

## TUTORIAIS FLASH

---

### O Componente ScrollPane

Copyright 2013 – Todos os Direitos Reservados  
Jorge Eider F. da Silva

Proibida a reprodução deste documento no todo ou em parte por quaisquer meios, seja digital, eletrônico ou impresso sem a expressa autorização do autor por escrito. Os infratores serão punidos de acordo com a Lei.

# TUTORIAIS FLASH

## O Componente ScrollPane

### Introdução

O componente **ScrollPane** é um recipiente para receber e mostrar movie clips, arquivos **JPEG**, **PNG**, **GIF** e **SWF** em uma área rolável, ou seja, conteúdos maiores do que a área estabelecida. Ao utilizar esse componente, você poderá limitar a área que deverá ser mostrada por esses tipos de mídia. Esse componente pode mostrar conteúdo proveniente de um disco local ou de um servidor. Esse conteúdo poderá ser configurado tanto durante o projeto quanto em tempo de execução usando o **ActionScript**.

Se você quiser especificar uma localização quando o conteúdo for carregado no **ScrollPane**, você deverá definir essa localização através das coordenadas X e Y, tais como 0,0.

### Interação do usuário com o componente ScrollPane

Um componente **ScrollPane** poderá estar habilitado ou desabilitado. Quando desabilitado, ele não receberá entrada nem do mouse nem do teclado. Quando habilitado ele receberá o foco tanto do mouse quanto do teclado, e portanto o usuário poderá utilizar as seguintes teclas para controlá-lo:

Tecla	Descrição
<b>Seta para baixo</b>	O conteúdo é rolado uma linha para cima.
<b>Seta para cima</b>	O conteúdo é rolado uma linha para baixo.
<b>End</b>	O conteúdo é rolado para o rodapé da área do componente.
<b>Seta para a esquerda</b>	O conteúdo é rolado uma linha para a direita.
<b>Seta para a direita</b>	O conteúdo é rolado uma linha para a esquerda.
<b>Home</b>	O conteúdo é rolado para o topo da área do componente.
<b>Page Down</b>	O conteúdo é rolado uma página para cima.
<b>Page Up</b>	O conteúdo é rolado uma página para baixo.

Você poderá usar o **ScrollPane** para mostrar qualquer conteúdo maior do que a área dimensionada do componente. Você poderá configurar o **ScrollPane** para permitir ao usuário arrastar o seu conteúdo para qualquer direção, definindo o parâmetro **scrollDrag** para **true**. Nesse caso, quando o ponteiro do mouse for posicionado dentro da área do componente, o cursor será alterado para uma pequena mão. Entre outros recursos, esse componente possui um evento para determinar se o conteúdo a ser carregado foi totalmente carregado. Dessa forma, você poderá criar funções específicas para serem executadas com garantia de que o conteúdo foi realmente carregado.

### Os parâmetros do componente ScrollPane

Para utilizar corretamente esse componente é necessário conhecer alguns dos seus parâmetros, que poderão ser definidos diretamente no painel **Component Inspector**, conforme mostramos a seguir:

Parâmetro	Descrição
<b>enabled</b>	Esse parâmetro é um valor booleano que indica se o componente deverá estar habilitado ou desabilitado quando a aplicação for executada. O valor padrão é <b>true</b> .
<b>horizontalLineScrollSize</b>	Indica o número de unidades que uma barra de rolagem horizontal será movida cada vez que o botão da seta é clicado. O valor padrão é <b>5</b> .

