

## TUTORIAIS FLASH

---

### O Componente UI ScrollBar

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.

## O Componente UI ScrollBar

---

### Introdução

Esse componente permite-lhe adicionar uma barra de rolagem a um campo de texto dinâmico. Você poderá adicionar essa barra durante o projeto ou em tempo de execução utilizando a linguagem ActionScript. Para usar esse componente você precisará criar primeiro um campo de texto no palco e em seguida arrastar o **UI ScrollBar** a partir do painel **Components** para qualquer margem do campo de texto.

Essa barra de rolagem funciona como qualquer outra, dispondo de duas pequenas setas nas extremidades para que o texto possa ser rolado para cima, para baixo, para a esquerda ou para a direita (dependendo da posição da barra), quando forem clicadas. Ela poderá ser anexada a qualquer uma das margens de um campo de texto, como também poderão ser utilizadas tanto horizontal quanto verticalmente.

Se o tamanho da barra de rolagem for menor do que o tamanho combinado de suas setas de rolagem, ela não será mostrada corretamente. Um dos botões de setas ficará escondido atrás do outro. O Flash não prevê esse tipo de erro. Nesses casos, é mais interessante ocultar a barra de rolagem utilizando o ActionScript. Caso o conteúdo inserido não ultrapasse a caixa de texto, o Flash ocultará os controles da barra de rolagem.

### Interação do usuário com o componente UI ScrollBar

Diferentemente da maioria dos componentes, o componente **UI ScrollBar** poderá receber a entrada do mouse continuamente. Isso acontece sempre que o usuário mantém o botão do mouse pressionado em uma das setas da barra de rolagem, como se fossem cliques repetidos. Por outro lado, não existe nenhuma interação com esse componente através do teclado.

### Os parâmetros do UI ScrollBar

Você poderá alterar os parâmetros desse componente em tempo de projeto, utilizando para isso o painel **Properties** para cada instância do **UI ScrollBar**, conforme descritos abaixo:

Parâmetro	Descrição
<b>direction</b>	É uma string que indica se a barra de rolagem será horizontal ou vertical. O valor padrão é: <b>vertical</b> .
<b>scrollTargetName</b>	Vincula uma instância de um componente <b>TextField</b> a uma instância de um componente <b>UI ScrollBar</b> .
<b>visible</b>	É um valor booleano que indica se o componente deverá ou não ficar visível.

Cada um desses parâmetros possui uma propriedade equivalente no ActionScript com o mesmo nome, que você poderá utilizá-las juntamente com os métodos e eventos disponíveis, e escrever códigos mais complexos para as suas aplicações em tempo de execução, conforme mostraremos abaixo:

### As propriedades do componente UI ScrollBar

Além dos parâmetros já descritos acima, são as seguintes as propriedades apresentadas no painel **Properties**.

Propriedade	Descrição
-------------	-----------

<b>Instance Name</b>	Utilizada para nomear a instância de uma <b>UI ScrollBar</b> .
<b>Position and Size</b>	Utilizada para estabelecer a posição (x e y) do componente no palco, e as suas dimensões (largura e altura).
<b>Color Effect</b>	Utilizada para estabelecer um efeito de cores e transparência (alpha) sobre o componente.
<b>Display</b>	Utilizada para aplicar efeitos visuais no componente.

### Os métodos do componente UI ScrollBar

Além dos parâmetros especificados acima, você poderá também utilizar os seguintes métodos específicos do componente **UI ScrollBar**:

<b>Método</b>	<b>Descrição</b>
<b>UI ScrollBar</b>	Cria uma nova instância do componente <b>UI ScrollBar</b> .
<b>getStyleDefinition</b>	Recupera o estilo de mapa padrão para o componente atual.
<b>setScrollProperties</b>	Configura a faixa e o tamanho do viewport de um componente <b>UI ScrollBar</b> . Esse método possui os seguintes parâmetros: <b>pageSize</b> , <b>minScrollPosition</b> , <b>maxScrollPosition</b> e <b>pageScrollSize</b> .
<b>update</b>	Força a barra de rolagem a atualizar suas propriedades imediatamente, após as mudanças.

### Os eventos do componente UI ScrollBar

Embora o **UI ScrollBar** não disponha de eventos específicos para execução, você poderá utilizar os eventos disponíveis para o mouse com esse componente, tais como: **Click**, **rollOver**, **rollOut**, entre outros, além de eventos de outras classes disponíveis no pacote **fl.controls**.

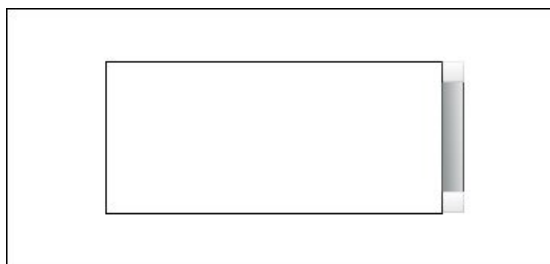
### Criando aplicações com o componente UI ScrollBar

Mostraremos a seguir alguns exemplos práticos de como utilizar alguns recursos do **UI ScrollBar**, inclusive em conjunto com outros componentes.

