

# Criação de um cadastro em Swing

## Java NetBeans

---

### Aplicações visuais e componentes

Giovanni Camargo Silva

[giovanni@unipacto.com.br](mailto:giovanni@unipacto.com.br)

GONÇALVES, Edson. Dominando o NetBeans, Rio de Janeiro, editora Ciência Moderna Ltda, 2016.

Esse material é totalmente baseado no livro Dominando NetBeans, com as atualizações para a versão portuguesa 6.1 do NetBeans, os exemplos serão os mesmos do livro, com a linguagem mais simples possível para o maior entendimento do aluno. Propõe ao aluno a compreensão e ao desenvolvimento de interfaces gráficas com as classes: Swing e AWT, manipulação de objetos, componentes e tratamento de eventos.

## ÍNDICE

DESENVOLVENDO APLICAÇÕES VISUAIS NA IDE-----	3
APLICAÇÃO NO DESIGN VIEW-----	3
TRABALHANDO COM COMPONENTES-----	5
INTRODUÇÃO AOS COMPONENTES DA BIBLIOTECA SWING -----	6
MAS O QUE É SWING? .....	6
TRABALHANDO COM COMPONENTES SWING -----	6
CRIANDO UM MENU .....	6
AGRUPANDO ITENS DE MENU .....	9
ADICIONANDO ÍCONES AO MENU .....	9
CRIANDO MENUS FLUTUANTES .....	9
COMENTANDO O CÓDIGO DO MENU FLUTUANTE -----	10
UMA OUTRA FORMA DE UTILIZAR O MENU POP-UP-----	10
TRABALHANDO COM ALGUNS COMPONENTES SWING-----	11
COMPONENTES BÁSICOS DE INTERFACE GRÁFICA -----	11
O PROJETO -----	11
JFRAME E JPANEL .....	11
JLABEL .....	11
JTEXTFIELD .....	12
CAIXA DE COMBINAÇÃO .....	12
JFORMATTEDTEXTFIELD .....	12
EXERCÍCIOS -----	14
JBUTTON .....	15

## DESENVOLVENDO APLICAÇÕES VISUAIS NA IDE

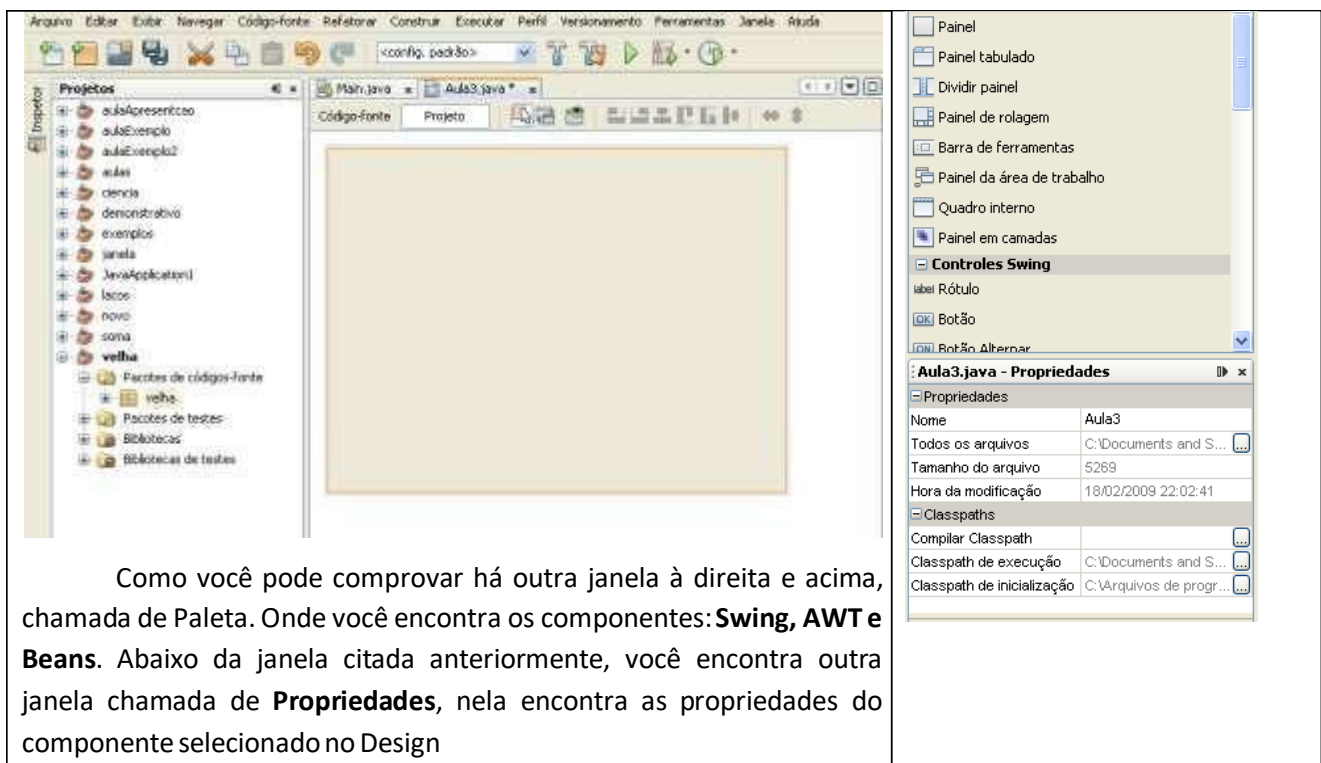
Com o NetBeans familiarizado se tem suporte suficiente para trabalhar com aplicações mais complexas, utilizando-se janelas.

