

- Choose page language[®]
 - [English](#)
 - [Japanese](#)
 - [Russian](#)
 - [Simplified Chinese](#)



- [NetBeans IDE](#)
- [NetBeans Platform](#)
- [Enterprise](#)
- [Plugins](#)
- [Docs & Support](#)
- [Community](#)

Search

[HOME](#) / [Docs & Support](#)

PRINTABLE VERSION

Projetando uma GUI Swing no NetBeans IDE



Este tutorial conduzirá você pelo processo de criação da interface gráfica do usuário (GUI) para uma aplicação nomeada ContactEditor, utilizando GUI Builder do NetBeans IDE. No processo, será feito o layout de uma GUI inicial que permite que você exiba e edite informações de contato dos indivíduos incluídos em um banco de dados de funcionários.

Neste tutorial, você aprenderá a: usar a interface do GUI Builder, criar um Contêiner de GUI, adicionar, redimensionar e alinhar componentes, ajustar o ancoramento de componentes, definir o comportamento de redimensionamento automático de componentes, editar propriedades de componentes.

Este tutorial demora aproximadamente 30 minutos para ser concluído.

Conteúdo

- [Conceitos Básicos](#)
 - [Criando um Projeto](#)
 - [Criando um Contêiner JFrame](#)
- [Familiarizando-se com o GUI Builder](#)
- [Conceitos-Chave](#)
 - [Design Livre](#)
 - [Posicionamento Automático do Componente \(Encaixe\)](#)
 - [Feedback Visual](#)
- [Primeiro o Principal](#)
 - [Adicionando Componentes: Conceitos Básicos](#)
 - [Adicionando Componentes Individuais ao Form](#)
 - [Adicionando Vários Componentes ao Form](#)
 - [Inserindo Componentes](#)
- [Avançando](#)
 - [Alinhamento de Componentes](#)
 - [Alinhamento da Linha de Base](#)
- [Revisando o Que Aprendemos](#)
 - [Adicionando, Alinhando e Ancorando](#)
 - [Dimensionamento de Componentes](#)
 - [Recuo](#)
- [Fazendo os Ajustes Finais](#)
- [Visualizando a GUI](#)
- [Implantando Aplicações GUI](#)
- [Consulte Também](#)

Para concluir este tutorial, você precisa dos seguintes recursos e softwares.

Software ou Recurso	Versão Necessária
NetBeans IDE	versão 6.9, 7.0, 7.1, 7.2, 7.3 ou 7.4
JDK (Java Development Kit)	versão 6 ou 7

Introdução

O GUI Builder do IDE torna possível construir GUIs de aparência profissional sem um conhecimento profundo dos gerenciadores de layout. Você pode criar forms, simplesmente colocando componentes onde desejar.

Para obter descrições sobre o feedback visual do GUI Builder, é possível utilizar a [Legenda de Feedback Visual do GUI Builder](#).

Criando um Projeto

Como todo o desenvolvimento em Java no IDE ocorre dentro dos projetos, precisamos primeiro criar um projeto `ContactEditor` para armazenar os códigos-fonte e outros arquivos do projeto. Um projeto do IDE é um grupo de arquivos de códigos-fonte Java e metadados associados que inclui arquivos de propriedades específicos do projeto, um script de construção Ant que controla as definições de construção e execução e um arquivo `project.xml`, que mapeia destinos de Ant para comandos do IDE. Embora as aplicações Java geralmente consistam em vários projetos do IDE, para o objetivo deste tutorial, construiremos uma aplicação simples, inteiramente armazenada em um único projeto.

Para criar um novo projeto de aplicação `ContactEditor`:

1. Escolha Arquivo > Novo Projeto. Se preferir, é possível clicar no ícone Novo Projeto na barra de ferramentas do IDE.
2. No painel Categorias, selecione o nó Java e no painel Projetos, escolha Aplicação Java. Clique em Próximo.
3. Insira `ContactEditor` no campo Nome do Projeto e especifique o local do projeto.
4. Deixe desmarcada a caixa de seleção Utilizar Pasta Dedicada para Armazenar Bibliotecas.
5. Verifique se a caixa de seleção Definir como Projeto Principal está marcada e desmarque o campo Criar Classe Principal.
6. Clique em Finalizar.

O IDE criará a pasta `ContactEditor` no sistema no local designado. Essa pasta contém todos os arquivos associados do projeto, como o script Ant, pastas para armazenamento de códigos-fontes e testes e uma pasta para os metadados específicos do projeto. Para exibir a estrutura do projeto, utilize a janela Arquivos do IDE.

Criando um Contêiner JFrame

Após criar a nova aplicação, você pode ter percebido que a pasta Pacotes de Códigos-fonte na janela Projetos contém um nó `<pacote default>` vazio. Para prosseguir com a construção da nossa interface, precisamos criar um contêiner Java no qual colocaremos os outros componentes requeridos da GUI. Nessa etapa, criaremos um contêiner utilizando o componente `JFrame` e o colocaremos em um novo pacote.

Para adicionar um contêiner `JFrame`:

1. na janela Projetos, clique com o botão direito do mouse no nó `ContactEditor` e selecione Novo > Form JFrame. Outra opção é localizar um form JFrame escolhendo Novo > Outro > Forms da GUI Swing > Form JFrame.
2. Insira `ContactEditorUI` como Nome da Classe.
3. Insira `my.contacteditor` como o pacote.
4. Clique em Finalizar.

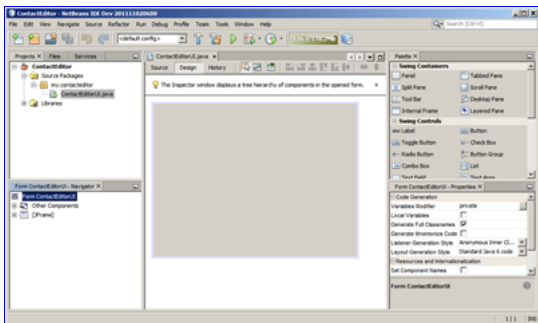
O IDE criará o form `ContactEditorUI` e a classe `ContactEditorUI` na aplicação `ContactEditorUI.java` e abrirá o form `ContactEditorUI` no GUI Builder. Observe que o pacote `my.contacteditor` substitui o pacote `default`.

[início](#)

Familiarizando-se com o GUI Builder

Agora que já configuramos o novo projeto para a aplicação, vamos nos familiarizar com a interface do GUI Builder.

Observação: Para explorar a interface do GUI Builder com uma demonstração interativa, assista o screencast [Explorando o GUI Builder \(.swf\)](#).



Quando adicionamos o contêiner `JFrame`, o IDE abriu o form `ContactEditorUI` recém-criado em uma guia do Editor com uma barra de ferramentas que contém vários botões, como mostra a ilustração anterior. O form `ContactEditor` aberto na view Design do GUI Builder e três janelas adicionais apareceram automaticamente ao longo das margens do IDE, permitindo navegar, organizar e editar forms GUI à medida que você os constrói.

As várias janelas do GUI Builder incluem:

- **Área de Design.** A janela principal do GUI Builder para criar e editar forms de GUI do Java. O botão Código-fonte da barra de ferramenta permite exibir o código-fonte de uma classe; o botão Design permite a exibição gráfica dos componentes da GUI; o botão Histórico permite o acesso ao histórico local de alterações do arquivo. Os botões adicionais da barra de ferramentas oferecem acesso conveniente aos comandos comuns, como escolher entre os modos Seleção e Conexão, alinhamento de componentes, definição do comportamento de dimensionamento automático e exibição de forms.
- **Navegador.** Fornece uma representação de todos os componentes, ambos visual e não-visuais na sua aplicação, como uma hierarquia de árvore. O Navegador também oferece apresentação visual sobre qual componente da árvore está sendo editado no GUI Builder, bem como permite organizar componentes nos painéis disponíveis.
- **Paleta.** Uma lista customizável de componentes disponíveis contendo guias para jfc/swing e JavaBeans componentes awt, bem como gerentes de layout. Além disso, é possível criar, remover e reorganizar as categorias exibidas na Paleta utilizando o personalizador.

alguns métodos, e propriedades, bem como o gerenciamento de eventos de forma simples e performática.

