

Desenvolvimento Desktop

Aula 04 - Componentes: Campo de Texto (JTextField) e Bot o (JButton)

Apresentação

Nesta aula continuaremos a utilizar os componentes Swing e realizar exemplos e implementações de fácil assimilação com eles. Na aula anterior, discutimos o uso do componente **JLabel**, considerado o mais popular de todos, e agora vamos explorar mais alguns, também considerados indispensáveis em qualquer aplicação, como o Campo de texto (**JTextField**) e o Botão (**JButton**), e, à medida que formos progredindo nas aulas, conheceremos outros componentes com finalidades mais específicas.



Vídeo 01 - Apresentação

Objetivos

Nesta aula, você será capaz de:

- Reconhecer e utilizar os componentes Swing.
- Aplicar os componentes Swing em ambientes gráficos.
- Distinguir propriedades gerais e específicas dos componentes Swing.

Utilizando os Componentes JTextField e JButton

Nesta aula você verá em detalhes como utilizar e implementar dois dos mais utilizados componentes em aplicações Java: o Campo de texto (**JTextField**) e o Botão (**JButton**). É praticamente impossível se desenvolver uma aplicação sem a presença desses dois componentes. O primeiro é utilizado para o usuário entrar com uma informação qualquer, como nome, endereço, telefone etc., e o segundo é muito utilizado para enviar essas informações ou confirmar ou negar uma pergunta de acordo com a situação. Você deve ter observado o uso desses componentes em várias janelas do próprio Windows. As novas versões do NetBeans traduzem os componentes para o português na exibição da paleta de projeto. **JTextField** = Campo de Texto. **JButton** = Button (ou Botão). O nome da Classe, claro, permanece em inglês. Vejamos, então, como utilizar os componentes dessa aula.

O Componente Campo de Texto (JTextField)

Esse componente é um dos mais básicos, muito comum em formulários ou em interfaces onde o usuário precisa digitar um texto com uma informação a ser enviada, como, por exemplo: nome, telefone e e-mail em um cadastro pessoal. Existem outros componentes que também tratam de informação do tipo texto, alguns deles serão vistos mais adiante nas próximas aulas.

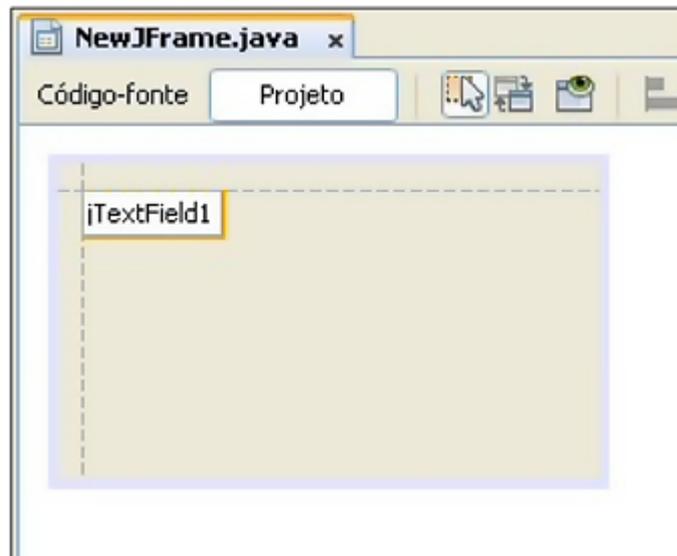
Como Implementar um Campo de Texto na minha Aplicação?

Vejamos como fazer isso de uma forma bem simples:

1. Execute o NetBeans e crie uma aplicação Java, conforme foi explicado na aula "**Introdução ao Swing**". Para o nome do projeto informe: **TextFieldDemo** (apenas uma sugestão, pode ser o nome que você achar mais conveniente) e, em seguida, adicione um frame (**JFrame**) ao projeto. Dê o mesmo nome para o formulário.

2. Após o projeto ser criado, selecione o **JFrame** e arraste um componente **Campo de texto** da paleta à direita para qualquer posição da tela, conforme mostra a **Figura 1**:

Figura 01 - Componente JTextField inserido no painel principal



Agora vamos ver mais detalhes e começar a explorar algumas propriedades, que também podem ser comuns a alguns outros componentes.

Para configurar algumas características do campo de texto, observe as opções disponíveis no painel **Propriedades** da janela logo abaixo da **paleta de componentes (Figura 2)**.

Figura 02 - Propriedades do componente campo de texto (JTextField)



3. Na lista de propriedades disponíveis, você poderá personalizar o componente e deixá-lo do seu jeito. É como se você fosse comprar um carro em uma loja. Você poderia escolher a cor, o modelo, direção hidráulica, ar condicionado etc. Nos componentes do NetBeans é a mesma coisa. Você poderá alterar o rótulo, o estilo, o tamanho e a cor da fonte, a cor de fundo, entre outras características. Vejamos como alterar algumas dessas propriedades na prática:

Para iniciar, vamos ver as seguintes propriedades

- **background** – permite alterar a cor do plano de fundo do componente, disponibilizando a especificação da cor em diversos padrões.
- **editable** – se esse checkbox estiver marcado, o componente ficará habilitado para a inserção e edição do texto, caso contrário, ficará desabilitado. Essa propriedade é marcada por padrão.