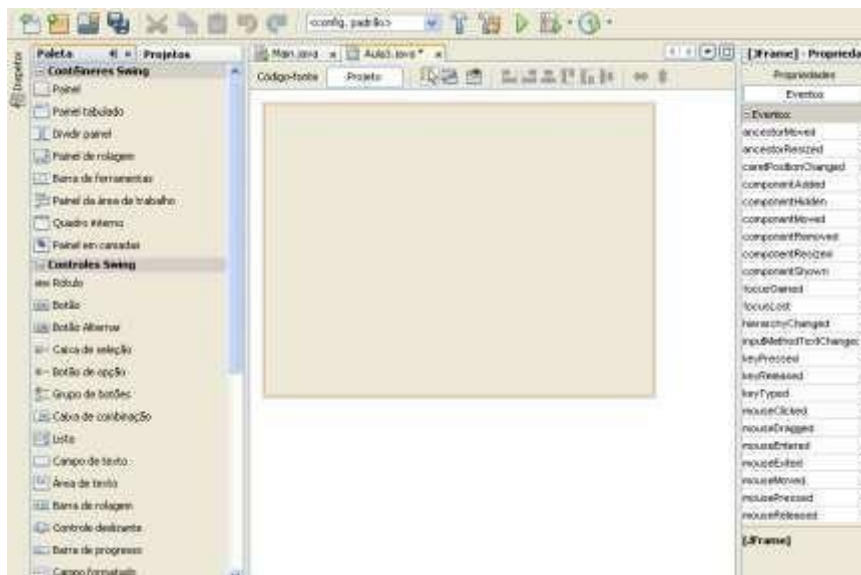
### APLICAÇÃO NO DESIGN VIEW

Graças ao seu editor **WYSIWYG** (what you see is what you get) – O que você vê é o que você começa. O Design View tem um modo fácil de desenvolver uma interface na IDE NetBeans.

Para criar um aplicativo visual clique no ícone Novo Arquivo ou vá ao menu Arquivo e clique em novo Arquivo.

Na caixa de diálogo **Novo Arquivo**, selecione o item **Java Formulário**, na próxima interface do diálogo você tem a opção de nomear a classe a ser criada por fim clique em **finalizar**. Aparecerão novas janelas com um editor visual para desenvolver seu design



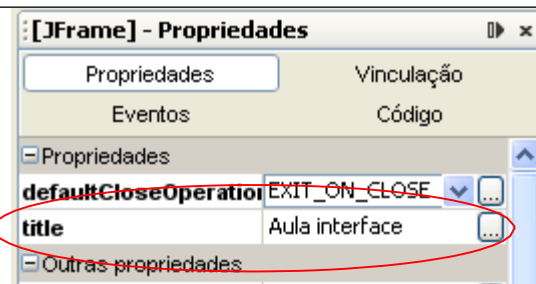


Na lateral direita, você encontra as janelas **Inspetor** e **Navegador**. A primeira janela mostra os componentes encontrados no seu programa visual. Em Navegador você pode navegar por entre declarações e métodos encontrados no desenvolvimento do aplicativo.

É claro que essas janelas não têm uma disposição fixa, você pode arrastá-las na posição que desejar. Para fazer isso, clique na paleta onde se encontra o nome da janela e arraste livremente para a posição desejada. Ao iniciar um arraste, você verá uma borda ao redor com Outra cor, indicando a movimentação da janela. Para uma melhor compreensão, mostro na imagem ao lado a disposição da janela que pode ser usada demonstrando a flexibilidade que sua IDE pode ter.

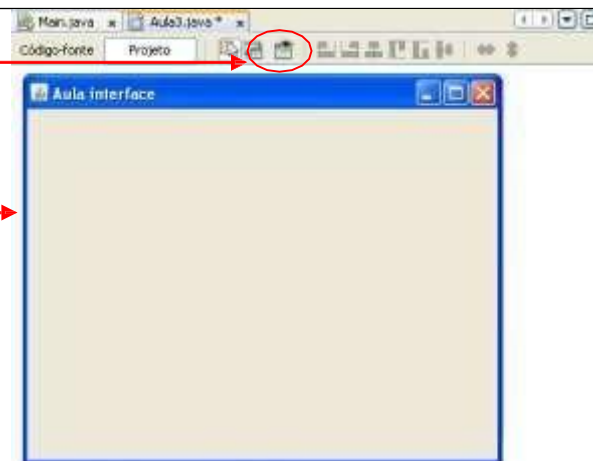
Ao escolher **JFrame Form**, você teve, através do assistente, uma pequena janela com alguns elementos pré-configurados logo de início.

Para que a adaptação ao Design View seja rápida, você vai começar com a mudança de título. Clicando em **JFrame** na janela **Inspetor**, vá até a janela **Propriedades**, ao lado da palavra **title**, digite o nome do aplicativo exemplo: 'Aula interface', e pressione **<Enter>**.

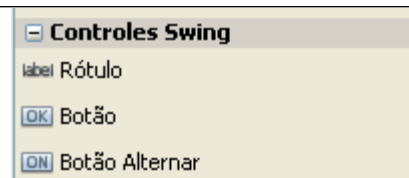




Para você visualizar como está ficando sua janela, clique no ícone **Visualizar Desenho**, como mostra a Figura

No qual você terá uma janela semelhante à mostrada na imagem. →



Como existe também uma janela de componentes, onde no desenvolvimento do seu aplicativo visual, você pode adicionar outros itens que não sejam os já existentes vindos do seu assistente, será adicionado ao aplicativo um **botão**, encontrado na janela **Paleta** como mostra a imagem a seguir:



<p>Coloque o <b>botão</b> mais ou menos ao centro de sua janela, como mostra a imagem.</p>	<p>Para alterar o texto desse componente, você pode ir até sua propriedade, na janela <b>Propriedades</b>, e alterara propriedade <b>text</b>, ou dar um duplo clique sobre o botão, mude o texto para: Fechar a Janela.</p>
	

Falta, antes de rodar o aplicativo, adicionar um evento ao botão para que ele feche a janela quando for clicado.

Na janela **Propriedades**, clique em **Eventos** e no evento **action Performed**, dê um clique ao lado e depois clique sobre o evento (onde está escrito **actionPerformed**). Se preferir, com o botão direito do mouse, ou o inverso do qual você usa, vá até o item **Eventos** e em **Action**, no seu menu de contexto, clique em **actionPerformed**.

Você será levado ao código, em Source, no local onde foram geradas linhas de códigos no seu aplicativo. Adicione o código em destaque a seguir:

```
private void JButton1ActionPerformed(java.awt.event.ActionEvent evt){  
    System.exit(0);  
}
```

**DICA:** *Você pode acionar vários eventos, entre outros códigos, no editor pressionando as teclas **Ctrl + Barra de espaço**.*

Feita essas alterações, salve usando a sequência no teclado **Ctrl+S** e vá ao botão **Run Main Project** e dê um clique (ou use a tecla **shift + F6**).

Como resultado você terá uma janela. Clique no botão criado **Fechar a Janela** para fechar a aplicação.