- **Janela Propriedades.** Exibe as propriedades do componente atualmente selecionado no navegador GUI Builder, janela janela, projetos, ou arquivos janela.

Se você clicar no botão Código-fonte, o IDE exibirá o código-fonte Java da aplicação no Editor com seções de código geradas automaticamente pelo GUI Builder e indicadas por áreas cinzas (elas ficam azuis quando selecionadas), chamadas Blocos Protegidos. Blocos Protegidos são áreas protegidas que não são editáveis na view de Código-fonte. Só é possível editar o código exibido nas áreas brancas do Editor na view de Código-fonte. Se for necessário fazer alterações no código em um Bloco Protegido, clique no botão Design para que o Editor do IDE volte ao GUI Builder, onde você poderá fazer os ajustes necessários ao form. Ao salvar as alterações, o IDE atualizará os códigos-fonte do arquivo.

Observação: para os desenvolvedores avançados, está disponível o Gerenciador de Paleta, que ativa adicionar componentes personalizados de JARs, bibliotecas ou projetos à Paleta. Para adicionar componentes personalizados por meio do Gerenciador de Paleta, escolha Ferramentas > Paleta > Componentes Swing/AWT.

[início](#)

Conceitos-Chave

O GUI Builder do IDE resolve o problema central da criação de GUIs do Java otimizando o fluxo de dados da criação de interfaces gráficas, liberando os desenvolvedores das complexidades dos gerenciadores de layout do Swing. Ele faz isso estendendo o GUI Builder atual do NetBeans IDE para dar suporte ao paradigma "Design Livre" com regras simples de layout, fáceis de compreender e utilizar. À medida que você faz o layout do form, o GUI Builder fornece linhas-guias visuais sugerindo o espaçamento e alinhamento ideais dos componentes. Em segundo plano, o GUI Builder converte as decisões de design em uma UI funcional, que é implementada com o uso de um gerenciador de layouts GroupLayout e outras construções do Swing. Como ele utiliza um modelo de layout dinâmico, as GUIs construídas com o GUI Builder se comportam como o esperado em runtime, ajustando para acomodar as alterações feitas sem alterar os relacionamentos definidos entre os componentes. Quando você redimensionar o form, alterar configurações regionais ou especificar outra aparência, a GUI se ajustará automaticamente para respeitar a aparência do destino e os suplementos e deslocamentos da funcionalidade.

Design Livre

No GUI Builder do IDE, é possível construir forms simplesmente colocando os componentes onde quiser, como se estivesse utilizando posicionamento absoluto. O GUI Builder descobre quais atributos de layout são obrigatórios e gera o código automaticamente. Você não precisa se preocupar com suplementos, âncoras, preenchimentos, etc.

Posicionamento Automático do Componente (Encaixe)

À medida em que os componentes são adicionados a um form, o GUI Builder fornece feedback visual, que auxilia no posicionamento dos componentes com base na aparência e funcionalidade do seu sistema operacional. O GUI Builder oferece dicas in-line úteis e outros feedbacks visuais em relação ao local no qual os componentes devem ser colocados no form, encaixando automaticamente os componentes na posição ao longo das linhas-guias. Ele faz essas sugestões com base nas posições dos componentes já colocados no form, enquanto permite que o preenchimento permaneça flexível, de maneira que a aparência e a funcionalidade do destino diferente sejam processadas corretamente em runtime.

Feedback Visual

O GUI Builder também oferece feedback visual sobre as relações de ancoragem e encadeamento de componentes. Esses indicadores permitem identificar rapidamente as várias relações de posicionamento e o comportamento fixo dos componentes que afetam a maneira como a GUI aparecerá e se comportará em runtime. Isso acelera o processo de design de GUI, permitindo que você crie rapidamente interfaces visuais com aspecto profissional e que funcionam.

[início](#)

Primeiro o Principal

Agora que você está familiarizado com a interface do GUI Builder, este é o momento para começar a desenvolver a UI da nossa aplicação ContactEditor. Nesta seção, veremos a utilização da Paleta do IDE para adicionar os vários componentes GUI necessários ao nosso form.

Graças ao paradigma Design Livre do IDE, não é necessário entrar em conflito com os gerenciadores de layout para controlar o tamanho e posição dos componentes nos contêineres. Tudo o que você precisa fazer é arrastar e soltar os componentes necessários no form da GUI conforme exibido nas ilustrações abaixo.

Observação: Consulte o screencast [Adicionando componentes individuais e múltiplos componentes \(.swf\)](#) para ver uma demonstração interativa na seção abaixo.

Adicionando Componentes: Conceitos Básicos

Embora o GUI Builder simplifique o processo de criação de GUI do Java, é sempre útil esboçar a maneira como quer que a interface se pareça antes de começar a fazer seu layout. Muitos designers de interfaces consideram isso uma técnica de "melhor prática", entretanto, para este tutorial, você pode simplesmente dar uma olhada como nosso form preenchido deve se parecer indo para a seção [Visualização da GUI](#).

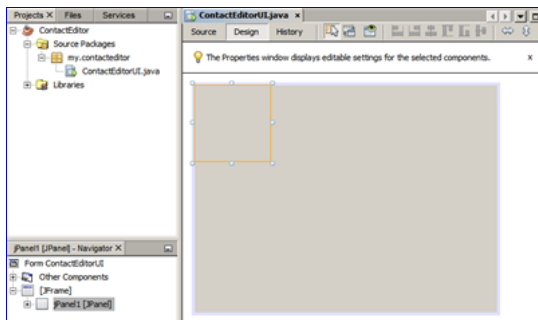
Como já adicionamos um JFrame como contêiner de nível superior do form, a próxima etapa é adicionar dois JPanels que nos permitirão unir os componentes da nossa UI utilizando bordas com títulos. Veja as ilustrações abaixo e observe o comportamento "arrastar e soltar" do IDE para fazer isso.

Para adicionar um JPanel:

1. Na janela Paleta, selecione o componente Painei na categoria Contêineres do Swing clicando e soltando o botão do mouse.
2. Mova o cursor até o canto esquerdo superior do form no GUI Builder. Quando o componente estiver localizado próximo às margens superior e esquerda do contêiner, linhas-guia de alinhamento horizontal e vertical aparecerão indicando as

margens preferenciais. Clique no form para colocar o JPanel nesse local.

O componente `JPanel` aparece no form `ContactEditorUI` com realce na cor laranja, o que significa que ele está selecionado. Após soltar o botão do mouse, pequenos indicadores serão exibidos para mostrar as relações de ancoragem do componente e um nó `JPanel` correspondente será exibido na janela Navegador, como mostra a ilustração a seguir.

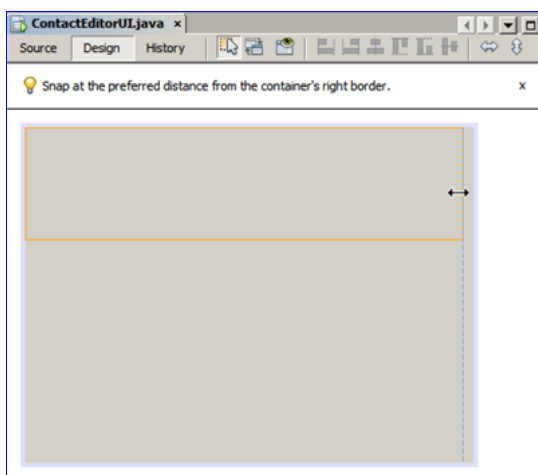


Depois, precisamos redimensionar o JPanel para dar espaço aos componentes que serão inseridos posteriormente, mas antes tomemos um minuto para indicar outras funcionalidades de visualização do GUI Builder. Para fazer isso, precisamos desmarcar o JPanel que acabamos de adicionar. Como nós ainda não adicionamos uma borda com título, o painel desaparecerá. Observe, entretanto, que quando ao passar o cursor sobre o JPanel, as margens mudarão para cinza claro para que sua posição possa ser vista claramente. Você precisa apenas clicar em qualquer lugar no componente para selecioná-lo novamente e fazer com que os handles de redimensionamento e os indicadores de ancoragem reapareçam.