<b>horizontalPageScrollSize</b>	Indica o número de unidades que uma barra de rolagem horizontal será movida cada vez que a trilha da barra é clicada. O valor padrão é <b>20</b> .
<b>horizontalScrollPolicy</b>	Mostra ou não a barra de rolagem horizontal. Esse valor poderá ser <b>on</b> , <b>off</b> ou <b>auto</b> . O valor padrão é: <b>auto</b> .
<b>scrollDrag</b>	É um valor booleano que determina se é permitido ao usuário arrastar ( <b>true</b> ) ou não ( <b>false</b> ) o conteúdo dentro do <b>ScrollPane</b> . O valor padrão é <b>false</b> .
<b>source</b>	É um objeto que determina um URL absoluto ou relativo que identifica a localização de um arquivo SWF ou de uma imagem (JPEG, GIF ou PNG) a ser carregada, o nome da classe de um movie clip na biblioteca, uma referência a um objeto de exibição, ou o nome de uma instância de um movie clip no mesmo nível do componente. O valor padrão é: <b>null</b> .
<b>verticalLineScrollSize</b>	Indica o número de unidades que uma barra de rolagem vertical será movida cada vez que o botão da seta é clicado. O valor padrão é <b>5</b> .
<b>verticalPageScrollSize</b>	Indica o número de unidades que uma barra de rolagem vertical será movida cada vez que a trilha da barra é clicada. O valor padrão é <b>20</b> .
<b>verticalScrollPolicy</b>	Mostra ou não a barra de rolagem vertical. Esse valor poderá ser <b>on</b> , <b>off</b> ou <b>auto</b> . O valor padrão é <b>auto</b> .
<b>visible</b>	Esse parâmetro é um valor booleano que indica se o componente deverá aparecer visível ou oculto quando a aplicação for executada. O valor padrão é: <b>true</b> .

Cada um desses parâmetros possui uma propriedade equivalente em ActionScript com o mesmo nome, que podem ser utilizadas juntamente com outras propriedades, métodos e eventos disponíveis para esse componente permitindo-lhe criar aplicações mais interativas, conforme descrevemos abaixo:

### As propriedades do componente ScrollPane

São as seguintes as propriedades disponíveis para o componente **ScrollPane**:

Propriedade	Descrição
<b>bytesLoaded</b>	É um número que informa a quantidade de bytes do conteúdo já carregado. O valor padrão é: <b>0</b> . Propriedade somente de leitura.
<b>bytesTotal</b>	É um número que informa a quantidade total de bytes do conteúdo a ser carregado. O valor padrão é: <b>0</b> . Propriedade somente de leitura.
<b>content</b>	É uma referência ao conteúdo carregado dentro do componente <b>ScrollPane</b> . Propriedade somente de leitura. O valor padrão é: <b>null</b> .
<b>percentLoaded</b>	É um inteiro entre 0 e 100 que indica o percentual do conteúdo já carregado. Propriedade somente de leitura. O valor padrão é: <b>0</b> .

### Os métodos do componente ScrollPane

São os seguintes os métodos disponíveis para o componente **ScrollPane**:

Método	Descrição
<b>ScrollPane</b>	Cria uma nova instância do componente <b>ScrollPane</b> .
<b>getStyleDefinition</b>	Recupera o estilo de mapa padrão para o componente atual.
<b>load</b>	O parâmetro <b>request</b> desse método aceita somente um objeto <b>URLRequest</b> cuja propriedade <b>source</b> contém uma string, uma classe ou um objeto <b>URLRequest</b> .
<b>refreshPane</b>	Recarrega o conteúdo do <b>ScrollPane</b> .
<b>update</b>	Atualiza as propriedades da barra de rolagem baseado na largura e altura do conteúdo.

### Os eventos do componente ScrollPane

Eventos são ações utilizadas pelos componentes para executar tarefas específicas, de acordo com a finalidade de cada situação. Veja a seguir os eventos disponíveis para esse componente e para que servem.

Evento	Descrição
<b>complete</b>	É executado quando o conteúdo tiver sido carregado totalmente.
<b>progress</b>	É executado enquanto o conteúdo está sendo carregado.
<b>scroll</b>	É executado quando o usuário rola o conteúdo usando as barras de rolagem no componente ou a rodinha do mouse.

Vejamos como utilizar esses recursos nos exemplos mais adiante neste capítulo e no DVD, que acompanha a presente obra.