## TRABALHANDO COM COMPONENTES

Até aqui, você aprendeu como iniciar um aplicativo e criar uma interface amigável em seu programa criado na IDE NetBeans.

Esses aplicativos criados foram desenvolvidos em uma interface de usuário gráfica (GUI. Graphical User Interface), no qual lhe proporciona a visualização em uma janela em que você pode manipular tanto com mouse como com o teclado.

Agora você será levado a conhecer um pouco mais da interface gráfica do Java usando seus

componentes mais comuns.

Desse ponto em diante, você também deterá mais prática na utilização do Matisse na IDE.

## INTRODUÇÃO AOS COMPONENTES DA BIBLIOTECA SWING

No início da linguagem Java, uma biblioteca de classes chamada de AWT (Abstract Window Toolkit — caixa de ferramentas de janelas abstratas) para programação GUI avançada, fazendo assim desde o princípio parte do JDK (Java Development Kit). Essa biblioteca funcionava bem em aplicações mais simples, mas infelizmente ela não atendia a criação de uma interface mais complexa, rica em elementos gráficos. O pior ocorria quando menus, barras de rolagem e campos de texto, demonstravam pequenas diferenças de comportamentos em diferentes plataformas.

Em 1997 é anunciado o Java Foundation classes, ou JFC, que substituiu e incluiu o AWT. A parte principal do JFC é o jogo novo de componentes de interface de Usuário que são muito mais complexos, flexíveis e portáteis. Estes componentes novos são chamados de SWING.

A JFC não inclui em seus recursos apenas componentes Swing, mas também uma API de acessibilidade, uma API2D e etc.

NOTA:

*O Swing não é um substituto completo do AWT só porque simplesmente fornece uma interface de usuários mais completa e complexa. A arquitetura básica do AWT ainda permanece inalterada.*

## MAS O QUE É SWING?

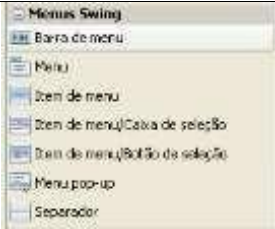
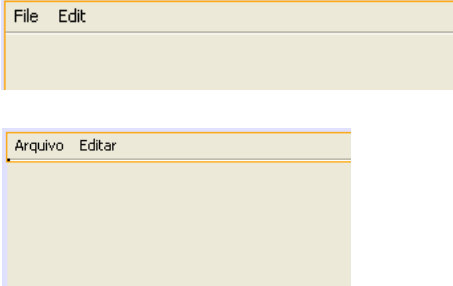
O Swing é o kit de ferramentas GUI que a Sun Microsystems desenvolveu para habilitar o empreendimento de desenvolver em Java. Os programadores podem usar Swing para criar amplas aplicações de Java com uma larga escala de componentes poderosos. Além disso, você pode estender facilmente ou pode modificar estes componentes para controlar seu aparecimento e comportamento. O nome representa a escolha da colaboração de seus desenhistas quando o projeto foi elaborado em 1996, Swing é de fato parte de uma família maior de produtos Java conhecida como Java Foundation Classes (JFC), que incorporam muitas das características da Netscape Internet Foundation Classes (IFC), como também aspectos de design da divisão IBM's Taligent e Lighthouse Design.

## TRABALHANDO COM COMPONENTES SWING.

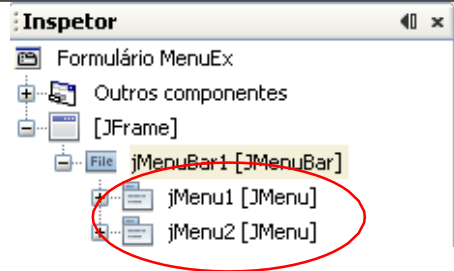
A partir de agora você irá trabalhar com um conjunto de componentes da classe Swing, incluindo as que foram mostradas anteriormente. Inicie um novo projeto. Crie uma nova classe, como feito anteriormente.

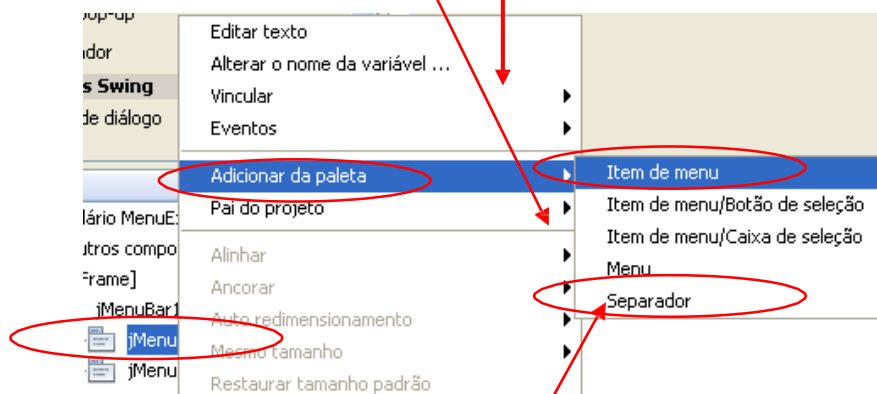
## CRIANDO UM MENU

Uma curiosidade no desenvolvimento de aplicativos usando O NetBeans é que você tem duas opções para o desenvolvimento inicial de um aplicativo com menu. O primeiro é em Categorias no item Sample Forms dentro do Java GUI Forms e selecionando em **Tipo de Arquivos** o item **Aplicação**, que já vem configurado com um menu inicial. O outro é a opção que você já escolheu, que não tem nenhum menu pré configurado. Os elemento de interface de usuário, são os menus suspensos, tão comuns em ambientes gráficos. Uma barra de menu no topo da janela contém os nomes dos menus suspensos. Clicando em uma opção através de seus nomes, este se abre e mostra seu conteúdo com itens do menu e dos sub menus. Ao clicar em um item desse menu, o menu se fecha e uma mensagem é enviada ao aplicativo.