Para redimensionar o JPanel:

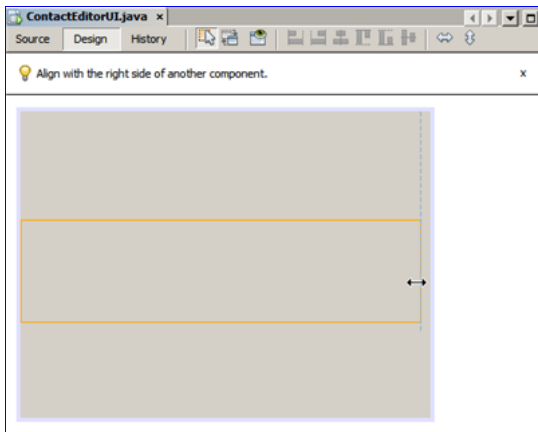
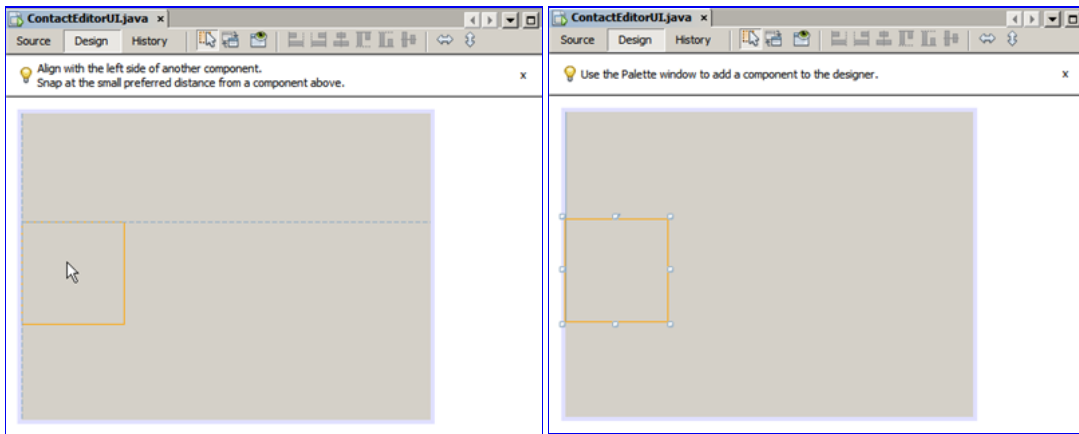
1. Selecione o JPanel recém-adicionado. Os pequenos handles de redimensionamento quadrados reaparecerão em torno do perímetro do componente.
2. Clique e segure o handle de redimensionamento na margem direita do JPanel e arraste-a até que a linha-guia de alinhamento pontilhada seja exibida próxima à margem do form.
3. Solte o handle de redimensionamento para redimensionar o componente.

O componente `JPanel` é estendido para ficar entre as margens esquerda e direita do contêiner de acordo com o deslocamento recomendado, como mostra a ilustração abaixo.



Agora que adicionamos um painel para conter a informação Nome da nossa UI, precisamos repetir o processo para adicionar outro

Agora que adicionamos um JPanel para conter a informação Nome da nossa GUI, precisamos repetir o processo para adicionar outro diretamente abaixo da primeira para a informação de E-mail. Consultando as ilustrações abaixo, repita as duas tarefas anteriores, prestando atenção ao posicionamento sugerido pelo GUI Builder. Observe que o espaçamento vertical sugerido entre os dois JPanels é muito mais estreito que entre as margens. Quando você tiver adicionado o segundo JPanel, redimensione-o para que preencha o espaço vertical restante do form.

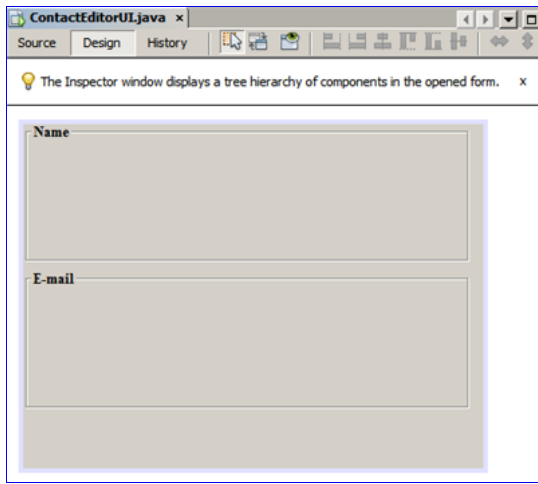


Como queremos distinguir visualmente as funções nas seções superior e inferior da nossa GUI, precisamos adicionar uma borda e título a cada JPanel. Primeiro faremos isso utilizando a janela Propriedades e depois a experimentaremos utilizando o menu pop-up.

Para adicionar bordas de título aos JPanels:

1. Selecione o JPanel superior no GUI Builder.
2. Na janela Propriedades, clique no botão de reticências (...) ao lado da propriedade Borda.
3. No editor de Bordas do JPanel exibido, selecione o nó TitledBorder no painel Bordas Disponíveis.
4. No painel Propriedades abaixo, insira `Name` para a propriedade Título.
5. Clique nas reticências (...) ao lado da propriedade Fonte, selecione Negrito em Estilo de Fonte e insira 12 no Tamanho. Clique em OK para fechar as caixas de diálogo.
6. Selecione o JPanel inferior e repita as etapas de 2 a 5, mas dessa vez clique com o botão direito do mouse no JPanel e acesse a janela Propriedades utilizando o menu pop-up. Insira `E-mail` para a propriedade Título.

As bordas com título serão adicionadas a ambos componentes JPanel.



Adicionando Componentes Individuais ao Form

Agora precisamos começar a adição dos componentes que apresentarão as informações de contato reais na nossa lista de contatos. Nessa tarefa, adicionaremos quatro JTextFields que exibirão as informações de contato e os JLabels que as descreverão. Ao fazer isso, observe as linhas-guia horizontais e verticais que o GUI Builder exibe, sugerindo o espaçamento preferencial dos componentes conforme definido pela aparência e funcionalidade do sistema operacional. Isso assegura que nossa GUI seja processada automaticamente respeitando a aparência do sistema operacional de destino em runtime.

Para adicionar um JLabel ao form:

1. Na janela Paleta, selecione o componente Label na categoria Controles do Swing.
2. Mova o cursor acima do JPanel `Name` adicionado anteriormente. Quando as linhas-guias aparecerem, indicando que o JLabel está posicionado no canto superior esquerdo do JPanel com pequena margem nos cantos superior e esquerdo, clique para posicionar o label.

O JLabel será adicionado ao form e um nó correspondente, representando o componente, será adicionado à janela Inspetor.

Antes de prosseguir, precisamos editar o texto de exibição do JLabel que acabamos de adicionar. Embora seja possível editar o texto de exibição do componente em qualquer ponto, a maneira mais fácil é fazer isso à medida que o adiciona.

