

Desenvolvimento Desktop

Aula 09 - Componentes JProgressBar, JSlider e JSpinner

Apresentação

Nesta aula, aprenderemos mais sobre alguns componentes Swing. Conheceremos a **Barra de progresso (JProgressBar)** bastante utilizada para mostrar ao usuário o acompanhamento no “carregamento” de arquivos, imagens ou vídeos. O **controle deslizante (JSlider)** muito utilizado em situações em que é necessária uma faixa de valores gradual do tipo mínimo e máximo, como controle de temperatura e velocidade, por exemplo. O **controle giratório (JSpinner)**, muito semelhante ao **controle deslizante**, que também utiliza uma faixa de valores mínimo e máximo, mas só é visível o valor atual. Muito usado na seleção de meses, anos, faixas de idades, faixas de salários etc.



Vídeo 01 - Apresentação

Objetivos

Ao final desta aula, você será capaz de:

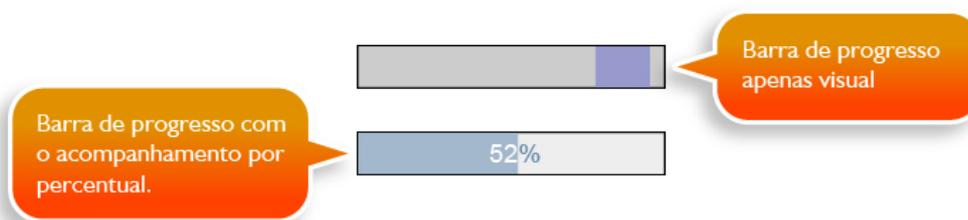
- Como utilizar o componente **barra de progresso** para mostrar o andamento da carga de um arquivo.
- Como o componente **controle deslizante** pode ser utilizado na prática.
- Onde podemos aplicar o **controle giratório** e como adequá-lo para sua aplicação.

O Componente Barra de Progresso (JProgressBar)

Navegando pela internet, com certeza você já se deparou com várias páginas de conteúdo “pesado”, seja com vídeos, figuras e/ou animações. Um bom exemplo são sites que disponibilizam a visualização de grandes imagens e de alta resolução. Dependendo do conteúdo ou até da especificação da máquina, o carregamento desse material visual pode demorar.

Uma forma de manter o usuário interessado e informado do quanto falta para ele visualizar a imagem é utilizando uma barra de progresso. Essa ferramenta possibilita ao usuário acompanhar, de forma visual (através de uma imagem) e/ou numérica (geralmente através de percentual) o progresso para a visualização do conteúdo. Veja dois exemplos na **Figura 1**:

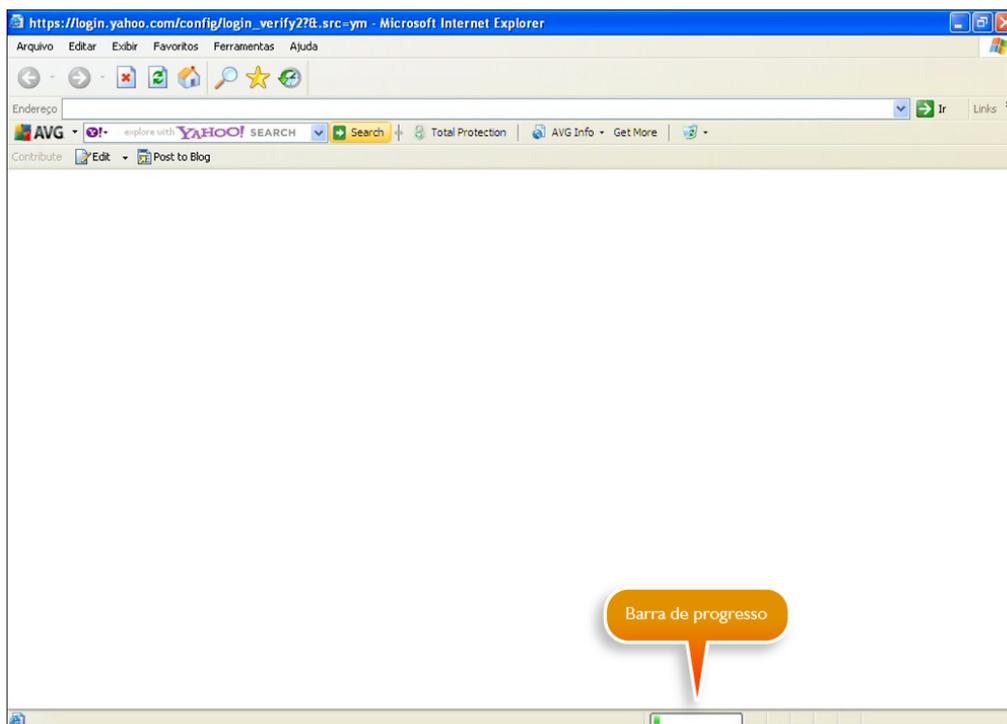
Figura 01 - Exemplos de barra de progresso