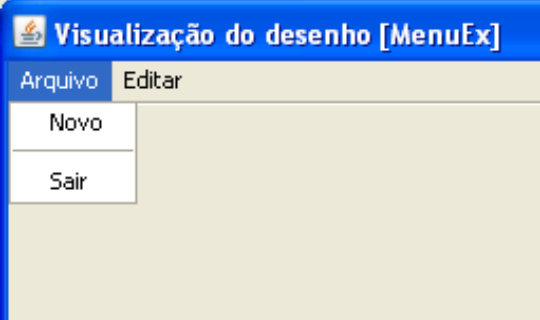
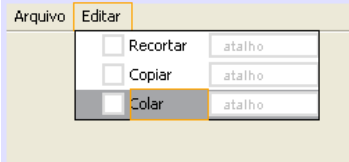
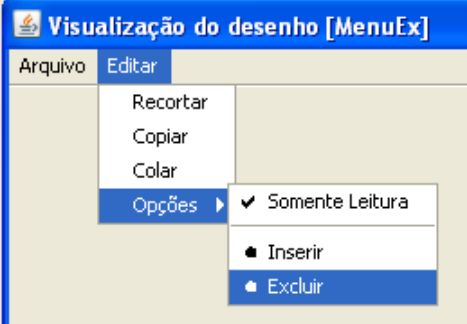
<p>Para criar um menu no NetBeans. vá na janela <b>Paleta</b> e selecione o componente Barra de Menu.</p> <p>Logo depois dê um clique no design do seu aplicativo no local em que desejar.</p>	
<p><b>O Design irá ser alterado, você notará a palavra File e Edit na parte superior, e abaixo um traço de separação.</b></p> <p>Dê um duplo clique e digite Arquivo em File, tecle Enter para confirmar, tendo alterado o texto do menu, na janela <b>Propriedades</b>, no campo <b>mnemonic</b>, digite a letra A, Faça o mesmo para Edit digite Editar, e no campo <b>mnemonic</b> digite a letra E.</p>	

Ao clicar em **Visualizar Desenho**, a letra A da palavra Arquivo do menu será sublinhada juntamente com a letra E de Editar, toda a vez que você mantiver pressionada a tecla auxiliar Alt e uma dessas teclas acionará o menu correspondente.

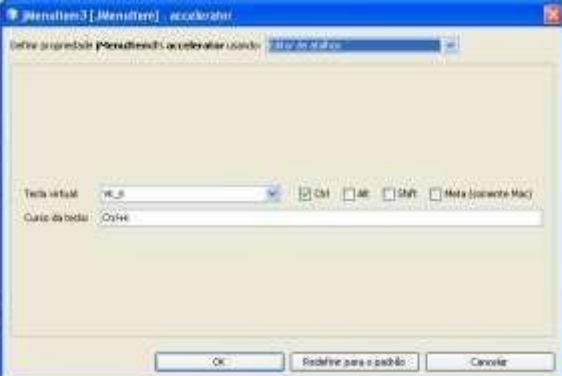
<p>Voltando ao design do seu menu, você acrescentar <b>item de menu</b>. Para fazer isso, na janela Inspetor, clique com o botão direito do mouse sobre o componente <b>jMenu1</b> ou <b>jMenu2</b>. No menu de contexto, vá até a opção Adicionar da <b>Paleta</b> e clique em <b>Item de Menu</b>, como mostra a imagem abaixo:</p>	
---	---



Novamente com o botão direito do mouse sobre o componente **jMenuBar1**, Selecione em, **Adicionar da Paleta**, o item **Separador** no menu de contexto. Repita novamente o ato do botão direito e adicione mais um componente **Item do menu**.

<p>Botão direito do mouse sobre o componente para editar o texto desse item de menu, na janela <b>Inspetor</b> selecione o componente <b>jMenuItem1</b> e na janela <b>Propriedades</b> altere a propriedade <b>text</b> digitando a palavra Novo. Selecionando o componente recém criado, <b>jMenuItem3</b> na janela <b>Inspetor</b>, altere na janela <b>Propriedades</b> a propriedade <b>text</b> digitando a palavra Sair. Clique no ícone <b>Visualizar Desenho</b> para visualizar como está ficando seu menu.</p>	
<p>No menu Editar, ao lado do menu <b>Arquivo</b>. Coloque mais três itens de menu e altere seus textos para Recortar, Copiar e Colar respectivamente.</p>	
<p>Adicione mais um componente <b>jMenu</b> clicando sobre o componente <b>jMenu2</b> e altere na sua <b>Propriedade</b>, no campo <b>text</b>, digitando a palavra Opções. Adicione o componente <b>Item de Menu/ Caixa de Seleção</b> clicando sobre o componente</p> <p>Adicione o componente <b>Separador</b> também clicando sobre o componente <b>jMenu3</b>. Ainda sobre o componente <b>jMenu3</b> adicione mais dois componentes <b>Item de Menu/Botão de Seleção</b>. Digite nas <b>Propriedades</b> no campo <b>text</b> Inserir no outro Excluir respectivamente. Seu menu será semelhante ao mostrado na Figura ao lado.</p>	

Para finalizar a construção desse menu, você adicionará mais um separador. Para que ele fique entre os itens Cotar e Opções no menu Editar. Para fazer isso. Adicione clicando como direito sobre o componente **Menu**. Note que o componente separador aparecerá ao final de todos os componentes desse menu. Arraste-o sobre o componente **jMenu3** na janela **Inspetor**. Note que ao arrastar você terá uma linha riscando o ponto onde ficará o componente quando soltar. Na construção dos menus, você havia adicionado na propriedade do menu Arquivo, em ***mnemonic***, a letra A e no menu Editar letra E. O GUI Builder da IDE NetBeans gera o método em Java `setMnemonic` para indicar que A é mnemônico para esse menu. Pressionando a tecla **Alt** e a letra **A** abre o menu como se você houvesse clicado no menu com o ponteiro do mouse.

<p>Na janela <b>Inspetor</b>, selecione o item <b>Recortar</b>, e na janela <b>Propriedades</b> vá ao item <b>accelerator</b> e clique nos botão com três pontos: Ao clicar, você verá a caixa de diálogo <b>accelerator</b> aparecer. Selecione na caixa de diálogo em <b>Virtual Key</b> o item <b>VK_K</b> e ao lado marque <b>Ctrl</b>. Note que em <b>Key Stroke</b> você terá a sequência <b>Ctrl+K</b> como mostra a imagem.</p>	
---	--



O método Java aqui usado é o `setAccelerator`. As teclas aceleradoras somente em ser associadas a itens de menus e não em menus. Com esse método você cria teclas de atalho para seus menus, como o feito no caso, CTRL+X. Dê um preview no programa para visualizar o atalho no menu. Os itens, Copiar e Colar, também receberão atalhos. Para isso, selecione o item Copiar e na janela Inspetor, adicione o atalho na janela Propriedades no item `accelerator`: Ctrl+C, faça agora o mesmo com o item Colar: Ctrl+V. Pode ser digitado.