- **font** – permite alterar a fonte, o estilo e o tamanho da fonte de um componente.
 - **foreground** – permite alterar a cor do texto do componente, disponibilizando a especificação da cor em diversos padrões.
 - **text** – essa propriedade é utilizada para você especificar um texto para o componente quando a aplicação for executada, ao invés de vir em branco. Por exemplo, você poderia informar ao usuário que nessa caixa de texto ele deverá inserir o nome de sua cidade de nascimento, mas o campo já poderia vir selecionado com uma cidade padrão, como, por exemplo, "**Natal**".
 - **tamanho horizontal e tamanho vertical** – utilizadas para alterar as dimensões de largura e altura do componente, permitindo mudar o tamanho padrão (quando o componente é inserido no **JFrame**) tanto digitando os valores nas caixas de texto, como diretamente no componente arrastando as abas de redimensionamento.
 - **opaque** – a cor de fundo que você escolheu na propriedade **background** só será mostrada se essa propriedade estiver marcada, caso contrário, o componente será exibido com o fundo padrão.
-

Como Alterar o Rótulo

1. Clique no **campo de texto** e, no painel **Propriedades**, clique na opção **text**, e altere o seu rótulo para Teste, conforme mostra a **Figura 3**:

Figura 03 - Alterando a propriedade text do componente campo de texto



2. Esse texto é o que aparecerá no componente **campo de texto**. Na maioria das vezes, para esse componente, esse campo fica em branco, pois é onde o usuário insere a informação.

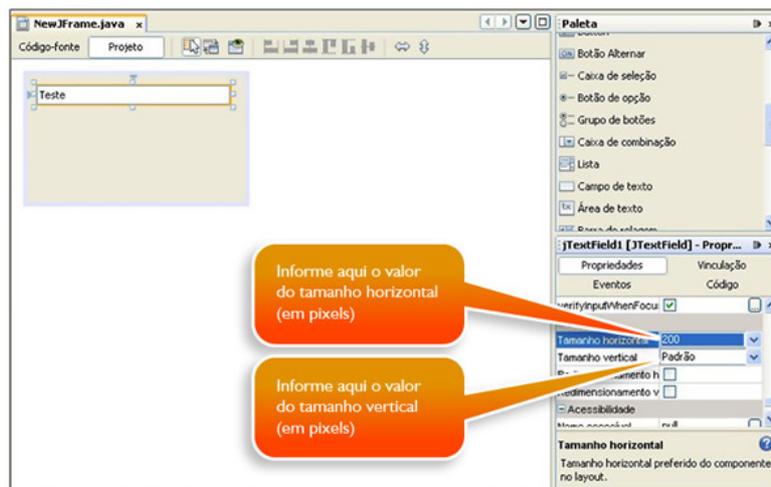
Observação

Essa é uma das propriedades que será vista novamente, mais adiante, incluindo alguma particularidade do componente apresentado.

Como Alterar a Largura e a Altura do Campo de Texto

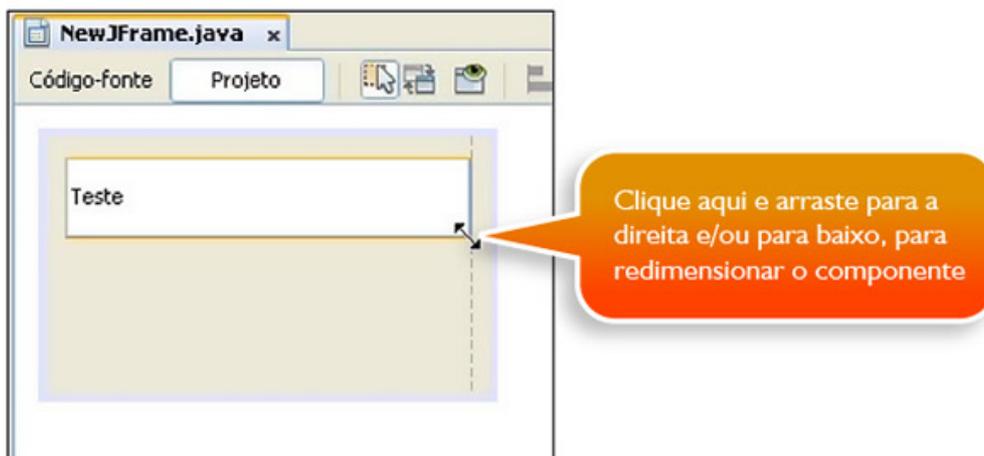
1. Para alterar as dimensões de um **campo de texto** (como também de alguns outros componentes), você dispõe de duas opções, conforme mostramos a seguir:
 - a. Utilizando as propriedades **Tamanho horizontal** e **Tamanho vertical**, e digitando diretamente no painel **Propriedades** os seus respectivos valores (em pixels), conforme mostra a **Figura 4**:

Figura 04 - Opções para alteração das dimensões do componente campo de texto (JTextField)



b. Ou arrastando as alças de dimensionamento do componente, visualmente conforme mostra a **Figura 5**:

Figura 05 - Dimensionando um campo de texto visualmente



Observação

Qualquer propriedade de um componente do NetBeans poderá ser alterada a qualquer momento e em qualquer situação que você desejar durante o projeto. As escolhas não são definitivas.