Fonte: www.download.oracle.com. Acesso em: 8 maio 2012

Outro exemplo bem fácil de verificar é observando o seu próprio navegador. Observe que na parte inferior (mais central no Internet Explorer e mais à direita no Mozilla Firefox), aparece uma barra de progresso que nos mostra como está o “carregamento” da página solicitada (**Figura 2**).

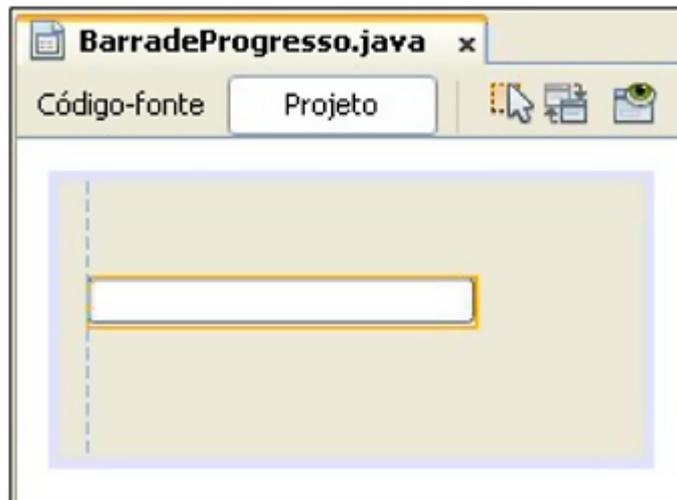
Figura 02 - Exemplo de barra de progresso no Internet Explorer



Vejamos passo a passo como utilizar essa ferramenta em nossas aplicações

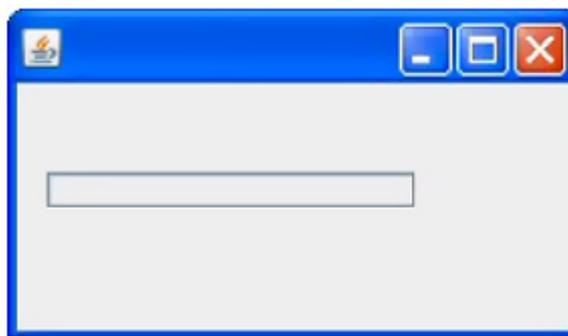
1. No NetBeans crie um novo projeto (**Arquivo > Novo projeto**).
2. Defina um nome e salve o projeto.
3. Clique com o botão direito no nome da aplicação e crie um novo **Formulário JFrame**.
4. Clique e arraste um componente **Barra de progresso** para a área do **JFrame**. Veja o exemplo da **Figura 3**:

Figura 03 - Componente barra de progresso



5. Execute a aplicação (**F6**) e confira com a **Figura 4**:

Figura 04 - Barra de progresso em execução



6. Você não verá nenhuma mudança nesse componente com as configurações padrão. Vejamos como alterar algumas propriedades:

As Propriedades do Componente Barra de Progresso

Lembre-se sempre das propriedades gerais que estamos vendo ao longo das aulas. Muitas são aplicadas a maioria dos componentes. Vejamos algumas particulares ao componente em questão.

- **maximum** – indica o valor máximo correspondente à "carga" da **barra de progresso**. O valor padrão é 100.
- **minimum** – representa o valor mínimo do componente, de forma similar a propriedade anterior. O valor padrão é 0.

- **orientation** – define o sentido da **barra de progresso**. O valor padrão é 0, indicando o sentido horizontal e 1 no sentido vertical.
- **value** – indica a partir de que valor o carregamento será iniciado. Esse valor deve ser um número entre o mínimo e o máximo definido nas propriedades correspondentes.