Para editar o texto de exibição de um JLabel:

1. Clique duas vezes no JLabel para selecionar a exibição de texto.
2. Digite `Nome:` e pressione Enter.

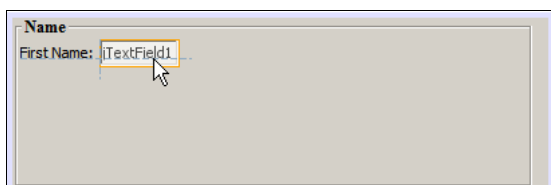
O novo nome do JLabel é exibido e a largura do componente será ajustada como resultado da edição.

Agora adicionaremos um JTextField para que possamos ter uma vista rápida da funcionalidade de alinhamento da linha de base do GUI Builder.

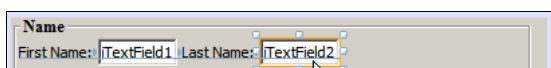
Para adicionar um JTextField ao form:

1. Na janela Paleta, selecione o componente Campo de texto na categoria Controles do Swing.
2. Mova o cursor imediatamente à direita do JLabel `Nome:` recém-adicionado. Quando a linha-guia horizontal aparecer, indicando que a linha de base do JTextField está alinhada com aquela do JLabel e o espaçamento entre os dois componentes for sugerido com uma linha-guia vertical, clique para posicionar o JTextField.

O JTextField se encaixará na posição no form alinhado com a linha de base do JLabel, conforme mostra a ilustração abaixo. Observe que o JLabel mudou ligeiramente para baixo para se alinhar à linha de base do campo de texto mais alto. Como sempre, um nó representando o componente será adicionado à janela Navegador.



Antes de prosseguir, precisamos adicionar mais um JLabel e JTextField imediatamente à direita desses já adicionados, conforme mostra a ilustração abaixo. Agora insira `Sobrenome:` como texto de exibição do JLabel e mantenha, por enquanto, o texto do espaço reservado dos JTextFields como está.

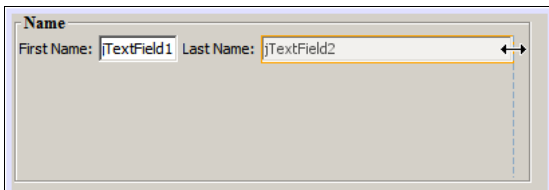




Para redimensionar um JTextField:

1. selecione o JTextField que nós acabamos de adicionar à direita do JLabel `Sobrenome:`.
2. Arraste o handle de redimensionamento direito do JTextField em direção à margem direita do JPanel de fechamento.
3. Quando as linhas-guia de alinhamento vertical aparecerem sugerindo a margem entre o campo de texto e a margem direita do JPanel, solte o botão do mouse para redimensionar o JTextField.

A margem direita do JTextField se encaixará em alinhamento à margem recomendada do JPanel, conforme mostra a ilustração abaixo.



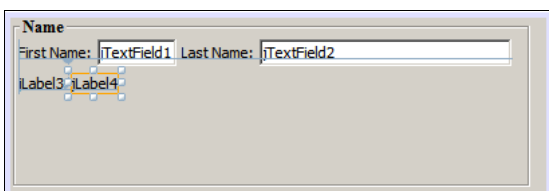
Adicionando Vários Componentes ao Form

Agora adicionaremos os JLabels `Título:` e `Apelido:` que descrevem dois JTextFields que adicionaremos em um minuto. Arrastaremos e soltaremos os componentes pressionando a tecla Shift para adicioná-los rapidamente ao form. Enquanto fazemos isso, observe novamente que o GUI Builder exibe linhas-guia horizontais e verticais sugerindo o espaçamento preferencial do componente.

Para adicionar vários JLabels ao form:

1. Na janela Paleta, selecione o componente Label na categoria Controles do Swing clicando e soltando o botão do mouse.
2. Mova o cursor sobre o form diretamente abaixo do JLabel `Nome:` adicionado anteriormente. Quando as linhas-guia aparecerem, indicando que a nova borda esquerda do JLabel está alinhada à do JLabel acima e houver uma pequena margem entre ambos, clique pressionando a tecla Shift para posicionar o primeiro JLabel.
3. Ainda pressionando a tecla Shift, clique para colocar outro JLabel imediatamente à direita do primeiro. Certifique-se de soltar a tecla Shift antes de posicionar o segundo JLabel. Se você se esquecer de soltar a tecla Shift antes de posicionar o último JLabel, basta pressionar a tecla Escape.

Os JLabels serão adicionados ao form criando uma segunda linha, como mostra a ilustração abaixo. Os nós que representam os componentes serão adicionados à janela Navegador.

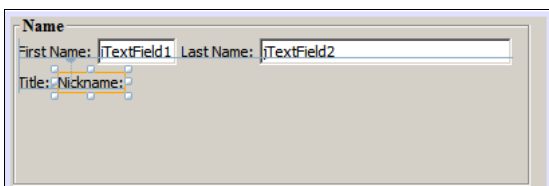


Antes de continuar, precisamos editar o nome do JLabels para que possamos ver o efeito dos alinhamentos que definiremos posteriormente.

Para editar o texto de exibição de JLabels:

1. clique duas vezes no primeiro JLabel para selecionar a exibição de texto.
2. Digite `Título:` e pressione Enter.
3. Repita as etapas 1 e 2, inserindo `Apelido:` para a propriedade de nome do segundo JLabel.

Os novos nomes dos JLabels serão exibidos no form e mudarão como resultado das larguras editadas, como mostra a ilustração abaixo.



Inserindo Componentes

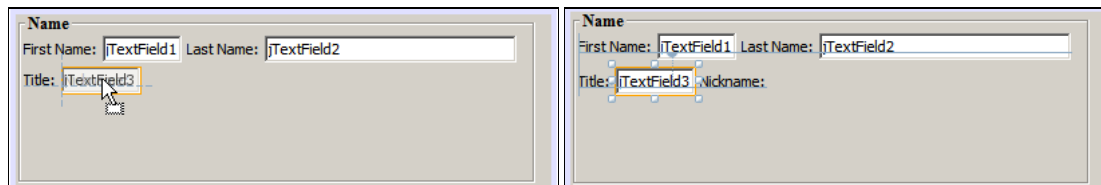
Observação: Consulte o screencast [Inserindo componentes \(.swf\)](#) para ver uma demonstração interativa na seção abaixo.

Geralmente, é necessário adicionar um componente entre componentes já colocados em um form. Quando você adiciona um componente entre dois componentes existentes, o GUI Builder automaticamente os mudará para dar espaço para o novo componente. Para demonstrar isso, inseriremos um JTextField entre os JLabels adicionados anteriormente, como mostram as duas ilustrações abaixo.

Para inserir um JTextField entre dois JLabels:

1. Na janela Paleta, selecione o componente Campo de texto na categoria Controles do Swing.
2. Mova o cursor acima dos JLabels `Título:` e `Apelido:` na segunda linha de maneira que o JTextField sobreponha ambos e fique alinhado às suas linhas de base. Se você tiver dificuldade em posicionar o novo campo de texto, poderá encaixá-lo à linha-guia esquerda do JLabel `Apelido:`, como mostra a primeira imagem abaixo.
3. Clique para posicionar o JTextField entre os JLabels `Título:` e `Apelido:`.

O JTextField se encaixará entre os dois JLabels. O JLabel da extremidade direita muda em direção à direita do JTextField para acomodar o deslocamento horizontal sugerido.



Ainda precisamos adicionar outro JTextField ao form que exibirá o apelido do contato no lado direito do form.

Para adicionar um JTextField:

1. Na janela Paleta, selecione o componente Campo de Texto na categoria Swing.
2. Mova o cursor para a direita do label `Apelido:` e clique para posicionar o campo de texto.

O JTextField se encaixará ao lado esquerdo do JLabel.