#### **Exemplo 1**

Esse exemplo anexará um componente **UI ScrollBar** a um campo de texto dinâmico criado no palco durante o projeto, cujo conteúdo será inserido durante a execução da aplicação. Vejamos como fazer isso:

1. Crie um novo documento (ActionScript 3.0);
2. Crie um campo de texto dinâmico no palco e dimensione-o para qualquer tamanho;
3. Abra o painel **Components (Ctrl + F7)** e arraste um componente **UI ScrollBar**, e posicione-o do lado direito do campo de texto dinâmico. Ele deverá se encaixar automaticamente no campo de texto. Veja na **Figura 19.1** como deverá ficar o layout depois disso:



**Figura 19.1 – Aparência original do componente UI ScrollBar quando vinculado a um campo de texto dinâmico.**

4. Nomeie a camada atual para: **componentes**;

5. No **Inspetor de Propriedades**, crie uma instância para o campo de texto dinâmico. Por exemplo: **texto**. Para alinhar o texto para ambas as margens clique no ícone **Justify**;
6. No parâmetro **scrollTargetName** do componente **UI ScrollBar**, digite: **texto**. Isso faz com que a barra de rolagem seja vinculada ao campo de texto.
7. Crie uma nova camada e chame-a de: **ações**;
8. No primeiro frame dessa camada insira o seguinte código:

```
stop();  
texto.wordWrap=true;  
texto.text="A besta ou balestra é uma arma com a aparência de uma espingarda, com um arco de flechas, acoplado na ponta de sua coronha, acionada por gatilho, que projeta setas, dardos similares a flechas. Ela foi bastante usada no século XVI e chegou a coexistir com e depois foi substituída pelos mosquetes, primeiras armas de fogo. Hoje, continua a ser fabricada, pois é usada, em algumas partes do mundo, por caçadores. A palavra besta teria sido sincopada do italiano balestra, que por sua vez deriva do latim tardio ballistra.";
```

### 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:

```
texto.wordWrap=true;
```

Definimos a propriedade **wordWrap** do campo de texto dinâmico para **true** para que cada linha do texto passe para a linha seguinte quando atingir a sua margem direita.

#### Na linha a seguir:

```
texto.text="A besta ou balestra é uma arma com a aparência de uma espingarda, com um arco de flechas, acoplado na ponta de sua coronha, acionada por gatilho, que projeta setas, dardos similares a flechas. Ela foi bastante usada no século XVI e chegou a coexistir com e depois foi substituída pelos mosquetes, primeiras armas de fogo. Hoje, continua a ser fabricada, pois é usada, em algumas partes do mundo, por caçadores. A palavra besta teria sido sincopada do italiano balestra, que por sua vez deriva do latim tardio ballistra.";
```

Definimos o texto a ser inserido no campo de texto dinâmico através da propriedade **text**.

9. Execute a aplicação e confira o resultado com o apresentado na **Figura 19.2** a seguir:

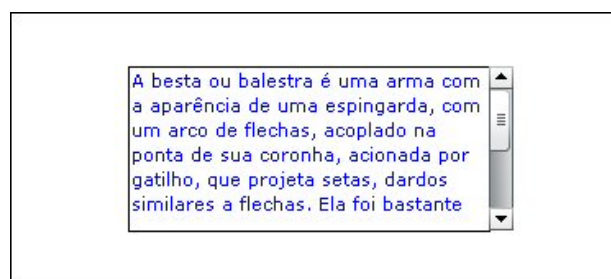


Figura 19.2 – Utilização do componente **UI ScrollBar** vinculado a um campo de texto dinâmico preenchido em tempo de execução.

\*\*\*\*\*

## Exemplo 2

Nesse exemplo utilizaremos um componente **UI ScrollBar** horizontal vinculado a um campo de texto dinâmico preenchido automaticamente em tempo de execução. Além disso, utilizaremos algumas tags em HTML, incluindo um link, alteraremos também a cor de fundo e mostraremos o número de caracteres que o texto contém. Veja a seguir como fazer isso:

1. Crie um novo documento (ActionScript 3.0);
2. Crie um campo de texto dinâmico com as seguintes dimensões: **300 x 25** pixels. Crie a seguinte instância para ele: **meuTexto**. E na opção **Behavior**, selecione **Multiline no wrap**, para que o texto seja mostrado em apenas uma única linha;
3. Abra o painel **Components (Ctrl + F7)**, arraste um componente **UI ScrollBar** e posicione-o no interior do campo de texto. Ele deverá se ajustar ao campo de texto automaticamente. Observe que a propriedade **scrollTargetName** do painel **Component Inspector** será alterada automaticamente para **meuTexto**;
4. Nomeie a camada atual para: **componentes**;
5. Crie uma nova camada e chame-a de: **ações**;
6. No primeiro frame dessa camada insira o seguinte código:

```
stop();
meuTexto.background=true;
meuTexto.backgroundColor=0xffff00;
meuTexto.htmlText="Já havia <u>registros</u> desta arma na Roma antes e depois
de cristo como arma de caça ou de guerra. Informações: <a
href='http://www.google.com'>www.google.com</a>";
trace(meuTexto.length);
```

### Vejamos os comentários sobre o código:

Nas linhas:

```
meuTexto.background=true;
meuTexto.backgroundColor=0xffff00;
```

Configuramos a propriedade **background** para **true** vinculada ao campo de texto **meuTexto** para que possamos utilizar uma cor para o fundo do campo de texto. Em seguida, utilizamos a propriedade **backgroundColor** para definirmos a cor de fundo do campo.