Na seção **Outras propriedades** você poderá alterar também as seguintes propriedades:

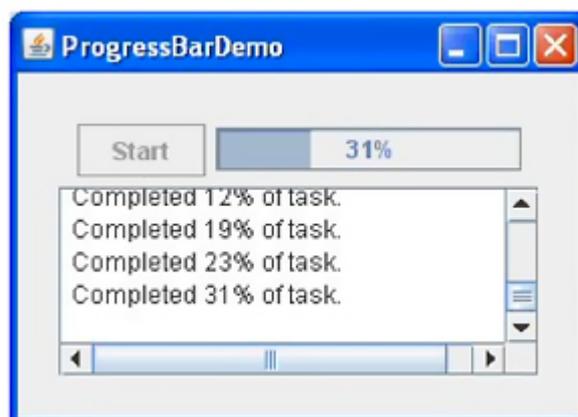
- **indeterminate** – se marcado, indica que o tamanho do arquivo a ser "carregado" é desconhecido. Com essa opção marcada e executando a aplicação, você perceberá que um retângulo azul ficará de um lado para o outro indefinidamente.
- **string** – indica o valor que aparecerá dentro do retângulo branco do componente. Geralmente, esse valor indica a porcentagem já carregada do arquivo que se aguarda.
- **stringPainted** – a marcação dessa opção exibe o valor especificado na propriedade string. Sem esse *checkbox* marcado, não aparecerá nada.

Para visualização de um exemplo prático, siga o link abaixo em seu navegador:

<<http://download.oracle.com/javase/tutorial/JSW/uiswing/components/ex6/ProgressBarDemo.jnlp>>

Abriu a aplicação? Clique no botão **Start** e acompanhe o progresso de carregamento, como mostra a **Figura 5**.

Figura 05 - Execução de aplicativo



Fonte: www.download.oracle.com. Acesso em: 8 abr. 2012

Entendeu melhor o funcionamento desse componente? Observe mais atentamente os programas que costuma utilizar e verá que esse é um componente bastante comum.



Vídeo 02 - JProgressBar

Atividade 01

1. Em que situações podemos utilizar o componente **barra de progresso**?
2. Qual a finalidade em manter o usuário informado do progresso de “carregamento” de um grande arquivo, por exemplo?
3. Qual a propriedade que permite a exibição de um valor dentro do componente **barra de progresso**?

O Componente Controle Deslizante (JSlider)

Esse componente é bastante utilizado quando se deseja permitir uma alteração gradual, como o controle de brilho e contraste em uma imagem ou uma situação na qual se pretende controlar a temperatura ou, ainda, o controle de velocidade de algum processo, por exemplo. Vejamos, então, como utilizá-lo em uma aplicação.

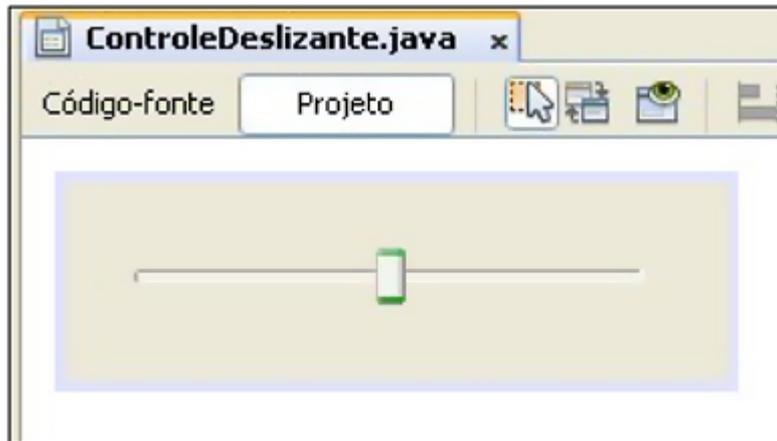
Como Utilizar o Componente Controle Deslizante

Vamos partir criando um novo projeto:

1. Execute o NetBeans e crie um novo projeto (**Arquivo > Novo projeto**).
2. Salve seu projeto com o nome: **ComponenteControleDeslizante**.
3. Crie um novo **Formulário JFrame**.

4. Clique e arraste um componente **Controle deslizante** para a área do projeto.
5. Visualmente você terá algo similar à **Figura 6**:

Figura 06 - Componente Controle deslizante durante o projeto



Continuando com o projeto, vejamos a seguir como alterar algumas propriedades.