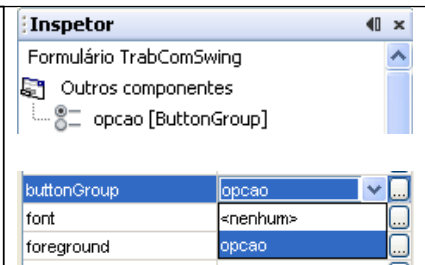


## AGRUPANDO ITENS DE MENU

Se você selecionar no menu Opções, Inserir e depois selecionar Excluir, notará que os dois ficam selecionados, ou seja, não existe uma opção entre os dois.

Para agrupar esses dois itens de menu, na janela Paleta, selecione **Grupo de botão**. Clique no design. Note que surgirá na janela Inspetor, abaixo de Outros Componentes um novo componente chamado de `buttonGroup1`, renomeie para opção. Agora é só selecionar os dois itens de menu, e escolher o grupo de botão criado (Opcao).

Agora você só poderá selecionar um entre os dois itens de menu, ou seja. Inserir ou Excluir.



## ADICIONANDO ÍCONES AO MENU

Os itens de menu são muito semelhantes aos botões. Na verdade, a classe estende a classe `AbstractButton`. Portanto, assim como nos botões, os menus tem também a possibilidade de ter ícones, além do rótulo.

Para se adicionar um ícone, selecione o componente `JMenuItem`, o que representa o Recortar e clique no botão com três pontos na propriedade `icon` na janela Propriedade. Aparecerá a caixa de diálogo `icon`, no qual você terá algumas opções para selecionar a sua imagem:

- URL — Você adiciona uma imagem usando um link.
- File — Um arquivo em um lugar qualquer em seu disco.
- Classpath — Sua imagem dentro de um diretório no seu pacote.
- No picture — O padrão, caso você não queira colocar uma imagem como ícone.

Selecione o item File e clique no botão Selecionar do Arquivo. Assim que selecionada a imagem, você terá a em Preview para visualizá-la. Basta confirmar se estiver correto. Agora que você sabe como fazer, coloque ícones nos outros menus.

## CRIANDO MENUS FLUTUANTES

Muitos aplicativos atuais usam menus flutuantes, mais conhecidos como menu pop-up. Desenvolver um menu flutuante no GUI Builder do NetBeans é muito simples. Na janela de componentes em **Paleta**, vá até o componente **Menu Pop-up** e arraste para o seu design, em qualquer parte do seu aplicativo. Um menu flutuante é criado da mesma forma como se cria um menu comum exceto que um menu flutuante não tem título. Na janela **Inspetor** você terá componente `JPopupMenu1`.

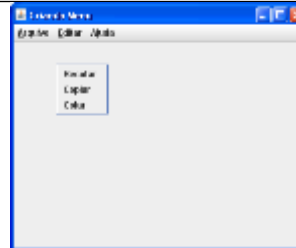
Adicione através da janela **Inspetor**, no componente **jPopupMenu1**, três componentes **JMenuItem**. Cada um representará Recortar, Copiar e Colar respectivamente. Adicione ícones nesses três itens também, tornando assim o menu muito mais agradável.

Para adicionar o menu ao seu aplicativo, você terá que trabalhar um pouco com o código. Na árvore de componentes da janela **Inspetor**, selecione (jFrame)

Selecione na janela **Propriedades**, em **Events** o evento **mouseReleased** dê um duplo clique, logo irá para o código fonte. E deverá ser acrescido o seguinte código.

```
Private void formMouseReleased(java.awt.event.MouseEvent evt){
    if(e.isPopupTrigger())
        jPopupMenu1.show(evt.getComponent(), evt.getX(), evt.getY());
}
```

Esse código vai exibir o menu flutuante na posição em que o mouse estiver quando você clicar com o botão direito. O código colocado foi necessário para que o evento do mouse ocorresse sobre o componente selecionado, e que quando o botão direito fosse acionado, o evento gatilho pop-up seria gerado.



## COMENTANDO O CÓDIGO DO MENU FLUTUANTE

void show( Component c, int x, int y) - exibe o menu pop-up.

Parâmetros: c o componente no qual o menu flutuante vai aparecer.

x, y são coordenadas (no espaço existente do componente) do canto superior esquerdo do menu flutuante.

boolean isPopupTrigger — retorna true se o evento do mouse for o gatilho pop-up. Ou seja, se você alterar o código, retirando ou comentando apenas o **if** perceberá na execução do aplicativo, que o menu pop-up é exibido tanto com o botão direito do mouse (o chamado gatilho), como também o botão esquerdo.

```
//if(evt.isPopupTrigger())
jPopupMenu1.show(evt.getComponent(), evt.getX(), evt.getY());
```

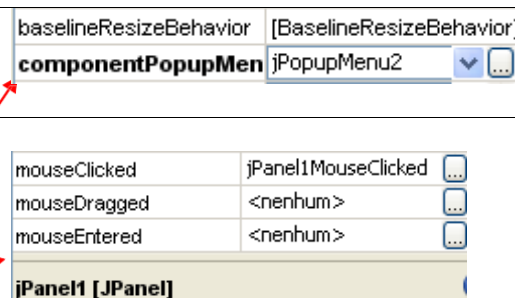
## UMA OUTRA FORMA DE UTILIZAR O MENU POP-UP

Claro que existe uma outra forma de você adicionar um menu do tipo pop-up na sua aplicação. Neste caso, comente o código anterior para que você possa ver essa nova forma em ação. Na janela **Paleta**, arraste o componente Painel para o seu aplicativo no GUI Builder do NetBeans. Arraste para ocupar toda a área do seu aplicativo. Se achar mais conveniente, altere o layout do componente **jFrame** para **BorderLayout** (preferível), e o coloque no centro.

Caso não esteja, Vá na janela **Propriedades** e na propriedade **Direction** selecione Center nesse componente.

Com o componente **jPanell** selecionado, vá na janela **Propriedades** e na propriedade **componentPopupMenu** selecione o componente jPopuprnnenu1, como mostra a imagem.