### **Criando aplicações com o componente ScrollPane**

Para entender melhor como funciona esse componente, mostraremos a seguir alguns exemplos práticos, inclusive em conjunto com outros componentes.

#### **Exemplo 1**

Nesse exemplo utilizaremos um componente **ScrollPane** e uma imagem qualquer, de forma que quando a aplicação for executada, a imagem seja carregada automaticamente no **ScrollPane**. Para isso, selecione uma imagem qualquer (de preferência grande), no formato JPG, e salve-a no mesmo diretório onde o arquivo FLA será salvo. Vejamos como fazer isso:

1. Crie um novo documento (ActionScript 3.0);
2. Abra o painel **Components (Ctrl + F7)**, e arraste um componente **ScrollPane** para o palco;
3. No painel **Properties**, dimensione-o para o tamanho: **300 x 200** pixels;
4. No painel **Component Inspector**, no parâmetro **source**, digite o nome da imagem com a extensão. No nosso caso: **carol1.jpg**;
5. Defina o parâmetro **scrollDrag** para **true**, para que o usuário possa arrastar a imagem em qualquer direção;
6. Nomeie a camada atual para: **componentes**;
7. Execute a aplicação e confira o resultado com o mostrado na **Figura 13.1** a seguir.



**Figura 13.1 – Componente ScrollPane com imagem carregada automaticamente.**

\*\*\*\*\*

#### **Exemplo 2**

Nesse exemplo utilizaremos um componente **ScrollPane**, dois botões e duas imagens quaisquer (de preferência grandes e formato **JPG**), de forma que quando cada botão for

clocado a imagem correspondente deverá ser carregada. Essas imagens deverão estar no mesmo diretório do arquivo **FLA**. Vejamos como fazer isso:

1. Crie um novo documento (ActionScript 3.0);
2. Abra o painel **Components (Ctrl + F7)**, e arraste um componente **ScrollPane** para o palco;
3. No painel **Properties**, dimensione-o para o tamanho: **300 x 200** pixels, e digite um nome para sua instância. Por exemplo: **fotos**;
4. No painel **Component Inspector**, defina o parâmetro **scrollDrag** para **true**, para que o usuário possa arrastar a imagem em qualquer direção;
5. Crie ou importe da biblioteca dois botões e arraste-os para o palco. Crie os rótulos: **Foto1** e **Foto 2**, e as respectivas instâncias para eles: **bt1** e **bt2**;
6. Nomeie a camada atual para: **componentes**;
7. Crie uma nova camada e chame-a de: **ações**;
8. No primeiro frame dessa camada insira o seguinte código:

```
stop();
flash.events.MouseEvent;
bt1.addEventListener(MouseEvent.CLICK, carregaFoto1);
function carregaFoto1(event:Event):void {
    fotos.source = "diego1.jpg";
}
bt2.addEventListener(MouseEvent.CLICK, carregaFoto2);
function carregaFoto2(event:Event):void {
    fotos.source = "diego2.jpg";
}
```

#### Vejamos os comentários sobre o código:

**OBS.:** Em todos os exemplos do presente capítulo incluiremos o comando **stop()** na primeira linha do código. Esse comando tem como objetivo interromper a aplicação quando a mesma for executada, mesmo que a aplicação não possua mais de um frame na sua linha do tempo. Consideramos um bom hábito essa prática. Portanto, não comentaremos mais essa linha nos próximos exemplos.

#### Na linha:

```
flash.events.MouseEvent;
```

Importamos a classe necessária para podermos utilizar o evento do mouse adequado para o nosso exemplo.

#### No bloco de código a seguir:

```
bt1.addEventListener(MouseEvent.CLICK, carregaFoto1);
function carregaFoto1(event:Event):void {
    fotos.source = "diego1.jpg";
}
```

Utilizamos o evento **CLICK** vinculado à instância do primeiro botão (**bt1**), de forma que quando o mesmo for clicado, a função **carregaFoto1** seja executada. Em seguida criamos a função **carregaFoto1** que por sua vez carregará no componente **ScrollPane** (instância: **fotos**) a imagem correspondente. Para isso utilizamos a propriedade **source**.