As Propriedades do Componente Controle Deslizante

Você verá agora algumas propriedades que poderão ser alteradas para uso em uma aplicação:

- **maximum** – indica o valor máximo correspondente a barra total do componente. O valor padrão é 100.
- **minimum** – representa o valor mínimo do componente, de forma similar a propriedade anterior. O valor padrão é 0.
- **minorTickSpacing** – mostra o espaçamento (em números) entre o valor mínimo e o valor máximo do controle. Por exemplo, se você informar o valor 10, a faixa de valores será mostrada de dez em dez. Entretanto, esses valores só serão visíveis se a propriedade **paintLabels** estiver marcada. Cuidado para não informar um valor muito pequeno, senão eles ficarão muito próximos dificultando a leitura dos números. O valor padrão é 0.
- **orientation** – define o sentido do componente. O padrão é o sentido horizontal, contudo ele também pode ser utilizado na

vertical.

- **paintTrack** – exibe ou não a trilha do controle.
- **paintLabels** – mostra os valores definidos nas propriedades **majorTickSpacing** e **minorTickSpacing**.
- **paintTicks** – mostra os tracinhos verticais referentes aos valores definidos nas propriedades **majorTickSpacing** e **minorTickSpacing**.
- **snapToTicks** – se esta propriedade estiver marcada só será permitido selecionar os valores predeterminados da faixa, caso contrário, será permitido selecionar qualquer valor intermediário.
- **value** – indica o ponto de partida do controle deslizante quando a aplicação for executada. O valor padrão é 50. Isso significa que o marcador do controle será posicionado no centro do controle.

Na seção **Outras propriedades**, veja também a seguinte propriedade:

- **inverted** – essa propriedade inverte o sentido do controle. Caso o componente esteja no sentido horizontal (padrão), com essa propriedade marcada, o controle é feito a partir da direita, e não da esquerda (que é o padrão).

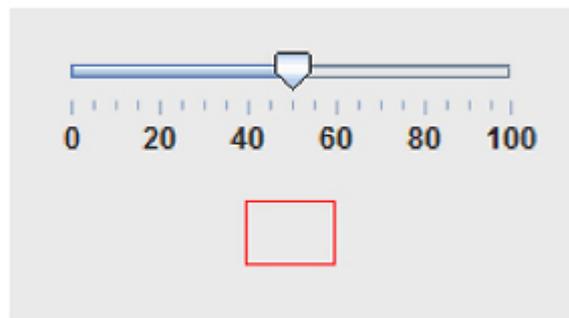
Vejamos agora um exemplo de como utilizar esse componente na prática.

Nesse exemplo, utilizaremos um componente **Controle deslizante** juntamente com um componente **Rótulo**, de forma que, quando o controle do componente for arrastado para qualquer um dos lados, o valor atual do controle seja mostrado no rótulo.

1. Execute o NetBeans e crie um novo projeto (**Arquivo à Novo projeto**).
2. Salve projeto com o nome: **ControleDeslizante**.
3. Crie um novo **Formulário JFrame**.
4. Clique e arraste um componente **Controle deslizante** para a área do projeto.
5. Para o intervalo de valores, mantenha os valores padrão das propriedades mínimo e máximo, ou seja, 0 e 100.

- Informe o valor 5 para a propriedade **minorTickSpacing** e 20 para a propriedade **majorTickSpacing**.
- Marque as propriedades **paintLabels** e **paintTicks** para mostrar os valores e os traços verticais.
- Marque a propriedade **snapToTicks** para que o controle deslizante fique vinculado apenas aos valores estabelecidos.
- Dê um nome para o controle. Por exemplo: **cdControle**.
- Arraste um componente rótulo e posicione-o logo abaixo do controle deslizante. Configure esse componente da forma que você achar mais conveniente (cor, fonte, tamanho etc.).
- Dê um nome para esse componente. Por exemplo: **lblRótulo**.
- Confira seu projeto com o mostrado na **Figura 7**:

Figura 07 - Componente controle deslizante com um componente rótulo



Agora que criamos o projeto, vamos ver como criar a parte funcional, ou seja, o evento apropriado para realizar a tarefa a qual foi destinado:

- Clique com o botão direito sobre o componente Controle deslizante.
- No menu apresentado, selecione: **Eventos > Change > stateChange**.
- Feito isso, será mostrado o código abaixo (**Figura 8**):

Figura 08 - Código referente ao evento do controle deslizante

```
private void cdControleStateChanged(javax.swing.event.ChangeEvent evt) {  
    // TODO add your handling code here:  
}
```