Em Eventos, ainda na janela **Propriedades**, vá ao evento **mouseClicked** e faça o procedimento já visto anteriormente.



```
private void jPanel1MouseClicked(java.awt.event.MouseEvent evt) {
    jPanel1.getComponentPopupMenu();
}
```

## TRABALHANDO COM ALGUNS COMPONENTES, SWING

O Swing oferece muitos componentes diferentes de interface com usuário, além de menus e Contêiner que você já viu, existem botões, campos de texto, rótulos, áreas de texto, caixas de verificação, botões de rádio, barras de rolagem, listas suspensas, caixas de combinação, slider, barras de progresso, caixas de mensagem e etc.

Você saberá aqui como trabalhar com alguns desses componentes.

### OBSERVAÇÃO:

*Todos os componentes Swing compartilham de uma superclasse comum, `javax.swing.JComponent` herdam vários métodos que você certamente usará em seus próprios aplicativos.*

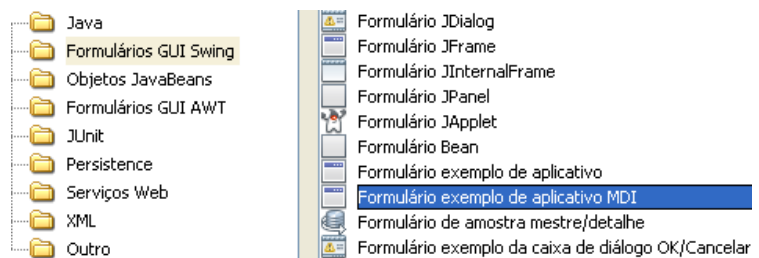
### COMPONENTES BÁSICOS DE INTERFACE GRÁFICA

Um programa GUI lava executando em diferentes plataformas tem uma aparência diferente às vezes, até diferentes interações de usuário em cada plataforma. Juntos, o aspecto e a maneira como o usuário interage com o programa são conhecidos como aparência e comportamento (—look and feel) desse programa.

Nos componentes a seguir também serão abordados algumas propriedades e comportamentos.

#### O PROJETO

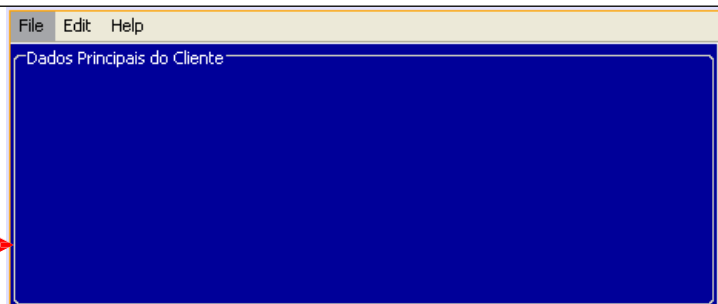
Crie um novo projeto chamado de MaisSwing. Clicando em Novo Arquivo, vá em Categorias no item **Formulários GUI Swing** selecionando em Tipo de Arquivo o item **Formulário exemplo de aplicativo MDI**, que já vem configurado com um menu inicial. Coloque o nome de componentes.



#### JFRAME E JPANEL

Todo programa Swing deve ter pelo menos um container (também chamado top-level container). Normalmente ele é um **JFrame** (uma janela completa, com decorações). Cada **JFrame** representa uma janela. O **JFrame** provê o suporte para outros componentes, assim como trabalho com eventos. Isso você já viu anteriormente.

O **JFrame** possui duas divisões. Uma delas é o **ContentPane**, que representa a parte da janela onde podemos montar nossa aplicação. O **ContentPane** na verdade é um **JPanel**, um outro container que normalmente é utilizado para receber componentes que fazem parte do



mesmo contexto de layout. Isso é necessário para separar o design de partes da janela. Exclua o componente Desktop e adicione um container **Painel** no seu design.

Selecione o componente que vem com o nome de **jPainel** na janela Propriedades e clique na propriedade border. Na caixa de diálogo border, selecione TitleBorder, em **Borda de Títulos**, na janela **Propriedades** abaixo digite —*Dados Principais do Cliente*” na propriedade **Título**, podendo também mudar a cor da fonte em **Fonte** <Enter>.

Crie mais dois painéis, abaixo, e coloque no título Outras Informações e Ações respectivamente.

## JLABEL

Os rótulos fornecem instruções de texto ou informações sobre uma GUI. Ele pode conter ícone ou não. Criado a partir da classe JLabel um rótulo é usado diversas vezes para identificar o objetivo de um outro determinado componente. Um rótulo não pode ser editado diretamente por um usuário. Para se criar um rótulo no GUI Builder do NetBeans, na janela Paleta arraste Rótulo para o design do seu aplicativo. Dê um duplo clique e digite Nome: Arraste mais quatro componentes JLabel e altere o texto para: Endereço, Cidade, UF, CEP respectivamente.

## JTEXTFIELD

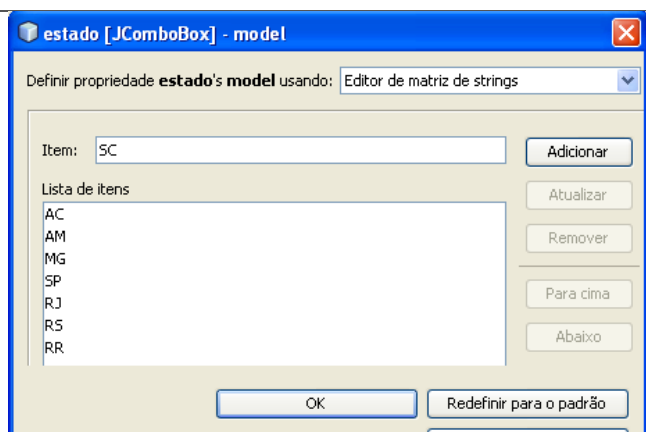
Um campo de texto é uma área em uma interface onde um usuário pode introduzir e modificar texto no teclado. Representado pela classe JTextField, no qual pode manipular uma linha de entrada. Adicione ao aplicativo um **Campo de Texto** limpe seu conteúdo nas propriedades em **Text** dimensione para o tamanho desejado.

Adicione mais dois componentes e **Campo de Texto** limpe também o conteúdo da mesma maneira anterior, Em especial, esses três JTextFields representarão os campos nome, endereço e cidade respectivamente.

