

Coloque um **Label** associado a cada *Text Area* tal como indicado na Figura seguinte:



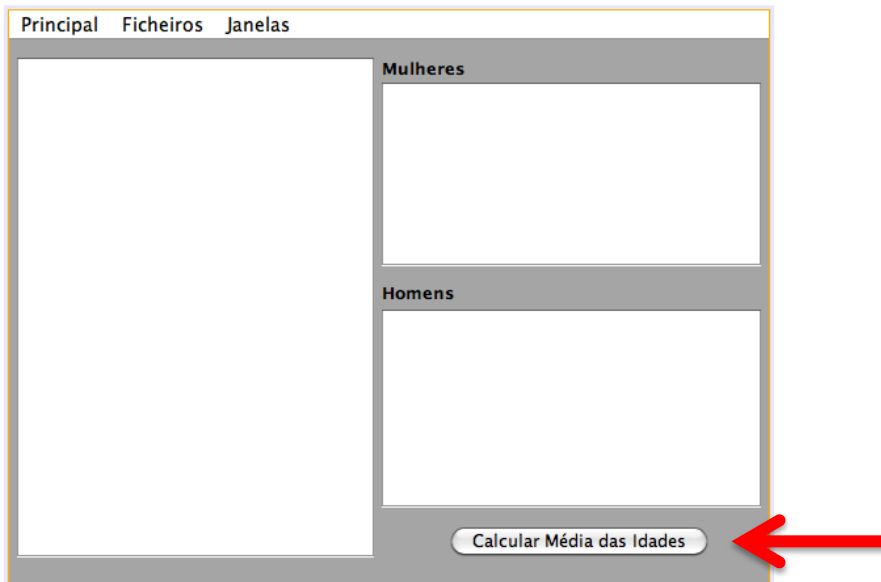
A opção do menu **Ficheiros-Separar** do seu interface deve fazer as seguintes tarefas:

- Chamar a função **separa_ficheiro** para criar os ficheiros **homens.txt** e **mulheres.txt**
- Chamar a função **ler_ficheiro** para obter o conteúdo do ficheiro **homens.txt** e colocar esse conteúdo na *Text Area* de nome *textArea2*
- Chamar a função **ler_ficheiro** para obter o conteúdo do ficheiro **mulheres.txt** e colocar esse conteúdo na *Text Area* de nome *textArea3*
- Corrigir os erros com *Block Try -- Catch*

Associe estas ações ao evento da opção do menu **Ficheiros-Separar**.

Corra a aplicação e teste a funcionalidade do menu **Ficheiros-Separar**.

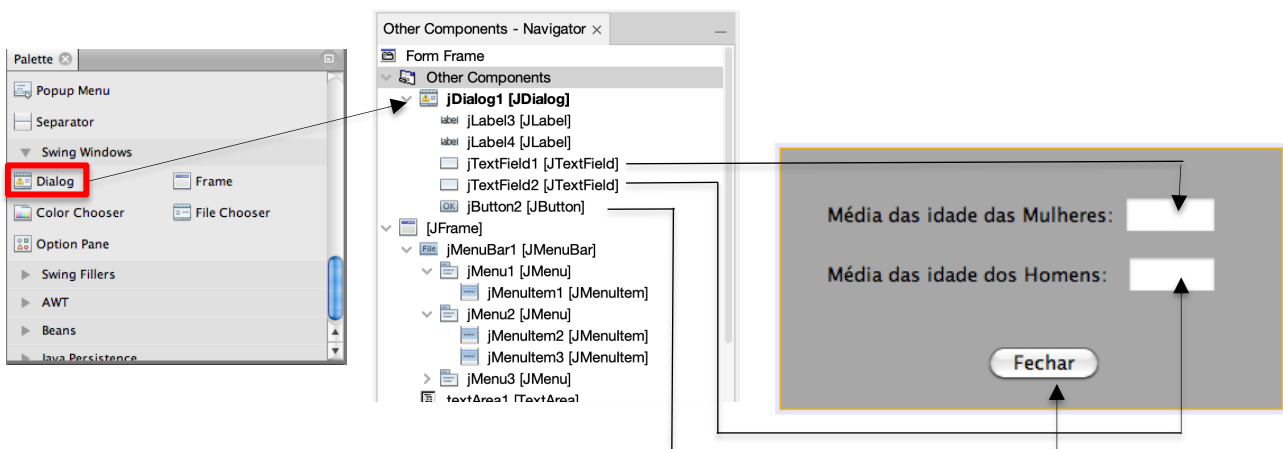
Adicione um **botão** (*Button*) à sua interface. Ao carregar nesse botão o utilizador terá acesso à informação sobre as médias das idades dos homens e das mulheres.