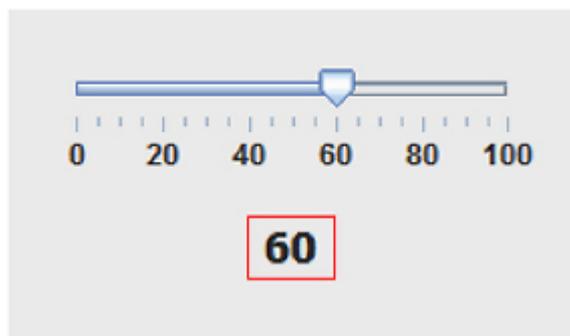
- Dentro desse código, insira as linhas conforme mostra a **Figura 9**:

Figura 09 - Código completado

```
private void cdControleStateChanged(javax.swing.event.ChangeEvent evt) {  
    // TODO add your handling code here:  
    int rotulo = cdControle.getValue();  
    String texto = Integer.toString(rotulo);  
    lblRotulo.setText(texto);  
}
```

5. Esse evento indica que quando o controle for arrastado, o código dentro das chaves será executado da seguinte forma:
 - a. Foi criada uma variável inteira chamada **rotulo** para receber o valor atual (**getValue()**) do componente **cdControle**.
 - b. Em seguida foi criada uma variável string chamada **texto** para converter a variável inteira **rotulo** para string de forma que o componente **lblRotulo** possa recebê-la.
 - c. Finalmente, o valor atual do controle é mostrado no **lblRotulo** como uma string.
6. Execute a aplicação e confira o resultado com a **Figura 10**:

Figura 10 - Execução de aplicativo



7. Deslize o controle para os lados e verifique se o valor mostrado no rótulo é o mesmo do controle.



Vídeo 03 - JSlider

Atividade 02

1. Que propriedade exibe ou não a trilha de um **controle deslizante**?
2. Se eu quiser que minha escala de valores comece do 0 quando a aplicação for executada, que propriedade deverei usar?
3. Cite duas situações que você poderia utilizar o componente **controle deslizante**.

O Componente Controle Giratório (JSpinner)

O componente **controle giratório** pode ser encontrado em aplicações que permita ao usuário controlar valores dentro de um intervalo, disponibilizando a facilidade de incremento e decremento a partir de pequenos botões com setas indicativas. Vejamos um exemplo prático.

Como Utilizá-lo em uma Aplicação

1. Execute o NetBeans e crie um projeto (**Arquivo > Novo projeto**).
2. Salve o projeto com o nome: **ControleGiratório**.
3. Clique com o botão direito na aplicação e crie um novo **Formulário JFrame**.
4. Clique e arraste um componente **controle giratório** para a área do projeto.
5. Você terá algo similar à **Figura 11**:

Figura 11 - Componente Controle giratório

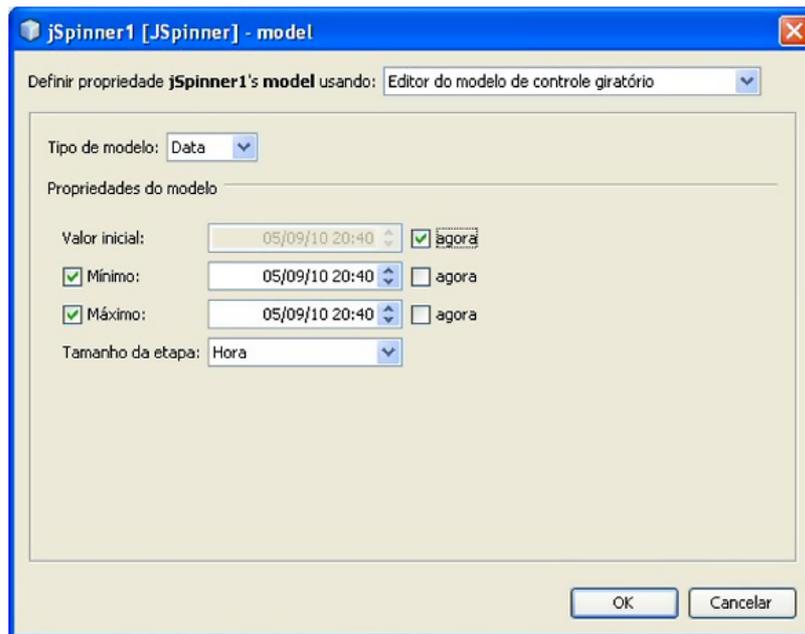


Vejamos a seguir como alterar algumas propriedades.