## CAIXAS DE COMBINAÇÃO

Uma caixa de combinação fornece uma lista de itens entre os quais o usuário pode escolher. A classe JComboBox de Swing pode ser usada para criar esse componente de interface com o usuário. Você terá um campo que representará os Estados, em UF, que ao invés de ser uma caixa de texto comum, será uma caixa de combinação, podendo você assim, selecionar ao invés de digitar.

Para criar uma caixa de combinação, na janela Paleta clique em **Caixa de Combinação**. Com esse componente selecionado, vá na janela **Propriedades** e clique na propriedade **Model**. Aparecerá uma caixa de diálogo como a mostrada na imagem. Aqui você adiciona ou edita o conteúdo do componente JComboBox, Se você for adicionar, clique no botão Add. Caso for editar, selecione o item em Lista de item e ao alterá-lo, clique no botão Edit Não desejando mais o item, exclua-o em Remove. Os botões **Para Cima** e **Abaixo** são usados para alterar a ordem da lista.



Adicione ou edite algumas opções, colocando algumas UF e confirme após. Caso você deseje que a pessoa também possa digitar, não somente selecionar na lista, marque a opção **Editable** na janela **Propriedades** desse componente, isso adicionará ao código o método **setEditable** que ao se tomar **true** fará com que você possa não somente selecionar como também editar um valor.

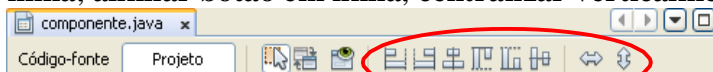
## JFormattedTextField

Semelhante ao componente `TextField`, esse componente surgiu com o intuito de Permitir que os dados entrados sejam formatados, para que o usuário coloque apenas as Informações de acordo com as regras de máscara, arraste o componente **Campo Formatado** da janela **Paleta**.

Limpe seu conteúdo. Essa caixa servirá para adicionar o CEP.

O arrastar dos componentes não foi abordado por ser algo de simples compreensão, já que o Matisse do GUI Builder Form do NetBeans é extremamente profissional.

Apenas observe que ao arrastar os componentes, você terá que alinhá-los. Para alinhar os Labels de Nome até CEP. selecione-os, mantendo a tecla Shift pressionada. Na parte superior do GIM Builder no Design, você verá uma barra de ferramentas para alinhamento dos componentes. Clique no ícone que desejar: **alinhar a esquerda, a direita, centralizar horizontalmente, Alinhar em cima da linha, alinhar botão em linha, centralizar verticalmente mudar redimensionamento horizon-**



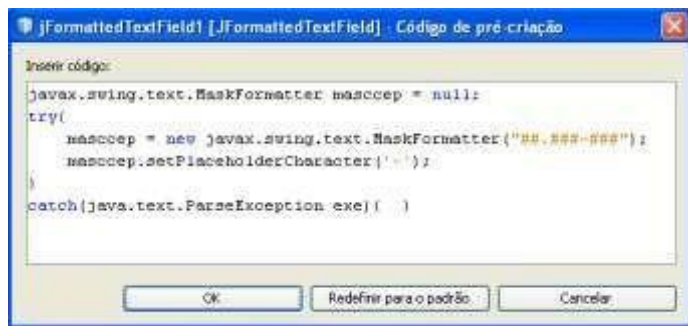
**tal e mudar redimensionamento Vertical.**

Selecione as duas caixas de texto que representam o Nome e o Endereço. Clique no ícone centralizar horizontal para que as duas fiquem da mesma largura sempre que a janela for redimensionada.

Selecione o componente `JFormattedTextField1`, vá até a janela **Propriedades** e clique em **Código**.

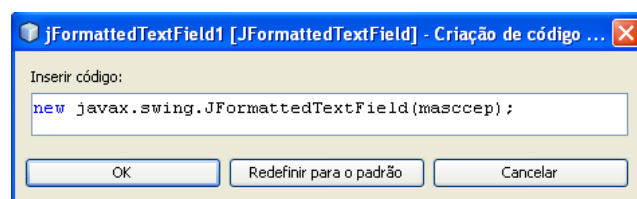
No item **Código de pré-criação**, dê um clique no botão com três pontos. Na caixa de diálogo digite o código a seguir e confirme após:

```
javax.swing.text.MaskFormatter masccep = null;
try{
    masccep = new javax.swing.text.MaskFormatter("##.###-###");
    masccep.setPlaceholderCharacter('-');
}
catch(java.text.ParseException exe){}
```



No item **Criação de Código Personalizado** dê um clique nos três botões e adicione o código a Seguir e confirme após:

```
new javax.swing.JFormattedTextField(masccep);
```



A classe **MaskFormatted** usada para formatar e editar Strings. O comportamento da classe **MaskFormatter** é controlado por um tipo de máscara de String que especifica os caracteres válidos que podem ser digitados naquele campo. A seguir você tem uma tabela com os tipos de caracteres que podem ser especificados.

Caractere	Descrição
#	Qualquer número válido, usa <code>Character.isDigit</code> ;
'	Caractere de escape, usado para o escape de qualquer caractere de formato especial.
U	Qualquer caractere( <code>Character.isLetter</code> ). Todas as letras minúsculas se tornam maiúsculas.

- L** Qualquer caractere(Character.isLetter), Todas as letras são transformadas para minúsculas.
- A** Qualquer caractere ou número (Character.isLetter ou Character.isDigit).
- ?** Qualquer caractere
- \*** Qualquer coisa
- H** Qualquer caractere hexadecimal (0-9, A-f ou A-F);

O que você viu foi o uso da classe para trabalhar com a máscara de uma entrada de CEP o objeto MaskFormatter deve ser criado dentro de um bloco try{ }, fora dele o código não compila.

Com isso, no trecho a seguir você definiu como seria a entrada dos dados:

```
mascep = new javax.swing.text.MaskFormatter("#####-###");
```

E como método setPlaceholderCharacter você definiu qual seria o caractere que seria mostrado ao usuário na hora de preencher o formulário:

```
mascep.setPlaceholderCharacter('-');
```

Quando você adicionou o código em **Criação de código Personalizado**, você instanciou o componente JFormattedTextField1 à máscara, como você mesmo poderá conferir no código `new javax.swing.JFormattedTextField(mascep);`

Você pode conferir compilando o seu código.