#### No bloco seguinte:

```
bt2.addEventListener(MouseEvent.CLICK, carregaFoto2);
function carregaFoto2(event:Event):void {
    fotos.source = "diego2.jpg";
}
```

Utilizamos o mesmo evento **CLICK** utilizado no botão 1. Quanto à função, alteramos apenas o nome e a imagem a ser carregada.

9. Execute a aplicação e confira o resultado com o mostrado na **Figura 13.2** abaixo:



Figura 13.2 – Componente ScrollPane sendo carregado com duas imagens.

\*\*\*\*\*

### Exemplo 3

Nesse exemplo criaremos inicialmente uma galeria de fotos pré-selecionadas como um movie clip, e em seguida utilizaremos um componente **ScrollPane** para inserirmos essa galeria, de forma que cada foto será mostrada totalmente quando uma das setas da barra de rolagem for clicada, para a esquerda ou para a direita. Vejamos como fazer isso:

#### Etapa 1: Criação da galeria das fotos pequenas

Para criar essa galeria, você precisará de:

Selecionar uma série de imagens (no nosso caso utilizamos sete fotos) no tamanho **90 x 60** pixels cada uma, e em seguida agrupá-las em um único objeto. Para isso:

1. Crie um novo documento (ActionScript 3.0);
2. Importe para o palco as fotos selecionadas e alinhe-as conforme mostra a **Figura 13.3** abaixo:

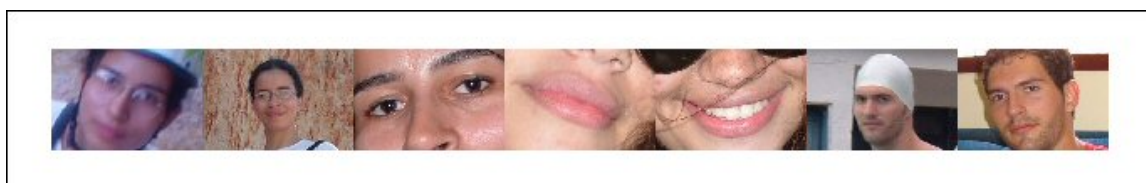


Figura 13.3 – Imagens alinhadas para a criação de uma galeria.

3. Nomeie a camada atual para: **galeria**;
4. Agrupe todas as fotos (**Ctrl + G**). O tamanho total das fotos deverá medir exatamente **630** pixels;
5. Salve o arquivo com o nome de: **galeria**, e compile (**Shift + F12**).

#### Etapa 2: Criação do documento principal

1. Crie um novo documento (ActionScript 3.0);
2. No painel **Properties**, redimensione o palco para **200 x 200**;

3. Nomeie a camada atual para: **componentes**;
4. Crie um texto estático e digite o título da aplicação: **Galeria de Fotos**;
5. Arraste um **ScrollPane** para o palco e centralize-o. Dê um nome para sua instância: **fotos**;
6. Na aba **Parameters** do painel **Component Inspector**, altere os seguintes parâmetros:

**horizontalLineScrollSize: 90.**

**source: galeria.swf**

7. O primeiro parâmetro significa que a cada clicada em uma das setas no **ScrollPane**, será passada uma foto de cada vez, tanto para a esquerda quanto para a direita, tendo em vista que o valor desse parâmetro (**90**) é exatamente a largura de cada imagem;
8. O segundo parâmetro carrega o conjunto de imagens (arquivo: **galeria.swf**) para o **ScrollPane**;
9. Execute a aplicação e confira o resultado com o mostrado na **Figura 13.5** a seguir:



Figura 13.4 – Componente ScrollPane utilizado para a exibição de uma galeria de imagens.