Propriedades do Componente Controle Giratório

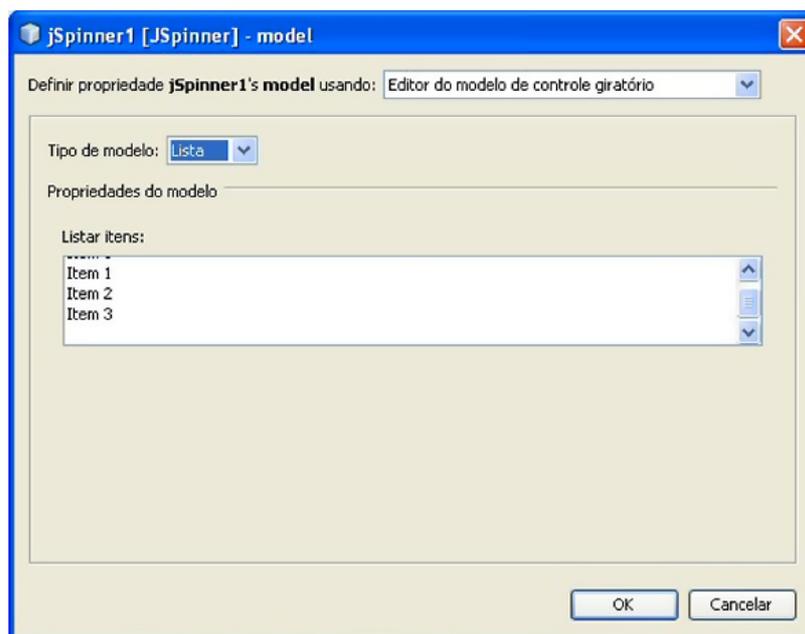
- **model** – essa é a propriedade mais importante de um **componente giratório**. Ela permite definir que tipo de informação o usuário poderá definir e configurar alguns parâmetros, conforme veremos a seguir:
- **DATA** – dispõe de diversos formatos de datas, incluindo a hora, como visto na **Figura 12**.

Figura 12 - Propriedade model mostrando os parâmetros do modelo data



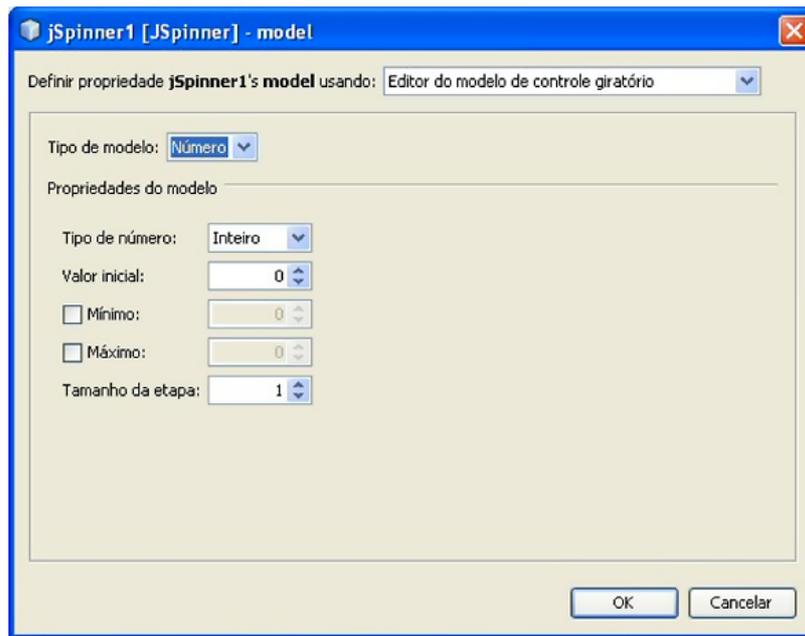
- **LISTA** – dispõe as informações em formato de lista, permitindo a edição de cada item. Veja a **Figura 13**.

Figura 13 - Propriedade model mostrando uma lista de opções



- **NÚMERO** – dispõe de diversos tipos e formatos de números, além do valor inicial, máximo, mínimo e o intervalo de acréscimo e decréscimo, conforme mostra a **Figura 14**.

Figura 14 - Propriedade model mostrando os parâmetros disponíveis para formatação de números



- **editor** – essa propriedade mostra o tipo de dado escolhido na janela da propriedade **model** (**Data**, **Lista** ou **Número**):
- **nextValue** – mostra o próximo valor da lista.
- **value** – mostra o valor atual da lista.