## Exercícios:

1 – Desenvolva a classe main();

O seguinte código deverá ser implementado:

```
package maisswing;
```

```
/** @author Giovanni */
```

```
public class Main {
    public static void main(String[] args) {
        new componente().setVisible(true);
    }
}
```

Agora você fará a parte inferior intitulada: **outras informações**. Adicione os seguintes componentes mostrados na tabela a seguir:

Componente	Quantidade	Propriedade texto
JFormattedTextField	2	Em branco
JLabel	1	Telefone:
	1	Fax:
	1	E-mail:
-	1	Est. Civil
JTextField	1	Em branco
JRadioButton	1	Solteiro
	1	Casado

2– Faça a máscara para o campo Telefone e Fax. Clique sobre o campo que representa.

3– Agrupe os botões rádios, para que quando marcar um desmarque o outro como feito no menu.

4– Renomeie as variáveis, para um possível aproveitamento, com nomes apropriados que indique o campo que estão representando.

5 – Na terceira parte, no componente JPanel3, cujo título é **Ações**, adicione três Botões e altere seus rótulos, como mostra a imagem

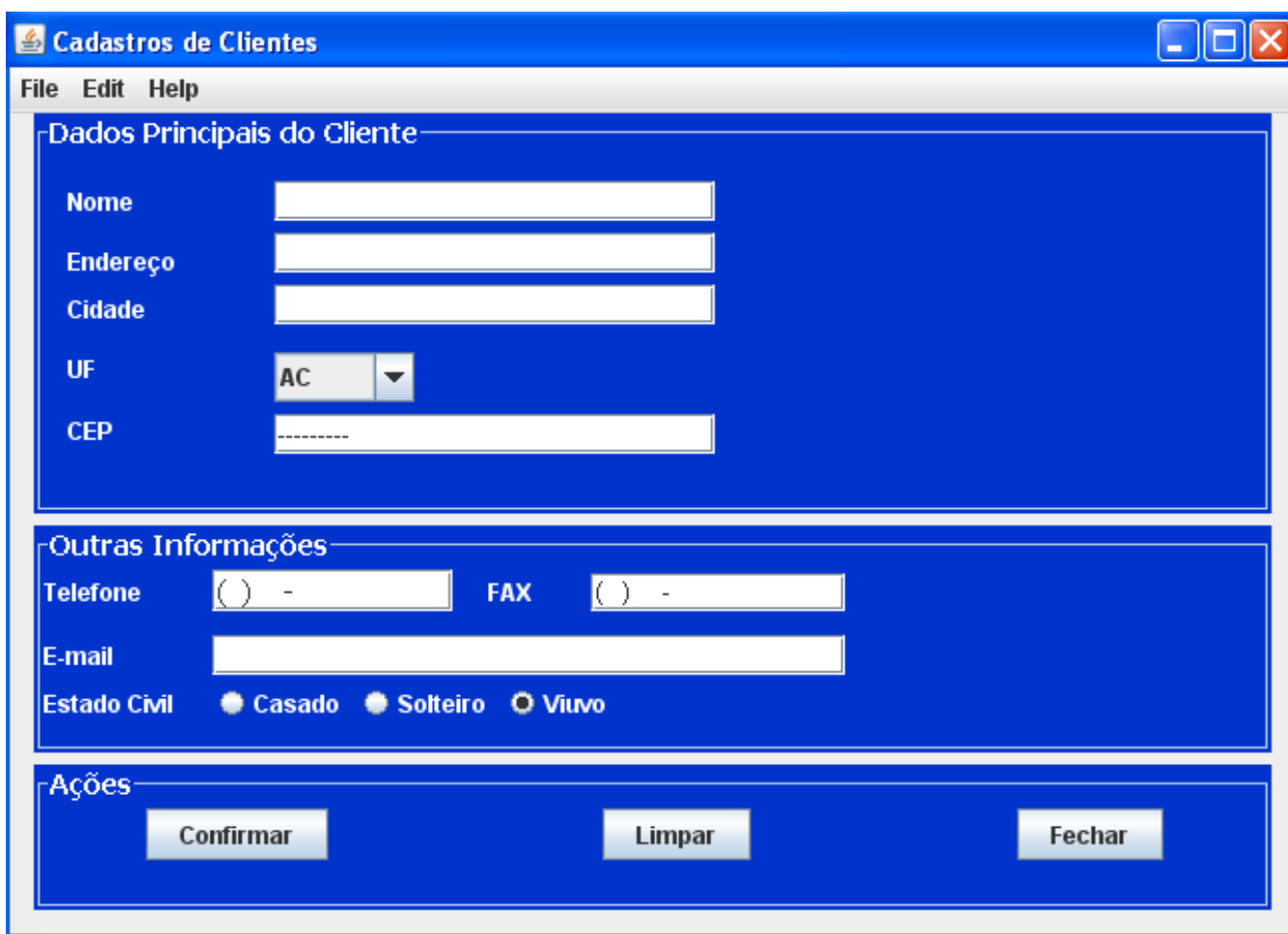


## JBUTTON

Um botão não é algo tão novo para você aqui no NetBeans, pois você já usou em outros locais nos aplicativos iniciais. Mas o que não foi feita é a explicação de seu uso.

Um botão é um componente usado pelo usuário para clicar e acionar uma ação específica. Um programa Java pode trabalhar com vários tipos de botões, um botão de comando gera um evento quando o usuário clica sobre ele com o mouse. Os botões de comando são criados com a classe **JButton**. O texto na face de um botão é chamado de rótulo do botão. Faremos uma abordagem desses eventos no próximo material a ser entregue.

No final estará o aplicativo com a seguinte interface:

The screenshot shows a Java Swing window titled "Cadastros de Clientes". It has a menu bar with "File", "Edit", and "Help". The main area is divided into three sections. The first section, "Dados Principais do Cliente", contains text labels for "Nome", "Endereço", "Cidade", "UF", and "CEP", each followed by a text input field. The "UF" field is a dropdown menu currently showing "AC". The second section, "Outras Informações", contains labels for "Telefone" and "FAX", each followed by a masked input field (e.g., "( ) - "). Below these is an "E-mail" label with a text input field. At the bottom of this section is the "Estado Civil" label with three radio buttons labeled "Casado", "Solteiro", and "Viuvo". The third section, "Ações", is at the bottom and contains three buttons: "Confirmar", "Limpar", and "Fechar".