Para redimensionar um JTextField:

1. Arraste os handles de redimensionamento do JTextField do label `Apelido:` adicionado na tarefa anterior à direita do JPanel de fechamento.
2. Quando as linhas-guia de alinhamento vertical aparecerem sugerindo a margem entre o campo de texto e as margens do JPanel, solte o botão do mouse para redimensionar o JTextField.

A margem direita do JTextField se encaixará em alinhamento com a margem recomendada do JPanel e o GUI Builder deduzirá o comportamento de redimensionamento correto.

3. Pressione Ctrl-S para salvar o arquivo.

[início](#)

Avançando


Alinhamento é um dos aspectos mais fundamentais da criação GUI com aparência profissional. Na seção anterior, tivemos uma vista rápida das funcionalidades de alinhamento do IDE durante a adição dos componentes JLabel e JTextField ao nosso form `ContactEditorUI`. Depois, veremos de maneira mais profunda as funcionalidades de alinhamento do GUI Builder, conforme trabalhamos com os outros componentes necessários para a nossa aplicação.

Alinhamento de Componentes

Observação: Consulte o screencast [Alinhando e ancorando componentes \(.swf\)](#) para ver uma demonstração interativa na seção abaixo.

Sempre que você adiciona um componente a um form, o GUI Builder os alinha efetivamente, conforme evidenciado pelas linhas-guia de alinhamento exibidas. Entretanto, às vezes, também é necessário especificar diferentes relacionamentos entre os grupos de componentes. Anteriormente, nós adicionamos quatro JLabels necessários para a nossa GUI `ContactEditor`, mas nós não os alinhamos. Agora alinharemos as duas colunas de JLabels para que as margens direitas se alinhem.

Para alinhar componentes:

1. mantenha pressionada a tecla `Ctrl` e clique para selecionar os JLabels `Nome:` e `Título:` do lado esquerdo do form.
2. Clique no botão `Alinhar à Direita` na coluna () na barra de ferramentas. Alternativamente, clique com o botão direito do mouse em um deles e escolha `Alinhar > Direita` na coluna no menu pop-up.
3. Repita isso também para os JLabels `Sobrenome:` e `Apelido:`.

A posição dos JLabels muda para que as margens direitas do texto de exibição sejam alinhadas. Os relacionamentos de ancoragem serão atualizados, indicando que os componentes foram agrupados.



Train

[Java Programming Language](#)



Supp

[Oracle Development Support Offering for NetBeans IDE](#)

Documentat

[General Java Development External Tools and Services](#)
[Java GUI Applications](#)
[Java EE & Java Web Development](#)
[Web Services Applications](#)
[NetBeans Platform and Module Development](#)
[PHP and HTML5 Applications](#)
[C/C++ Applications](#)
[Mobile Applications](#)

[Sample Applications](#)
[Demos and Screenshots](#)

More

[FAQs](#)
[Contribute Documentation](#)
[Docs for Earlier Releases](#)

Antes de finalizarmos com os JTextFields adicionados anteriormente, precisamos nos certificar de que os dois JTextFields que inserimos entre os JLabels estejam definidos para se redimensionarem corretamente. Ao contrário dos dois JTextFields que foram esticados para a margem direita do form, o procedimento de redimensionamento dos componentes inseridos não é definido automaticamente.

Para definir o procedimento de redimensionamento de um componente:

1. Clique nos dois componentes JTextField inseridos para selecioná-los no GUI Builder.
2. Com os dois JTextFields selecionados, clique com o botão direito do mouse em um deles e escolha Redimensionamento automático > Horizontal no menu pop-up.

Os JTextFields serão definidos com redimensionamento horizontal em runtime. As linhas-guia de alinhamento e os indicadores de ancoragem serão atualizados, fornecendo comentário visual dos relacionamentos dos componentes.

Para configurar os componentes para que tenham o mesmo tamanho:

1. Clique em todos os quatro JTextFields do form para selecioná-los.
2. Com os JTextFields selecionados, clique com o botão direito do mouse em um deles e escolha Definir Mesmo Tamanho > Mesma Largura no menu pop-up.

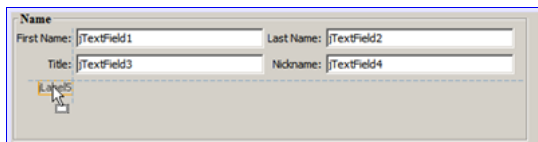
Todos os JTextFields terão a mesma largura definida e os indicadores serão adicionados na parte superior destes, fornecendo feedback visual dos relacionamentos dos componentes.

Agora precisamos adicionar outro JLabel que descreva a JComboBox que permitirá aos usuários selecionar o formato das informações que nossa aplicação ContactEditor exibirá.

Para alinhar um JLabel a um grupo de componentes:

1. Na janela Paleta, selecione o componente Label na categoria Swing.
2. Mova o cursor abaixo dos JLabels *Nome:* e *Título:* no lado esquerdo do JPanel. Quando as linhas-guia aparecerem, indicando que a nova borda direita do JLabel está alinhada às margens direitas do grupo de componentes acima (os dois JLabels), clique para posicionar o componente.

O JLabel se encaixará em uma posição alinhada à direita com a coluna dos JLabels acima, conforme mostra a ilustração abaixo. O GUI Builder atualiza as linhas de status de alinhamento, indicando as relações de espaçamento e ancoragem do componente.



Como nos exemplos anteriores, clique duas vezes no JLabel para selecionar o texto de exibição e insira *Exibir Formato:* para nome de exibição. Observe que quando o JLabel se encaixa na posição, os outros componentes mudam para acomodar o texto de exibição maior.

Alinhamento da Linha de Base

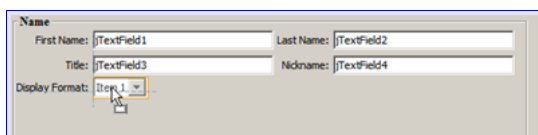
Quando você adiciona ou move componentes que incluem texto (JLabels, JTextFields, etc.), o IDE sugere alinhamentos baseados nas linhas de base do texto nos componentes. Quando inserimos o JTextField anteriormente, por exemplo, sua linha de base foi automaticamente alinhada aos JLabels adjacentes.

Agora adicionaremos a caixa de combinação que permitirá aos usuários selecionar o formato das informações que nossa aplicação ContactEditor exibirá. À medida que adicionamos JComboBox, alinharemos a linha de base com o texto do JLabel. Observe, mais uma vez as linhas-guia de alinhamento da linha de base aparece para nos auxiliar com o posicionamento.

Para alinhar as linhas de base dos componentes:

1. Na janela Paleta, selecione o componente Caixa de combinação na categoria Controles do Swing.
2. Mova o cursor à direita do JLabel recém-adicionado. Quando a linha-guia horizontal aparecer, indicando que a linha de base do JComboBox está alinhada à linha de base do texto do JLabel e o espaçamento entre os dois componentes for sugerido com uma linha-guia vertical, clique para posicionar a caixa de combinação.

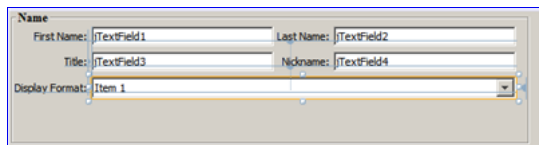
O componente se encaixará em uma posição alinhada com a linha de base do JLabel à esquerda, conforme mostra a ilustração abaixo. O GUI Builder exibe as linhas de status, indicando as relações de espaçamento e ancoragem do componente.



Para redimensionar o JComboBox:

1. Selecione a Caixa de Combinação no GUI Builder.
2. Arraste o handle de redimensionamento na margem direita do JComboBox até que as linhas-guia de alinhamento apareçam sugerindo o deslocamento preferencial entre as margens do JComboBox e do JPanel.