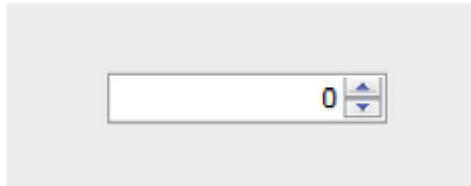
Vejamos o Uso desse Componente em uma Atividade Prática

Nesse exemplo, mostraremos como utilizar o componente giratório para mostrar os dias da semana, e quando uma opção for selecionada, o seu conteúdo será mostrado em um rótulo, ou seja, o dia da semana escolhido.

1. Execute o NetBeans e crie um projeto (**Arquivo > Novo projeto**).
2. Salve projeto com o nome: **ControleGiratório2**.
3. Clique com o botão direito na aplicação e crie um novo **Formulário JFrame**.
4. Clique e arraste um componente **Controle giratório** para a área do projeto.

5. Você terá algo similar à **Figura 15**:

Figura 15 - Aparência padrão de um controle giratório

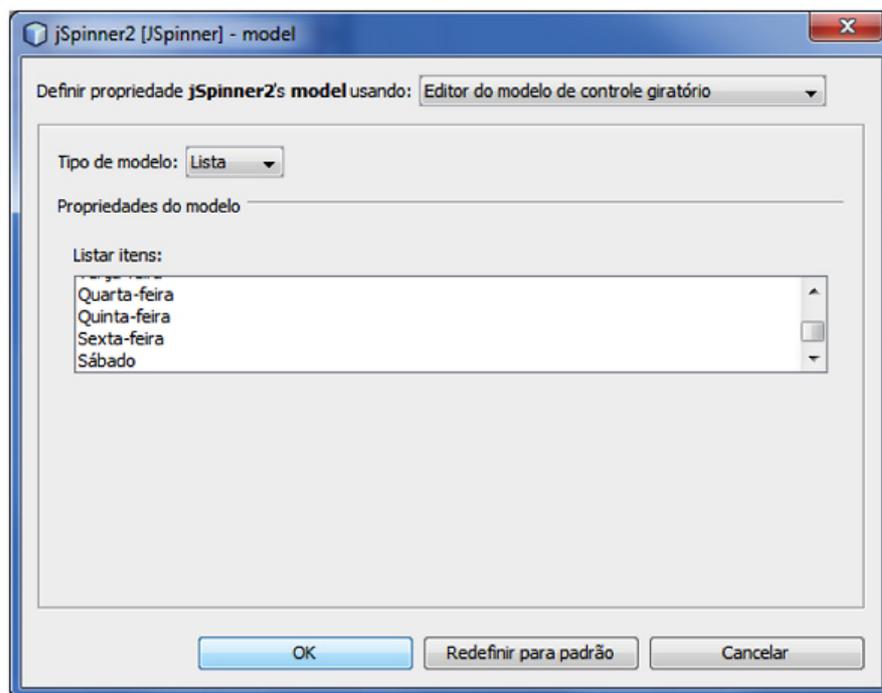


6. Clique na propriedade **model**. Na janela apresentada, altere:

- O tipo de modelo para: **Lista**.
- Apague a lista de itens atual e preencha com os dias da semana

7. Veja exemplo na **Figura 16**:

Figura 16 - Propriedade model com as informações preenchidas



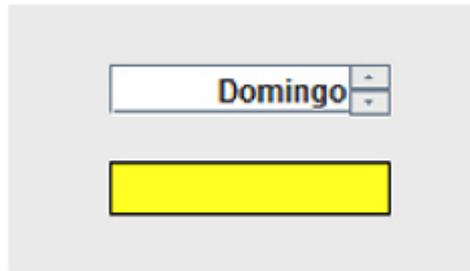
8. Clique em **OK** para confirmar.

9. No painel **Navegador**, altere o nome do **Controle giratório** para: **cgSemana**. Para isto, clique no nome da variável (JSpinner1) com o botão direito e selecione **Refatorar > Renomear**.

10. Acrescente um componente **Rótulo** logo abaixo do **Controle giratório**.

11. Altere o nome do rótulo para: **lblResultado**.
12. Apague o conteúdo da propriedade **text**, aplique uma cor de fundo qualquer e uma borda. Lembre-se de marcar a propriedade **opaque** para que a cor de fundo que você escolheu possa ser mostrada.
13. Feito isso, a aparência de seu projeto deverá estar parecida com a **Figura 17**:

Figura 17 - Aparência final do projeto



Agora aplicaremos um evento ao **componente giratório** de forma que quando um item da lista for selecionado ele seja mostrado no rótulo.