Como Alterar a Cor de Fundo do Campo de Texto (Background)

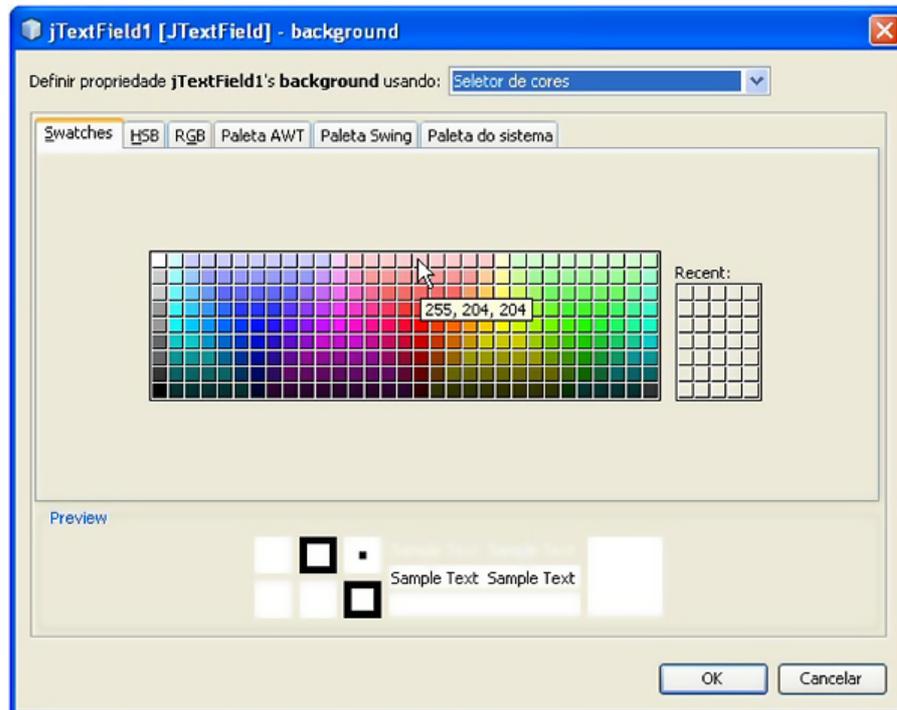
1. Para alterar a cor de fundo de um campo de texto (como também de alguns outros componentes), localize a propriedade **background** na paleta **Propriedades**, conforme mostramos na **Figura 6**:

Figura 06 - Propriedade para alterar a cor de fundo de um campo de texto



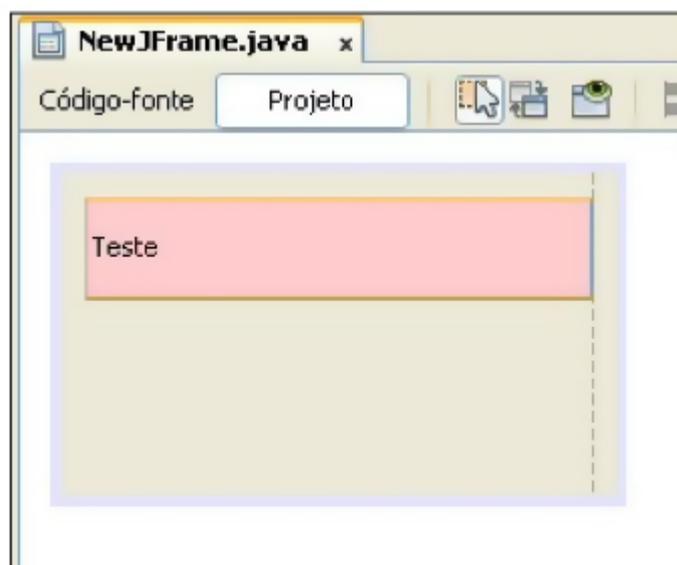
2. Clique no pequeno quadrado à direita. A seguinte janela estará disponível para a escolha da nova cor conforme a **Figura 7**:

Figura 07 - Definindo a cor de fundo para um campo de texto



3. Clique na cor escolhida e, em seguida, no botão **OK**. Observe que a cor foi alterada (**Figura 8**):

Figura 08 - Cor de fundo alterada



Observação

Você também pode alterar as propriedades diretamente no painel de **Propriedades** sem necessidade de abrir a janela de diálogo, ou seja, no caso da propriedade **background**, se você já souber o código da cor que deseja, basta digitar diretamente o valor de cada cor “[255,204,204]” e apertar **Enter**. A cor do plano de fundo será alterada para a cor especificada.



Vídeo 02 - JTextField

Atividade 01

1. Analise cada uma das situações a seguir relacionadas com o componente **JTextField** e responda V (para Verdadeira) ou F (para Falsa):
 - a. Se a propriedade `editable` estiver marcada significa que o texto do componente pode ser alterado. ()
 - b. A cor de fundo selecionada na propriedade **background** só é visualizada se a propriedade **opaque** estiver marcada. ()
 - c. A propriedade **font** permite alterar a fonte, o estilo e o tamanho da fonte de um componente. ()
 - d. Esse tipo de componente não permite que o usuário digite números na sua propriedade **text**. ()
2. Para alterarmos a cor do texto de um **campo de texto**, utilizamos a propriedade **text**:
 - a. Verdadeiro
 - b. Falso

3. Crie uma aplicação utilizando um componente **campo de texto**, que contenha o texto "**Digite sua resposta aqui.**", tamanho vertical de 144 *pixels*, tamanho horizontal de 198 pixels e o plano de fundo na cor verde '[153,255,51]'.