Conforme mostra a ilustração abaixo, a margem direita do JComboBox se encaixará em alinhamento com a margem recomendada do JPanel e a largura do componente será automaticamente definida para se redimensionar ao form.



3. Pressione Ctrl-S para salvar o arquivo.

A edição de modelos de componentes está além do escopo desse tutorial, portanto deixaremos, por enquanto, a lista de itens de espaço reservada da JComboBox como está.

[início](#)

Revisando o Que Aprendemos

Terminamos um bom início de construção da nossa GUI ContactEditor, mas tomemos um minuto para recapitular o que aprendemos enquanto adicionamos mais alguns componentes que nossa interface requer.

Até agora, nós nos concentramos na adição de componentes à nossa GUI ContactEditor utilizando as linhas-guia de alinhamento do IDE para nos ajudar no posicionamento. Entretanto, é importante compreender que a ancoragem é outra parte integrante do posicionamento de componentes. Embora ainda não tenhamos discutido isso, você já se beneficiou com essa funcionalidade sem perceber. Como mencionado antes, ao adicionar um componente a um form, o IDE sugere o posicionamento preferencial da aparência de destino com linhas-guia. Depois de posicionados, os novos componentes também são ancorados na margem mais próxima do contêiner ou componente para assegurar que os relacionamentos entre os componentes sejam mantidos em runtime. Nesta seção, nós nos concentraremos em concluir as tarefas de maneira mais otimizada, ao ressaltarmos o trabalho que o GUI Builder está fazendo nos bastidores.

Adicionando, Alinhando e Ancorando

O GUI Builder permite fazer o layout de forms rápida e facilmente otimizando gestos típicos do fluxo de trabalho. Quando você adiciona um componente a um form, o GUI Builder o encaixa automaticamente na posição preferencial e define as relações de encadeamento necessárias para que você se concentre no design dos forms em vez de conflitar com detalhes de implementação complicados.

Para adicionar, alinhar e editar o texto de exibição de um JLabel:

1. Na janela Paleta, selecione o componente Label na categoria Controles do Swing.
2. Mova o cursor sobre o form imediatamente abaixo do título E-mail do JPanel inferior. Quando as linhas-guia aparecerem, indicando que está posicionado no canto superior esquerdo do JPanel com uma pequena margem nos cantos superior e esquerdo, clique para posicionar o JLabel.
3. Clique duas vezes no JLabel para selecionar a exibição de texto. Depois digite `Endereço de E-mail:` e pressione Enter.

O JLabel se encaixará na posição preferencial no form e será ancorado nas margens superior e esquerda do JPanel de fechamento. Assim como ocorreu anteriormente, um nó correspondente, representando o componente, também será adicionado à janela Navegador.

Para adicionar um JTextField:

1. Na janela Paleta, selecione o componente Campo de texto na categoria Controles do Swing.
2. Mova o cursor imediatamente à direita do label `Endereço de E-mail` recém-adicionado. Quando as linhas-guia aparecerem, indicando que a linha de base do JTextField está alinhada à linha de base do texto no JLabel e a margem entre os dois componentes for sugerida com uma linha-guia vertical, clique para posicionar o campo de texto.

O JTextField se encaixará à direita do JLabel `Endereço de E-mail`: e será encadeado ao JLabel. O nó correspondente também será adicionado à janela Inspetor.

3. Arraste o handle de redimensionamento do JTextField em direção à direita do JPanel até que as linhas-guia de alinhamento sejam exibidas sugerindo o deslocamento entre as margens do JTextField e do JPanel.

A margem direita do JTextField se encaixará na linha-guia de alinhamento indicando as margens preferenciais.

Agora precisamos adicionar o JList que exibirá a lista de contatos completa do ContactEditor.

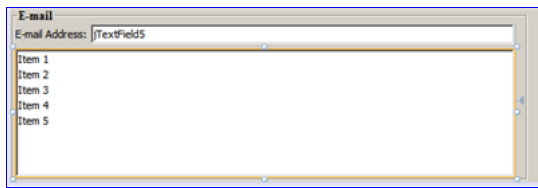
Para adicionar e redimensionar um JList:

1. Na janela Paleta, selecione o componente Lista na categoria Controles do Swing.
2. Mova o cursor imediatamente abaixo do JLabel `Endereço de E-mail` adicionado anteriormente. Quando as linhas-guia aparecerem, indicando que as margens superior e esquerda do JList estão alinhadas às margens preferenciais ao longo da

margin esquerda do JPanel e do JLabel acima, clique para posicionar o JList.

3. Arraste o handle de redimensionamento do JList em direção à direita do JPanel de fechamento até que as linhas-guia de alinhamento sejam exibidas indicando que tem a mesma largura do JTextField acima.

O JList se encaixará na posição designada pelas linhas-guia de alinhamento e seu nó correspondente será exibido na janela Inspetor. Observe também que o form se expande para acomodar o JList recém-adicionado.



Como as JLists são utilizadas para exibir longas listas de dados, geralmente requerem a adição de um JScrollPane. Quando você adiciona um componente que requer um JScrollPane, o GUI Builder o adiciona automaticamente para você. Como os JScrollPane são componentes não visuais, é necessário utilizar a janela Inspetor para exibir ou editar os JScrollPane que o GUI Builder criou.

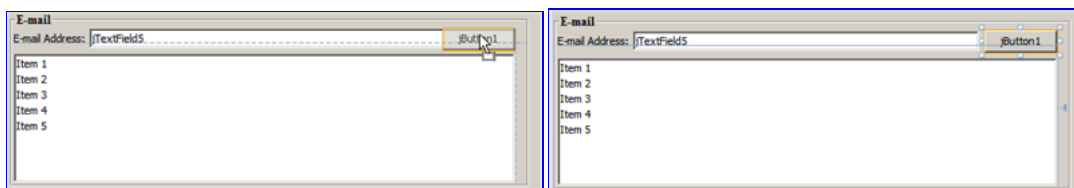
Dimensionamento de Componentes

Observação: Consulte o screencast [Redimensionando e recuando componentes \(.swf\)](#) para ver uma demonstração interativa na seção abaixo.

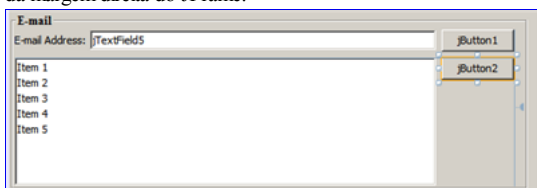
É sempre benéfico definir vários componentes relacionados, como botões em caixas de diálogo modais, para que tenham o mesmo tamanho a fim de oferecer consistência visual. Para demonstrar isso, adicionaremos quatro JButtons ao nosso form ContactEditor que permitirão adicionar, editar e remover entradas individuais da nossa lista de contatos, como mostram as ilustrações abaixo. Depois, definiremos os quatro botões para que tenham o mesmo tamanho e possam ser facilmente reconhecidos e ofereçam a funcionalidade relacionada.

Para adicionar, alinhar e editar o texto de exibição de vários botões:

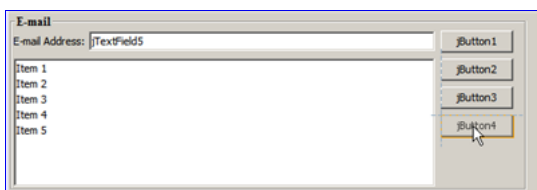
1. Na janela Paleta, selecione o componente Botão na categoria Controles do Swing.
2. Mova o JButton acima da margem direita do JTextField Endereço de E-mail no JPanel inferior. Quando as linhas-guia aparecerem, indicando que a linha de base e a margem direita do JButton estão alinhadas às margens do JTextField, clique pressionando Shift para posicionar o primeiro botão ao longo da margem direita do JFrame. A largura do JTextField diminui para acomodar o JButton onde você soltar o botão do mouse.