10. Observe que, como o tamanho do arquivo **galeria.swf** é maior do que a área do **ScrollPane**, é criada automaticamente uma barra de rolagem horizontal;

\*\*\*\*\*

#### Exemplo 4

Nesse exemplo utilizaremos um componente **ScrollPane** para carregar cinco imagens, e um componente **NumericStepper**, de forma que à medida que o valor do **NumericStepper** for alterado, será carregada a imagem correspondente no **ScrollPane**. O **NumericStepper** será numerado de 1 a 5 e será utilizada uma matriz para armazenar as imagens. Vejamos como fazer isso:

Inicialmente, selecione 5 imagens quaisquer com as dimensões: **90 x 60** pixels, no formato **.jpg** e armazena-as no mesmo diretório da aplicação. Feito isso:

1. Crie um novo documento (ActionScript 3.0);
2. Importe as cinco imagens pré-selecionadas para a biblioteca do usuário;
3. Acesse a biblioteca (**Ctrl + L**), clique com o botão direito na primeira foto e selecione a opção **Linkage**. Na caixa de diálogo **Linkage Properties**, marque a opção **Export for ActionScript**. Na caixa de texto **Class**, digite: **foto1**;
4. Repita o mesmo passo para o restante das fotos;
5. Nomeie a camada atual para: **componentes**;
6. Dimensione o palco para: **250 x 150** pixels;
7. Crie um texto estático e digite o título: **Galeria de Fotos**;
8. Arraste um componente **ScrollPane** para o palco. Dimensione-o para **90 x 60** pixels e dê um nome para sua instância: **fotos**;
9. Arraste um componente **NumericStepper** para o palco. Altere sua largura para **50** pixels. Dê um nome para sua instância: **contador**.

10. No painel **Component Inspector**, altere o parâmetro **maximum** para **5** e **minimum** para **1**;
11. Crie uma nova camada e chame-a de: **ações**;
12. No primeiro frame dessa camada insira o seguinte código:

```
stop();  
//  
fotos.source="foto1";  
var imagens:Array=new Array();  
imagens=["foto1","foto2","foto3","foto4","foto5"];  
//  
contador.addEventListener(Event.CHANGE, carregaFoto);  
//  
function carregaFoto(event:Event):void {  
    fotos.source = imagens[contador.value-1];  
}
```

### Vejamos os comentários sobre o código:

Na linha:

```
fotos.source="foto1";
```

Iniciamos com o carregamento da primeira imagem no componente **ScrollPane** através de sua propriedade **source**, para evitar que o componente seja iniciado vazio.

Na linha:

```
var imagens:Array=new Array();
```

Criamos uma nova matriz através de seu respectivo construtor e armazenamos na variável **imagens**.

Na linha seguinte:

```
imagens=["foto1","foto2","foto3","foto4","foto5"];
```

Armazenamos na matriz **imagens** as fotos que utilizaremos na nossa galeria.

Na linha a seguir:

```
contador.addEventListener(Event.CHANGE, carregaFoto);
```

Utilizamos o evento **CHANGE** vinculado ao componente **NumericStepper** através de sua instância **contador**, de forma que quando o valor do mesmo for alterado, a função **carregaFoto** seja executada.

No bloco de código:

```
function carregaFoto(event:Event):void {  
    fotos.source = imagens[contador.value-1];  
}
```

Criamos a função **carregaFoto** que será executada quando o valor do **NumericStepper** for alterado, de forma que carregue no **ScrollPane** através da propriedade **source** a imagem correspondente ao seu valor na matriz **imagens** subtraído de uma unidade, tendo em vista que toda matriz começa com o índice 0 (zero) e o valor inicial do **NumericStepper** é 1.

13. Execute a aplicação e confira o resultado com o mostrado na **Figura 13.5** a seguir:





Figura 13.5 – Utilização de um componente ScrollPane juntamente com um componente NumericStepper para exibir uma galeria de fotos.