O Componente Botão (JButton)

Você, ao utilizar programas com interface gráfica em ambientes como Windows e Linux, por exemplo, certamente, já se deparou com botões em suas interfaces. Portanto, o seu uso é bastante óbvio, pois os botões permitem a ativação de certas ações, como por exemplo, realizar confirmações (normalmente, através do botão **OK**), cancelar ações (**Cancel**), abrir novas janelas, enviar um e-mail, enviar um texto em um programa de mensagem instantânea (tipo msn) e assim por diante. É bom lembrar que algumas propriedades, já discutidas anteriormente nos componentes **JLabel** e **TextField**, se comportam da mesma forma para outros componentes como o **Button**. Em vista disso, não vamos discuti-las novamente, apenas utilizá-las nas aplicações e nos exercícios.



Vídeo 03 - JButton

Mas como Implementar um Botão na minha Aplicação?

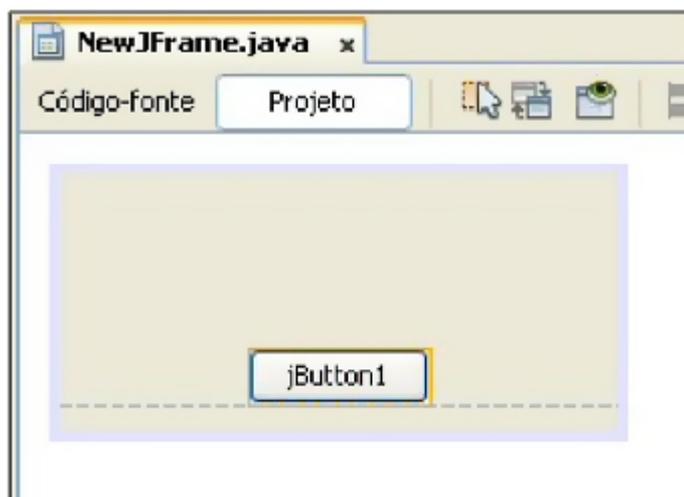
1. Primeiramente, crie uma aplicação e adicione um Frame (**JFrame**).
2. Em seguida, procure na paleta de componentes à direita o ícone do botão (**Button**). Veja a **Figura 9**:

Figura 09 - Paleta com o componente JButton em destaque.



3. Clique no botão e arraste-o para alguma posição dentro do Frame. Essa posição será determinada pelo gerenciador de layout associado ao frame (**Figura 10**). Os gerenciadores de layout disponíveis serão vistos mais adiante no curso.

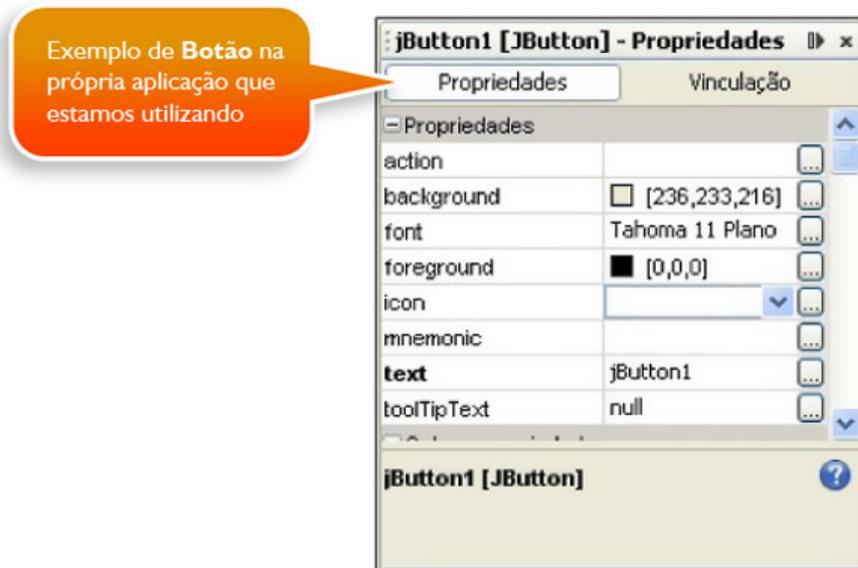
Figura 10 - Inserindo um botão no JFrame.



4. Uma vez posicionado o botão, você poderá configurar diversas propriedades dele e, também, realizar diversas ações com o botão. Veja também que a classe que implementa um botão, na API Java, é a **JButton**.
5. Para configurar as características do botão, observe as opções disponíveis no painel **Propriedades** da janela logo abaixo da **paleta de componentes**. Observe que a própria interface gráfica do NetBeans usa esses

componentes, como o botão **Propriedades** que acabamos de ver na **Figura 11**.

Figura 11 - Propriedades do componente Botão (JButton).