3. Mova o cursor sobre o canto superior direito do JList no JPanel inferior. Quando as linhas-guia aparecerem, indicando que as margens superior e direita do JButton estão alinhadas às margens do JList, clique pressionando Shift para posicionar o segundo botão ao longo da margem direita do JFrame.

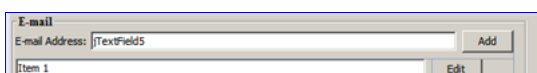


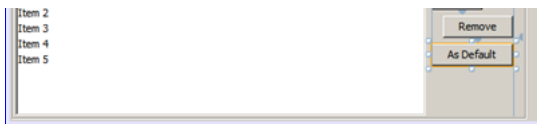
4. Adicione dois outros JButtons abaixo dos dois já adicionados para criar uma coluna. Certifique-se de posicionar os JButtons de maneira que o espaçamento sugerido seja respeitado e consistente. Se você se esquecer de soltar a tecla Shift antes de posicionar o último JButton, basta pressionar a tecla Escape.



5. Defina o texto de exibição de cada JButton. (Você pode editar o texto de um botão, clicando com o botão direito do mouse no botão e escolhendo Editar Texto. Ou você pode clicar no botão, pausar e clicar novamente.) Insira Adicionar para o botão superior, Editar para o segundo, Remover para o terceiro e Default para o quarto.

Os componentes JButton se encaixam nas posições designadas pelas linhas-guia de alinhamento. A largura dos botões muda para acomodar os novos nomes.



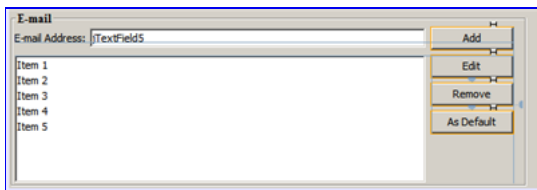


Agora que os botões estão posicionados onde queremos, nós definiremos os quatro botões para que tenham o mesmo tamanho e consistência visual, bem como para esclarecer a funcionalidade relacionada.

Para definir componentes do mesmo tamanho:

1. Selecione os quatro JButtons pressionando a tecla Control enquanto faz a seleção.
2. Clique com o botão direito do mouse em um deles e escolha Mesmo Tamanho > Mesma Largura no menu pop-up.

Os JButtons serão definidos com o mesmo tamanho do botão que tiver o nome mais longo.



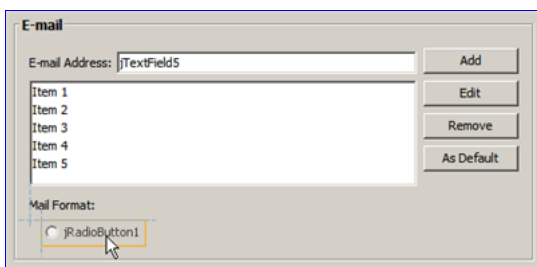
Recuo

Frequentemente é necessário unir vários componentes em outro para que fique claro que pertencem a um grupo de funções relacionadas. Um caso típico, por exemplo, é colocar várias caixas de seleção relacionadas abaixo de um label comum. O GUI Builder permite fazer o recuo facilmente oferecendo linhas-guia especiais que sugerem o deslocamento preferencial para a aparência e funcionalidade do seu sistema operacional.

Nesta seção, adicionaremos alguns JRadioButtons abaixo de um JLabel que permitirá aos usuários personalizar a maneira como a aplicação exibe dados. Veja as ilustrações abaixo enquanto faz isso ou clique no link Exibir Demonstração seguindo o procedimento para ver uma demonstração interativa.

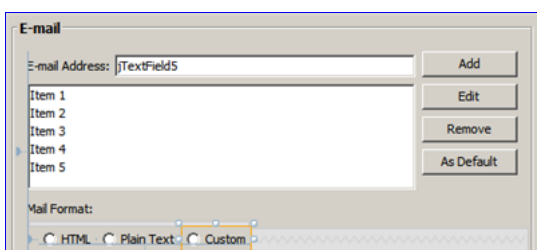
Para recuar JRadioButtons abaixo de um JLabel:

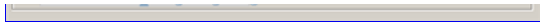
1. Adicione um JLabel nomeado `Formato de Correspondência` ao form abaixo do JList. Certifique-se de que o label esteja alinhado à esquerda do JList acima.
2. Na janela Paleta, selecione o componente Botão de Opção na categoria Swing.
3. Mova o cursor abaixo do JLabel recém-adicionado. Quando as linhas-guia aparecerem, indicando que a borda esquerda do JRadioButton está alinhada com a do JLabel, mova o JRadioButton ligeiramente para a direita até as linhas-guia de recuo secundárias aparecerem. Clique pressionando a tecla Shift para posicionar o primeiro botão de opção.



4. Mova o cursor à direita do primeiro JRadioButton. Clique pressionando a tecla Shift para inserir o segundo e o terceiro JRadioButtons, respeitando o espaçamento sugerido do componente. Certifique-se de soltar a tecla Shift antes de posicionar o último JRadioButton.
5. Defina o texto de exibição de cada JRadioButton. (Você pode editar o texto de um botão, clicando com o botão direito do mouse no botão e escolhendo Editar Texto. Ou você pode clicar no botão, pausar e clicar novamente.) Insira `HTML` para o botão de opção esquerdo, `Texto Simples` para o segundo e `Personalizado` para o terceiro.

Três JRadioButtons serão adicionados ao form e recuados abaixo do JLabel `Formato de Correio`.



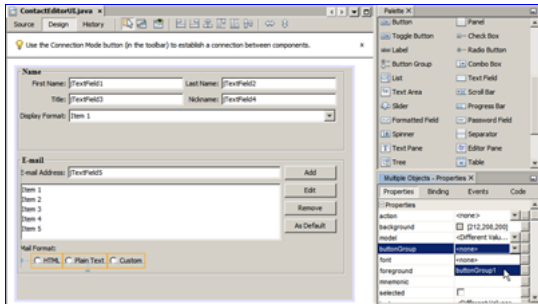


Agora, precisamos adicionar os três JRadioButtons a um ButtonGroup para permitir o comportamento de alternância esperado, no qual somente um botão de opção pode ser selecionado por vez. Por sua vez, isso assegurará que as informações de contato da aplicação ContactEditor serão exibidas no formato de e-mail da nossa escolha.

Para adicionar JRadioButtons a um ButtonGroup:

1. Na janela Paleta, selecione o componente Grupo de Botões na categoria Controles do Swing.
2. Clique em qualquer lugar na área de design do GUI Builder para adicionar o componente ButtonGroup ao form. Observe que o ButtonGroup em si não é exibido no form, no entanto, é visível na área Outros Componentes do Navegador.
3. Selecione os três JRadioButtons do form.
4. Na janela Propriedades, selecione buttonGroup1 na caixa de combinação da propriedade buttonGroup.

Três JRadioButtons serão adicionados ao grupo de botões.



5. Pressione Ctrl-S para salvar o arquivo.

[início](#)

Fazendo os Ajustes Finais

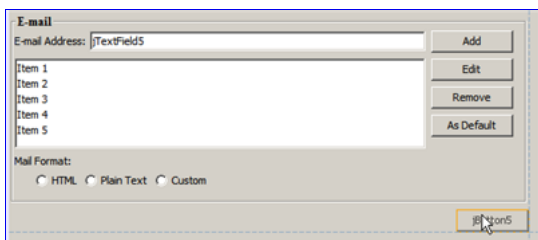
Conseguimos esboçar nossa GUI da aplicação ContactEditor, mas ainda há algumas coisas a serem feitas. Nesta seção, observaremos algumas das tarefas de layout típicas que o GUI Builder simplifica.