\*\*\*\*\*

## Exemplo 5

Nesse exemplo utilizaremos um componente **ScrollPane** em tempo de execução para carregar uma imagem de um servidor (qualquer um de sua preferência), de forma que à medida que a barra de rolagem vertical for movida, a altura do componente seja alterada em intervalos de 5 pixels. Da mesma forma, à medida que a barra de rolagem horizontal for movida, a largura do componente seja também alterada em intervalos de 5 pixels. Vejamos como fazer isso:

1. Crie um novo documento (ActionScript 3.0);
2. Abra o painel **Components (Ctrl + F7)**, e arraste um componente **ScrollPane** para a biblioteca;
3. Nomeie a camada atual para: **ações**;
4. No primeiro frame dessa camada insira o seguinte código:

```
stop();
//
import fl.containers.ScrollPane;
import fl.events.ScrollEvent;
import fl.controls.ScrollBarDirection;
//
var minhaURL="http://www.fifacupsoftheworld.com/imagens/ferrari2.jpg";
var imagem:ScrollPane = new ScrollPane();
imagem.scrollDrag=true;
imagem.move(30,30);
imagem.setSize(250, 200);
imagem.source = minhaURL;
addChild(imagem);
//
imagem.addEventListener(ScrollEvent.SCROLL, alteraValor);
//
function alteraValor(event:ScrollEvent):void {
    switch (event.direction) {
        case ScrollBarDirection.HORIZONTAL :
            imagem.width+=5;
            break;
        case ScrollBarDirection.VERTICAL :
            imagem.height+=5;
            break;
    }
}
```

### Vejamos os comentários do código acima:

No bloco de código abaixo:

```
import fl.containers.ScrollPane;
import fl.events.ScrollEvent;
```

```
import fl.controls.ScrollBarDirection;
```

Importamos as respectivas classes do componente **ScrollPane** para que possamos utilizar os recursos adequadamente na nossa aplicação. A primeira classe nos permite criar instâncias do componente, a segunda classe no permite utilizar os seus eventos e a terceira classe nos permite definir a direção da barra de rolagem.

Na linha:

```
var minhaURL="http://www.fifacupsoftheworld.com/imagens/ferrari2.jpg";
```

Utilizamos a variável **minhaURL** para armazenar o endereço absoluto da imagem que deverá ser carregada no componente **ScrollPane**. Informe um endereço válido que contenha uma imagem qualquer.

No bloco de código seguinte:

```
var imagem:ScrollPane = new ScrollPane();
imagem.scrollDrag=true;
imagem.move(30,30);
imagem.setSize(250, 200);
imagem.source = minhaURL;
addChild(imagem);
```

Criamos uma nova instância (**imagem**) do componente **ScrollPane** através de seu construtor e alteramos as seguintes propriedades:

- **scrollDrag** – configuramos essa propriedade para **true** para que a imagem possa ser arrastada para qualquer direção com o mouse.
- **move** – essa propriedade posiciona o componente no palco através das coordenadas **x** e **y** respectivamente.
- **setSize** – essa propriedade define o tamanho do componente (largura e altura).
- **source** - Essa propriedade define a imagem a ser carregada no componente quando a aplicação for executada.

Em seguida, inserimos o componente no palco através do método **addChild()**.

Na linha:

```
imagem.addEventListener(ScrollEvent.SCROLL, alteraValores);
```

Utilizamos o evento **SCROLL** vinculado à instância **imagem** do componente, de forma que quando for realizada uma rolagem, a função **alteraValor** seja executada.

Finalmente, no bloco de código a seguir:

```
function alteraValores(event:ScrollEvent):void {
    switch (event.direction) {
        case ScrollBarDirection.HORIZONTAL :
            imagem.width+=5;
            break;
        case ScrollBarDirection.VERTICAL :
            imagem.height+=5;
            break;
    }
}
```

Criamos a função chamada **alteraValores** de forma que quando a mesma for executada, ocorra o seguinte:

Se a barra que for rolada for a vertical, a altura do componente será acrescida de 5 pixels, e conseqüentemente, a barra de rolagem desaparecerá. Caso contrário, se a

barra que for rolada for a horizontal, a largura do componente será acrescida também de 5 pixels, e a barra de rolagem também desaparecerá.