6. Na lista de propriedades disponíveis você poderá personalizar o seu botão e deixá-lo do seu jeito. Além das propriedades que já vimos anteriormente, vamos ver mais algumas, incluindo propriedades específicas desse componente:

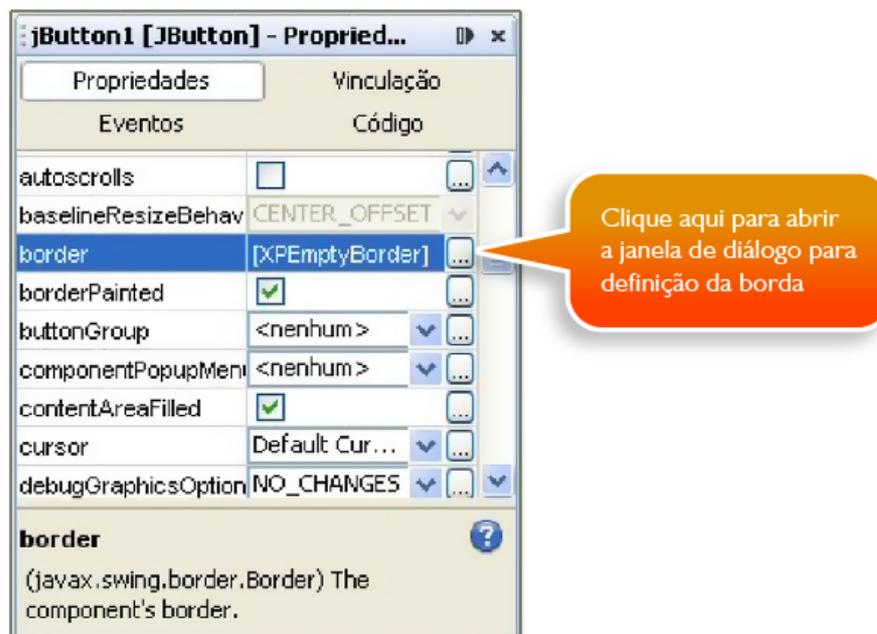
- **border** – essa propriedade permite definir uma borda para o componente, possibilitando a escolha do tipo de linha, cor e espessura da borda.
- **enabled** – se esse checkbox estiver marcado, significa que o componente ficará habilitado, ou seja, no caso do botão, poderá ser clicado, caso contrário, ficará desabilitado e não será executada nenhuma ação definida para esse componente. Por padrão, esse checkbox já é marcado.
- **icon** – essa propriedade permite inserir um ícone no componente.
- **toolTipText** – essa propriedade permite incluir um texto explicativo que aparece para o usuário quando o cursor passar por cima do componente.

Personalizar uma Borda

Pode ser usada para dar destaque ao componente. Para alguns componentes, essa propriedade pode ser essencial, como, por exemplo, para **rótulos** que não contenham um texto inicial, usados para apresentar o resultado de uma ação (vamos ver um exemplo prático ainda nesta aula).

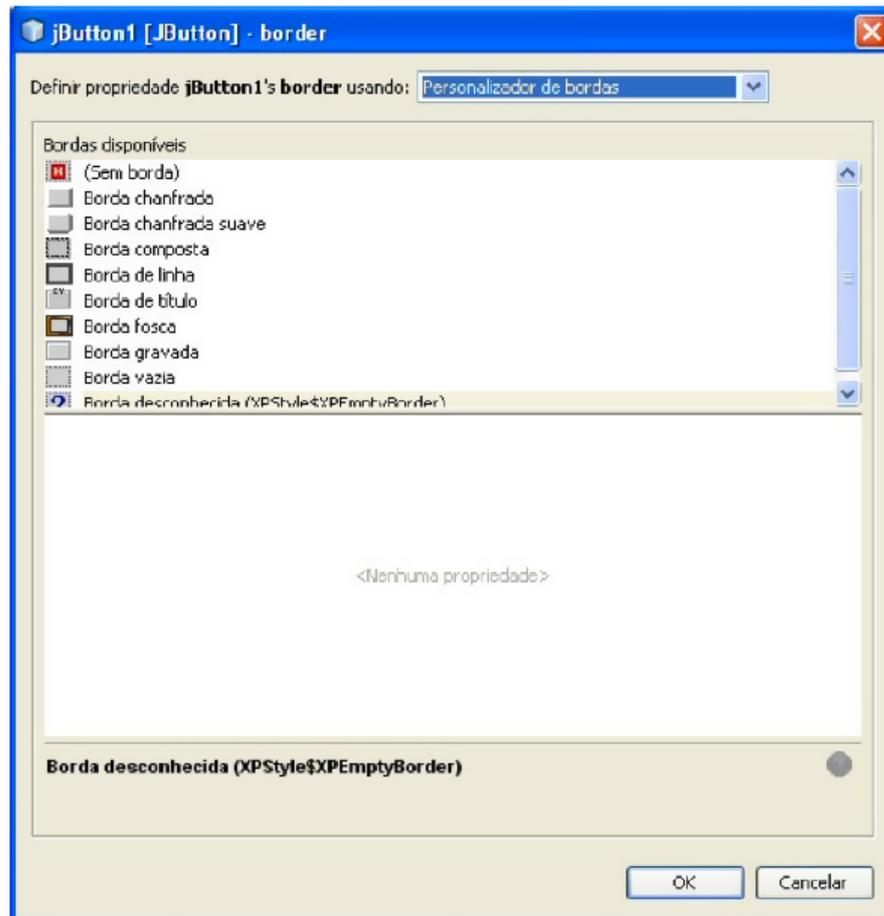
1. Clique na opção **border** e, em seguida, clique no pequeno botão (de reticências) à sua direita para abrir a caixa de diálogo, conforme mostra a **Figura 12**:

Figura 12 - Alterando a propriedade "border" de um componente botão.