1. Clique com o botão direito no componente giratório e selecione a opção: **Eventos > Change > stateChanged** do menu apresentado.
2. No código criado, conforme mostra a **Figura 18**:

Figura 18 - Evento para o controle giratório

```
private void cgSemanaStateChanged(javax.swing.event.ChangeEvent evt) {  
    // TODO add your handling code here:  
}
```

3. Dentro desse código, insira as declarações contidas no exemplo da **Figura 19**:

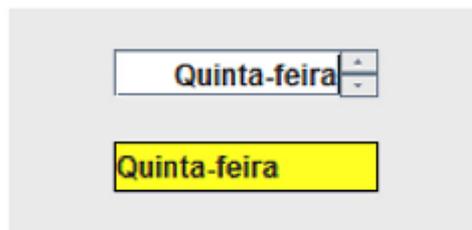
Figura 19 - Código utilizado para mostrar no rótulo o conteúdo atual do controle giratório

```
private void cgSemanaStateChanged(javax.swing.event.ChangeEvent evt) {  
    // TODO add your handling code here:  
    SpinnerModel diaSemana = cgSemana.getModel();  
    Object value = diaSemana.getValue();  
    lblResultado.setText((String) value);  
}
```

Vejam os Alguns Comentários sobre o Código

- Inicialmente, criamos uma variável **diaSemana** do tipo **SpinnerModel** para armazenar a lista do **componente giratório** (**cgSemana.getModel()**).
 - Em seguida, criamos um novo objeto **value** (**diaSemana.getValue()**) para armazenar o valor que for selecionado na lista.
 - Finalmente, mostramos no rótulo (**lblResultado.setText**) o valor selecionado na lista convertido para string.
4. Execute a aplicação e confira com a **Figura 20**:

Figura 20 - Aplicação após a execução



5. Utilize o mouse ou as teclas de setas “para cima” e “para baixo” para alterar o conteúdo do componente.



Vídeo 04 - JSpinner

Atividade 03

1. Que tipos de dados podemos utilizar em um **componente giratório**?
2. O que a propriedade **nextValue** nos mostra?
3. Desenvolva uma aplicação utilizando um componente **controle giratório** de tipo numérico entre -50 e 50, com um intervalo de 5 unidades entre eles.

Conclusão

Chegamos ao final de mais uma aula. Esperamos que você tenha aprendido mais e ampliado seus conhecimentos com os novos componentes. Sempre que criar uma aplicação, tente utilizar o máximo das propriedades dos componentes já estudados.

Resumo

Nesta aula, você foi apresentado a mais alguns componentes. Você aprendeu que o Progress Bar é muito útil quando se quer mostrar o processo de carga de um arquivo que está sendo transferido para seu computador. O controle deslizante mostra uma escala de valores que pode ser utilizada em diversas situações em uma aplicação, e o controle giratório, muito semelhante ao controle deslizante, muito utilizado na criação de listas em que somente uma opção é possível. Com mais esses componentes, você já poderá desenvolver aplicações bem mais aprimoradas. Na próxima aula, veremos como criar eventos para os componentes. Até lá.

Autoavaliação

1. Em que situação pode se ter uma **barra de progresso** indefinida?
2. É interessante utilizar a **barra de progresso** em arquivos "carregados" rapidamente?
3. Que propriedade permite definir o sentido dos componentes **controle deslizante** e **barra de progresso**?
4. Quais os dois valores possíveis que podemos utilizar na propriedade **orientation** de um componente **barra de progresso**?
 - a. 1 e 2
 - b. 0 e 1
 - c. HORIZONTAL e VERTICAL

Referências

THE JAVA tutorials. **How to Use Progress Bars.** Disponível em: <<http://download.oracle.com/javase/tutorial/uiswing/components/progress.html>>. Acessado em: 15 abr. 2012.

_____. **How to use sliders.** Disponível em: <<http://download.oracle.com/javase/tutorial/uiswing/components/slider.html>>. Acesso em: 15 abr. 2012.

_____. **How to use spinners.** Disponível em: <<http://download.oracle.com/javase/tutorial/uiswing/components/spinner.html>>. Acesso em: 15 abr. 2012.