Finalizando

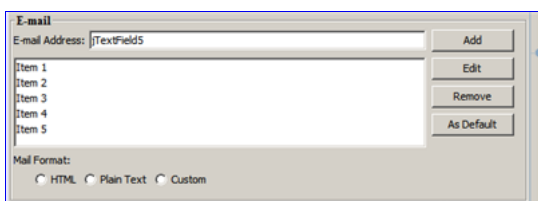
Agora precisamos adicionar os botões que permitirão aos usuários confirmar as informações que inserirem de um contato individual e adicioná-las à lista de contatos ou cancelar, deixando o banco de dados inalterado. Nesta etapa, adicionaremos os dois botões requeridos e os editaremos para que sejam exibidos com o mesmo tamanho no form, embora seus textos de exibição tenham tamanhos diferentes.

Para adicionar e editar o texto de exibição dos botões:

1. se o JPanel inferior estiver estendido na margem inferior do form JFrame, arraste a margem inferior do JFrame para baixo. Isso proporciona espaço entre as margens do JFrame e do JPanel para os botões OK e Cancelar.
2. Na janela Paleta, selecione o componente Botão na categoria Controles do Swing.
3. Mova o cursor sobre o form abaixo do JPanel E-mail. Quando as linhas-guia aparecerem, indicando que a borda direita do JButton está alinhada ao canto direito do JFrame, clique para posicionar o botão.



4. Adicione outro JButton à esquerda do primeiro, certificando-se de colocá-lo utilizando o espaçamento sugerido ao longo da margem inferior do JFrame.
5. Defina o texto de exibição de cada JButton. Insira OK para o botão esquerdo e Cancelar para o botão direito. Observe que a largura dos botões muda para acomodar os novos nomes.
6. Defina os dois JButtons para terem o mesmo tamanho selecionando-os, clicando com o botão direito do mouse em um deles e selecionando Mesmo Tamanho > Mesma Largura no menu pop-up.






Os componentes `JButton` serão exibidos no form e seus nós correspondentes serão exibidos na janela Navegador. O código dos componentes `JButton` também será adicionado ao arquivo de código-fonte do form, que é visível na view Código-fonte do Editor. Todos os `JButtons` serão definidos com o mesmo tamanho do botão que possuir o maior nome.

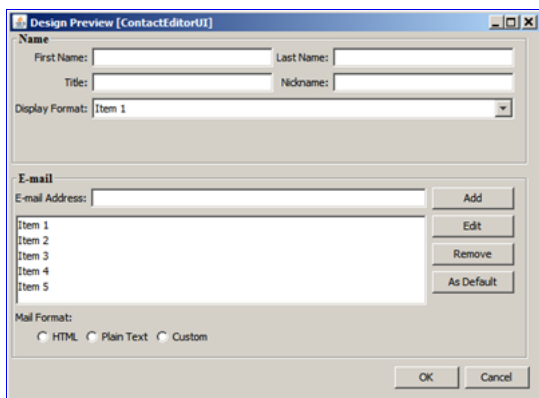
7. Pressione Ctrl-S para salvar o arquivo.

A última coisa que precisamos fazer é deletar o texto do espaço reservado nos vários componentes. Observe que, embora a remoção do texto do placeholder após a configuração preliminar de um form possa ser uma técnica útil para evitar problemas de alinhamento e relações de ancoragem dos componentes, a maioria dos desenvolvedores geralmente remove esse texto no processo de layout de seus forms. À medida que percorre o form, selecione e delete o texto do espaço reservado de cada um dos `JTextFields`. Deixaremos os itens de placeholder no `JComboBox` e no `JList` para um tutorial posterior.

[início](#)

Visualizando a GUI

Agora que você criou com êxito a GUI `ContactEditor`, é possível experimentar a interface para ver os resultados. É possível visualizar o form à medida que trabalha clicando no botão Visualizar Form () na barra de ferramentas do GUI Builder. O form será aberto em sua própria janela, permitindo que você o teste antes da construção e execução.



[início](#)

Implantando Aplicações GUI

Para que as interfaces criadas com o GUI Builder funcionem fora do IDE, a aplicação deve ser compilada nas classes do gerenciador de layout `GroupLayout` e essas classes também devem estar disponíveis no runtime. Essas classes são incluídas no Java SE 6, mas não no Java SE 5. Se você desenvolver a aplicação para ser executada no Java SE 5, sua aplicação precisa utilizar a biblioteca de extensões de layout Swing.

Se o IDE estiver sendo executado no JDK 5, o IDE gerará o código da aplicação automaticamente para utilizar a biblioteca de extensões de layout Swing. Quando você implanta a aplicação, é necessário incluir a biblioteca de extensões de layout Swing. Quando você constrói a aplicação (Construir > Construir Projeto Principal), o IDE fornece automaticamente uma cópia do arquivo JAR da biblioteca na pasta `dist/lib` da aplicação. O IDE também adiciona todos os arquivos JAR da pasta `dist` no elemento `Class-Path` no arquivo `manifest.mf` do arquivo JAR da aplicação.

Se você estiver executando o IDE no JDK 6, o IDE gera o código de sua aplicação para utilizar as classes `GroupLayout` que estão no Java SE 6. Isso significa que é possível implantar a aplicação para ser executada em sistemas com o Java SE 6 instalado e não é necessário encapsular sua aplicação com a biblioteca de extensões de layout Swing.

Observação: se a aplicação for criada com o JDK 6, mas você precisar que também seja executada no Java SE 5, poderá fazer com que o IDE gere seu código para utilizar a biblioteca de extensões de layout Swing em vez das classes no Java SE 6.

Abra a classe `ContactEditorUI` no Editor de GUI. No Navegador, clique com o botão direito do mouse no Form `ContactEditorUI` e escolha Propriedades no menu pop-up. Na caixa de diálogo Propriedades, altere o valor da propriedade Estilo de Geração de Layout para Biblioteca de Extensões de Layout Swing.

Distribuindo e Executando Aplicações GUI Standalone

Para preparar a aplicação de GUI para distribuição fora do IDE:

- Compacte a pasta `dist` do projeto em um arquivo compactado ZIP. (A pasta `dist` também pode conter uma pasta `lib`, que também poderia necessitar ser incluída).

Para executar sua aplicação, clique com o botão direito do mouse no nome do projeto e selecione Executar a partir do menu de contexto. Na caixa de diálogo Executar Projeto, selecione a classe principal (`my.contacteditor.ContactEditorUI`, se estiver falando sobre o projeto que você acabou de criar) e clique em OK. Sua aplicação está ativa e em execução.

Para executar uma aplicação GUI standalone na linha de comandos:

1. Navegue até a pasta `dist` do projeto.
2. Digite o seguinte:

```
java -jar <nome do jar>.jar
```

Observação: se você encontrar este erro:

```
Exception in thread "main" java.lang.NoClassDefFoundError: org/jdesktop/layout/GroupLayout$Group
```

Verifique se o arquivo `manifest.mf` faz referência à versão atualmente instalada da Biblioteca de extensões de Layout Swing.

[início](#)

[Enviar Feedback neste Tutorial](#)

Consulte Também

Agora você concluiu o tutorial Criando uma GUI Swing. Para obter informações sobre como adicionar funcionalidade às GUIs que você cria, consulte:

- [Introdução à Construção de GUIs](#)
- [Manipulando Imagens em uma Aplicação da GUI](#)
- [FAO do GUI Builder](#)
- [Trilha de Aprendizado das Aplicações de GUI do Java](#)
- [Implementando GUIs de Java](#) em *Desenvolvendo Aplicações com o NetBeans IDE*

[início](#)



- [SiteMap](#)
- [About Us](#)
- [Contact](#)
- [Legal & Licences](#)

By use of this website, you agree to the [NetBeans Policies and Terms of Use](#). © 2013, Oracle Corporation and/or its affiliates. Sponsored by **ORACLE**