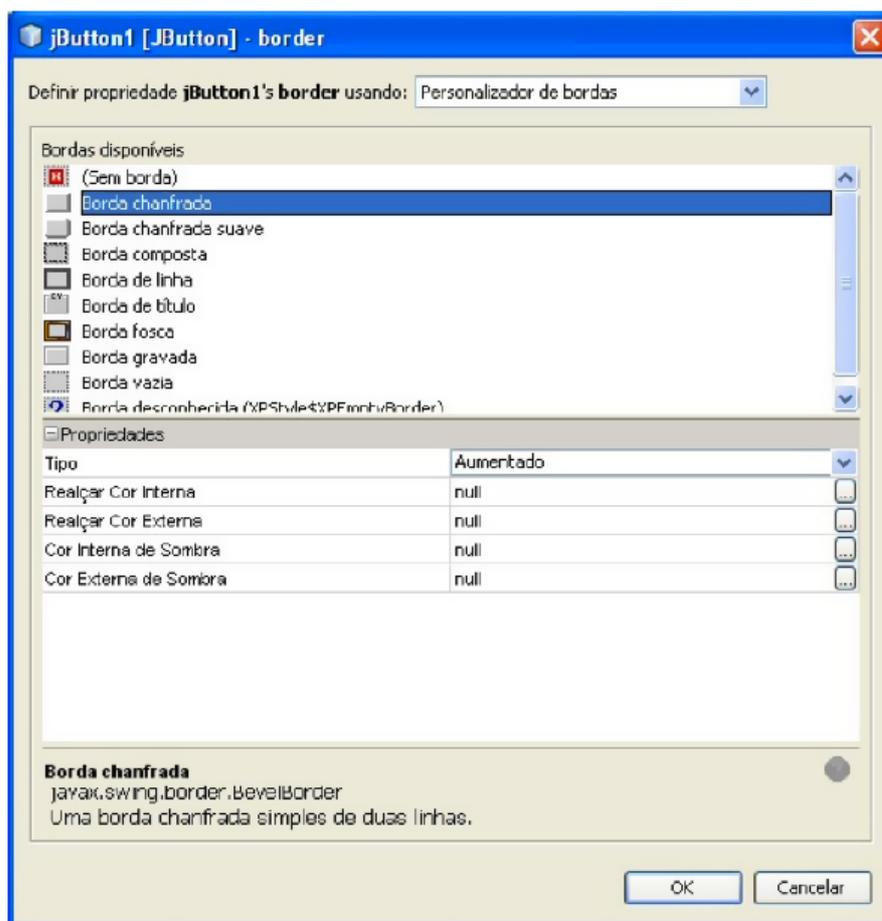
2. Uma janela similar à **Figura 13** aparecerá. Escolha o tipo de borda entre os disponíveis na lista.

Figura 13 - Escolhendo o tipo de borda



3. Ao clicar em um dos tipos de borda, algumas características que podem ser configuradas aparecerão listadas logo abaixo. Veja a **Figura 14**.

Figura 14 - Propriedades do tipo de borda selecionado



4. A partir daí varie as propriedades de acordo com o desejado e clique no botão **OK**. Observe as mudanças (**Figura 15**):

Figura 15 - Botão com a borda alterada



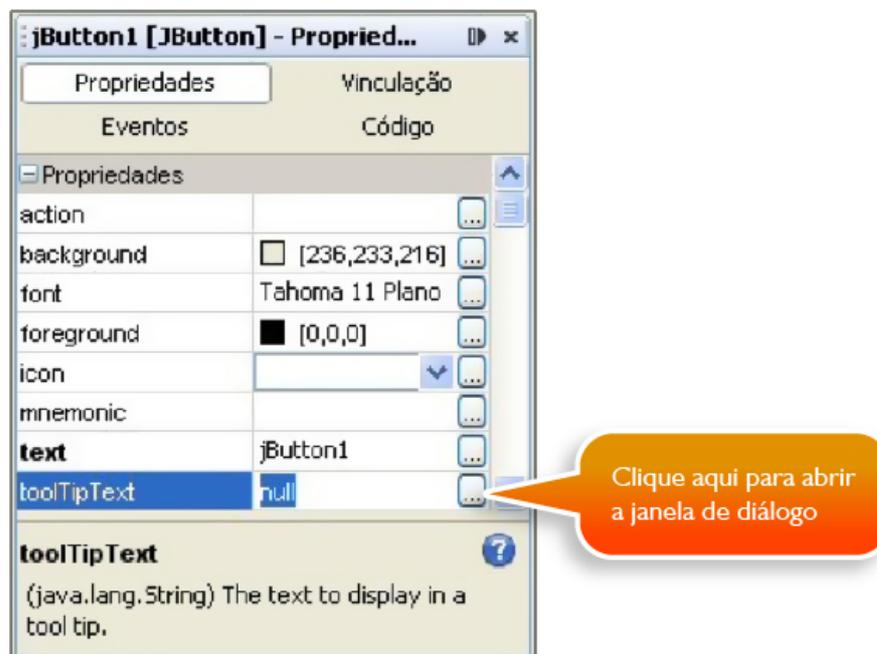
Para ver o resultado final, execute a aplicação (atalho: tecla **F6**).