Para calcular a média das idades de cada género tem de implementar uma função que leia o ficheiro **alunos.txt** e execute esse cálculo. No ficheiro **Ficha2.java** implemente uma função que receba como argumento o género ("M" ou "F") e devolve a média das idades das pessoas desse género encontradas no ficheiro. Use a função `Integer.parseInt` para converter `String` para `int`:

```
static double calcula_MediaIdade(String genero){
    ...
    Scanner input = new Scanner(new FileInputStream("alunos.txt"));
    while ((input.hasNextLine())) {
        ...
    }
    input.close();
    return ...;
}
```

Para visualizar esta informação quando o utilizador clica no botão, adicione a partir da Palette, um **Dialog** e localize-o em *Other Components*. Formate o seu **JDialog** com dois **Label**, dois **TextField** e um **Button**, para que fique com o seguinte aspeto:



Regresse à sua janela principal e associe um evento ao botão “**Calcular Média das Idades**”:

- Chamar a função `Ficha2.calcula_MediaIdade` com o argumento “**F**” e guardar a média calculada numa variável
- Chamar a função `Ficha2.calcula_MediaIdade` com o argumento “**M**” e guardar a média calculada numa variável
- Mostrar os valores calculados no **jDialog** criado anteriormente

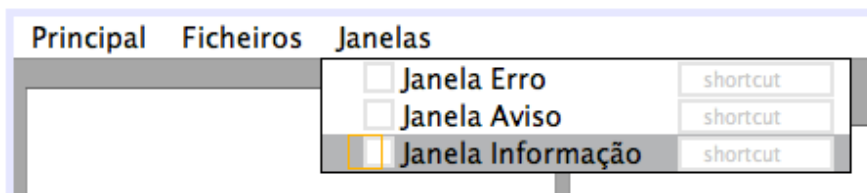
```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    double homens=0,mulheres=0;
    mulheres = Ficha2.calcula_MediaIdade("F");
    homens = Ficha2.calcula_MediaIdade("M");

    jDialog1.setSize(500, 300);
    jDialog1.setLocation(200, 200);
    jTextField1.setText(Double.toString(mulheres));
    jTextField2.setText(Double.toString(homens));
    jDialog1.setVisible(true);
}
```

Associe um evento ao botão “**Fechar**” do `jDialog`:

```
jDialog1.setVisible(false);
```

Adicione 3 opções ao menu Janelas:



É possível criar janelas de erro, aviso, informações, entre outras, da seguinte forma:

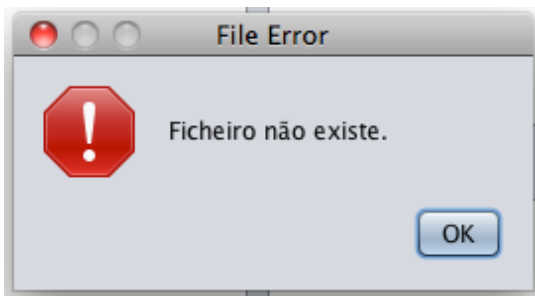
Associe um evento à opção do menu **Janela Erro** e coloque este código na função:

```
JOptionPane.showMessageDialog(this,
    "Mensagem de Erro.",
    "Error",
    JOptionPane.ERROR_MESSAGE);
```

Repita esta operação para as outras duas opções do menu Janelas, alterando o código de forma que a janela fique personalizada em conformidade:

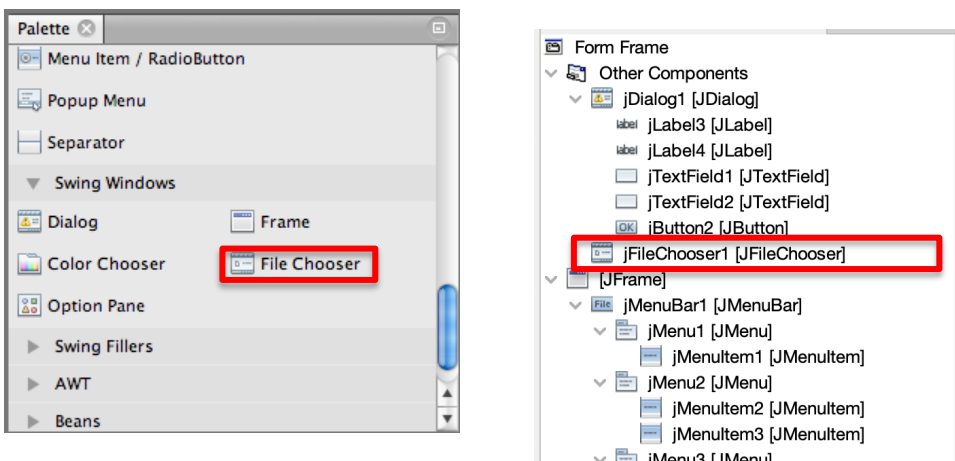


Pode usar este tipo de janelas para informar situações de erro, por exemplo, quando um ficheiro não existe. Altere o seu código para que quando é seleccionada a opção **Ficheiros-Abrir** ou **Ficheiros-Separar** surja a seguinte janela:



Outras funcionalidades que pode adicionar ao seu interface:

Escolha da Palette um **FileChooser** e coloque-o em *Other Components*:



Altere o evento da opção **Ficheiros-Abrir** de forma a que o ficheiro “alunos.txt” seja seleccionado usando o **FileChooser**:

- Visualizar o **FileChooser**:

```
jFileChooser1.showOpenDialog(this);
```

- Obter o nome do ficheiro escolhido:

```
String nomeF= jFileChooser1.getSelectedFile().getAbsolutePath();
```

- Execute a aplicação e selecione a opção Abrir para ver se o **FileChooser** funciona corretamente.