5. Execute a aplicação e confira o resultado com o mostrado na **Figura 13.8** abaixo:



**Figura 13.6 – Componente ScrollPane criado em tempo de execução cujas dimensões serão alteradas quando as barras de rolagens forem utilizadas.**

\*\*\*\*\*

### Exercícios de Fixação

1. O parâmetro **scrollDrag** é utilizado para permitir que o usuário arraste ou não o conteúdo do componente **ScrollPane**:
  - a) Verdadeiro
  - b) Falso
2. Uma nova instância chamada **sPane** de um componente **ScrollPane** pode ser criada em tempo de execução da seguinte forma:
  - a) `var sPane: ScrollPane = new ScrollPane(sPane);`
  - b) `var sPane.ScrollPane = new ScrollPane();`
  - c) `var sPane: ScrollPane = new ScrollPane();`
  - d) `var sPane: ScrollPane = new ScrollPane();`
  - e) `var sPane: ScrollPane = new ScrollPane();`
3. O valor padrão para os parâmetros **horizontalScrollPolicy** e **verticalScrollPolicy** de um componente **ScrollPane** é:
  - a) on
  - b) true
  - c) auto
  - d) false
  - e) off
4. O valor padrão para o número de unidades dos parâmetros **horizontalLineScrollSize** e **verticalLineScrollSize** de um componente **ScrollPane** é:
  - a) 1
  - b) 2
  - c) 3
  - d) 4
  - e) 5

5. Para informarmos o conteúdo a ser carregado em um componente **ScrollPane** utilizamos o parâmetro **source**:
- Verdadeiro
  - Falso
6. Para criarmos um componente **ScrollPane** em tempo de execução é necessário que o mesmo esteja presente no palco:
- Verdadeiro
  - Falso
7. Para sabermos qual a quantidade total de bytes do conteúdo de um componente **ScrollPane** utilizamos o seguinte método:
- totalBytes
  - getTotalBytes
  - getBytesTotal
  - getBytesTotally
  - bytesTotal
8. Quando um componente **ScrollPane** está com o foco, para movermos o seu conteúdo para o topo via teclado utilizamos a(s) tecla(s):
- End
  - Seta para cima
  - Seta para baixo
  - Ctrl + Home
  - Home
9. Qual o valor padrão da propriedade **percentLoaded** de um componente **ScrollPane**?
- 0
  - 1
  - 2
  - 3
  - 4
10. Que evento é executado quando o usuário rola o conteúdo de um componente **ScrollPane**?
- move
  - change
  - scroll
  - rollOn
  - mouseOn

\*\*\*\*\*

### Exercícios Propostos

- Crie uma aplicação utilizando um componente **ScrollPane** juntamente com um componente **NumericStepper**, de forma que:
  - Quando a aplicação for executada carregue uma imagem automaticamente.
  - Quando o valor do componente **NumericStepper** for alterado, o valor **alpha** da imagem no componente **ScrollPane** deverá ser alterado com o percentual correspondente a esse valor.

O intervalo de valores do **NumericStepper** deverá ser entre 10 e 100 com intervalos de 10 pontos.

Veja uma sugestão de layout para esse exemplo:



2. Crie uma aplicação com seis componentes **ScrollPane** em tempo de execução, de forma que:
- Quando a aplicação for executado os componentes **ScrollPane** sejam carregados automaticamente com seis imagens diferentes, provenientes de um servidor (qualquer um de sua preferência) ou local.
  - As imagens deverão ser arrastadas com o mouse.

Veja uma sugestão de layout para esse exemplo:



\*\*\*\*\*