Incluir um Comentário

Essa propriedade é bastante utilizada em diversos programas. Ela disponibiliza uma informação adicional sobre o componente em questão, sem que essa informação esteja fixa na tela, ocupando espaço. Ela somente se torna visível quando o usuário coloca o cursor sobre o componente.

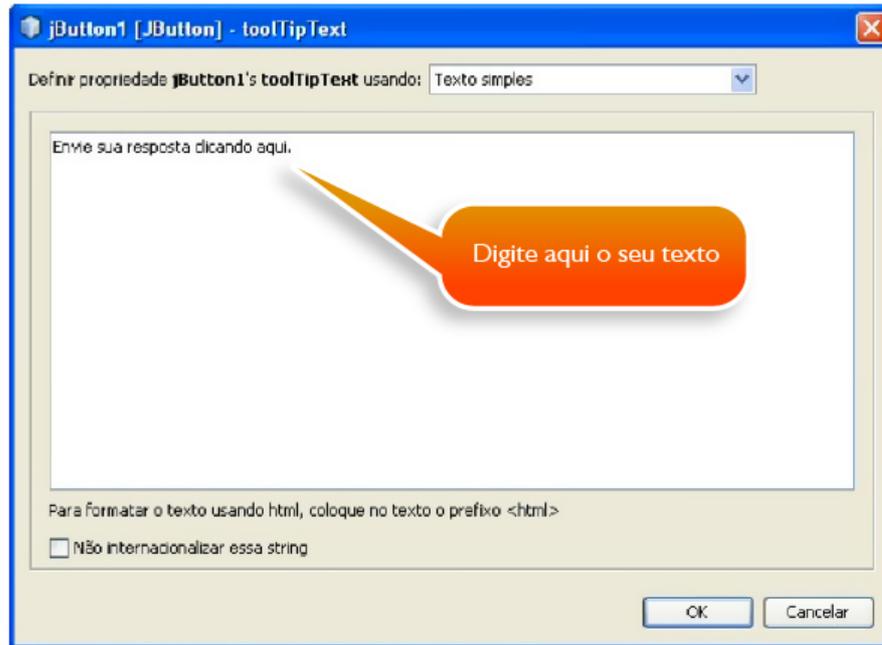
1. Clique no componente e localize a opção **toolTipText** e, em seguida, clique no pequeno botão (de reticências) à sua direita para abrir a caixa de diálogo, conforme mostra a **Figura 16**:

Figura 16 - Alterando a propriedade toolTipText.



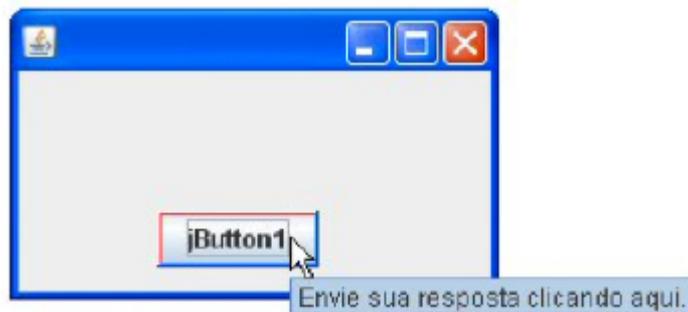
2. Uma janela similar à **Figura 17** aparecerá. Clique na área de texto indicada na figura e digite o texto informativo.

Figura 17 - Inserindo o texto



3. Lembre que o texto deve conter uma informação clara e curta. O usuário deve entender a mensagem com poucas palavras, facilitando a navegação na tela.
4. Execute a aplicação (**F6**) e confira o resultado com a **Figura 18**:

Figura 18 - Resultado da propriedade toolTipText



Para Alterar o Estado de Habilitado e Desabilitado (Enabled)

Essa propriedade também é bastante utilizada. Ela permite controlar o estado de disponibilidade do componente impedindo sua utilização quando desabilitado. Em algumas situações, faz-se necessário controlar o estado dos componentes, a fim de evitar erros.

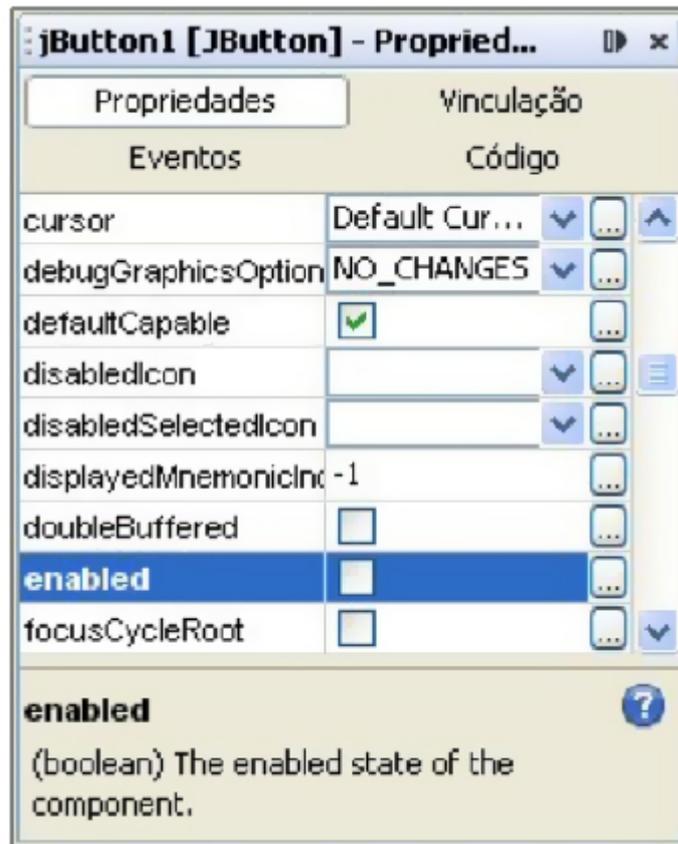
1. Localize a opção **enabled** no painel de **Propriedades**, conforme mostra a **Figura 19**:

Figura 19 - Utilizando a propriedade "enabled".



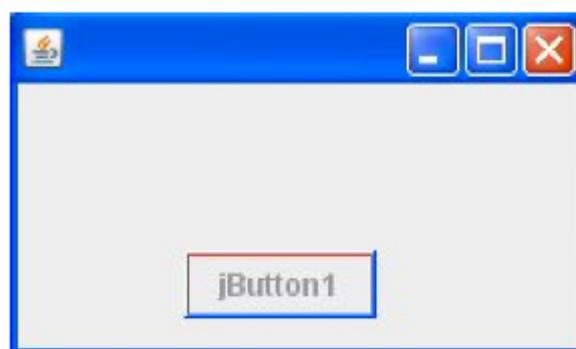
2. Verifique que a opção já está marcada. Por padrão, todo componente inserido já traz essa opção marcada, disponibilizando o componente na aplicação. Para desabilitá-lo, apenas clique no pequeno quadrado, desmarcando a opção (**Figura 20**).

Figura 20 - Propriedade enabled desativada.



3. Execute a aplicação. Passe o mouse sobre o componente e observe que o texto do botão mudou de cor. Essa tonalidade mais clara indica que o componente está desabilitado, mesmo clicando não se obterá nenhuma alteração ou resposta do componente (**Figura 21**):

Figura 21 - Botão desabilitado.



OBS.: A ação dessa propriedade pode ser também verificada em outros componentes, tais como nos componentes do tipo texto, por exemplo. Nesse caso, a opção de digitação não é permitida.



Vídeo 04 - Utilizando JTextField e JButton

Atividade 02

1. Para desabilitarmos um componente **JButton** em uma aplicação, isto é, não deixar que o usuário clique nele, devemos desmarcar o checkbox da propriedade _____ encontrada no painel **Propriedades**:
 - a. text
 - b. enabled
 - c. disabled
 - d. focusable
2. Um componente **JButton** nos permite adicionar um ícone juntamente com um rótulo, mas para isso devemos utilizar a propriedade _____ disponível no painel **Propriedades**:
 - a. imagem
 - b. icon
 - c. ícones
 - d. não é possível um ícone em um componente **JButton**
3. Crie uma aplicação em que o usuário possa informar seu nome, endereço, telefone e e-mail de contato e, posteriormente, enviar essas informações através de um botão. A cor de fundo do botão deve ser azul e os caracteres pretos.

Resumo

Nesta aula você aprendeu a utilizar alguns dos componentes mais básicos e úteis que são utilizados para a criação de aplicações gráficas utilizando a API Java Swing, como os campos de texto e os botões. Aprendeu também a utilizar algumas de suas propriedades a fim de melhorar o aspecto e o visual de seus projetos. Viu também que algumas propriedades são comuns para vários componentes e se comportam da mesma forma. Experimente explorar outras propriedades não discutidas aqui utilizando os mesmos procedimentos abordados na aula.

Autoavaliação

1. Cite duas situações em que poderemos utilizar os componentes **campo de texto (JTextField)** e o **botão (JButton)** na mesma aplicação.
2. Qual das propriedades abaixo não se aplica a um componente **JTextField**?
 - a. background
 - b. foreground
 - c. icon
 - d. text
3. Descreva três propriedades de cada componente que vimos nessa aula, especificando a funcionalidade de cada uma.

Referências

The JAVA Tutorials. **How to Use Text Fields**. Disponível em: <<http://download.oracle.com/javase/tutorial/uiswing/components/textfield.html>>. Acesso em: 5 abr. 2012.

_____. **How to Use Buttons, Check Boxes, and Radio Buttons**. Disponível em: <<http://download.oracle.com/javase/tutorial/uiswing/components/button.html>>. Acesso em: 5 abr. 2012.