Na linha:

```
meuTexto.htmlText="Já havia <u>registros</u> desta arma na Roma antes e depois
de cristo como arma de caça ou de guerra. Informações: <a href =
'http://www.google.com'>www.google.com</a>";
```

Utilizamos a propriedade **htmlText** para que possamos utilizar as tags do HTML em nosso texto, e ao mesmo tempo definimos o texto a ser inserido no campo de texto dinâmico. As tags utilizadas foram: **<u>** e **</u>** para sublinhar uma parte do texto, e no final um link para uma página na Internet.

Na linha seguinte:

```
trace(meuTexto.length);
```

Utilizamos a propriedade **length** para mostrar no painel **Output** a quantidade de caracteres que o texto contém.

7. Execute a aplicação e confira o resultado com o mostrado na **Figura 19.3** a seguir:



Figura 19.3 – Utilização do componente `UI ScrollBar` com barra horizontal utilizando recursos do HTML e de estilo.

Arraste a barra de rolagem até o final do texto e experimente clicar no link para abrir a página indicada.

\*\*\*\*\*

### Exemplo 3

Nesse exemplo utilizaremos um componente `UI ScrollBar` horizontal vinculado a um campo de texto em tempo de execução. O campo de texto será preenchido automaticamente também em tempo de execução. O texto será carregado a partir de um servidor remoto. Veja a seguir como fazer isso:

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

```
stop();
import flash.net.URLLoader;
import fl.controls.UI ScrollBar;
import flash.events.Event;
//
var meuTexto:TextField = new TextField();
meuTexto.border = true;
meuTexto.width = 200;
meuTexto.height = 16;
meuTexto.x = 200;
meuTexto.y = 150;
//
var scrollBar:UI ScrollBar = new UI ScrollBar();
scrollBar.direction = "horizontal";
scrollBar.setSize(meuTexto.width, meuTexto.height);
scrollBar.move(meuTexto.x, meuTexto.height + meuTexto.y);
//
// Posiciona os componentes no palco.
addChild(meuTexto);
addChild(scrollBar);
// Carrega texto.
var carregaTexto:URLLoader = new URLLoader();
var request:URLRequest = new URLRequest
("http://www.fifacupsoftheworld.com/texto.txt");
carregaTexto.load(request);
carregaTexto.addEventListener(Event.COMPLETE, cargaCompleta);
//
function cargaCompleta(event:Event) {
    // Insere o texto carregado no campo de texto.
```

```

    meuTexto.text = carregaTexto.data;
    // Define meuTexto como destino para a barra de rolagem.
    scrollBar.scrollTarget = meuTexto;
}

```

### Vejam os comentários sobre o código:

Nas linhas:

```

import flash.net.URLLoader;
import fl.controls.UI ScrollBar;
import flash.events.Event;

```

Importamos as classes dos componentes e eventos necessários para que possamos utilizá-los em nossa aplicação adequadamente.

No bloco de código:

```

var meuTexto:TextField = new TextField();
meuTexto.border = true;
meuTexto.width = 200;
meuTexto.height = 16;
meuTexto.x = 200;
meuTexto.y = 150;

```

Criamos uma instância (**meuTexto**) da classe **TextField** utilizando o seu devido construtor para que possamos utilizá-lo juntamente com o componente **UI ScrollBar**. Em seguida definimos:

- A sua borda, configurando-a para **true**.
- A sua largura.
- A sua altura.
- E a sua posição no palco através das coordenadas **X** e **Y**.

Nas linhas seguintes:

```

var scrollBar:UI ScrollBar = new UI ScrollBar();
scrollBar.direction = "horizontal";
scrollBar.setSize(meuTexto.width, meuTexto.height);
scrollBar.move(meuTexto.x, meuTexto.height + meuTexto.y);

```

Criamos uma instância (**scrollBar**) do componente **UI ScrollBar** utilizando o seu respectivo construtor, e em seguida definimos:

- A sua direção.
- O seu tamanho, com as mesmas medidas do campo de texto criado, para que ambos fiquem vinculados.
- E a sua posição no palco através das mesmas coordenadas **X** e **Y** do campo de texto.

Nas linhas abaixo:

```

addChild(meuTexto);
addChild(scrollBar);

```

Inserimos as instâncias no palco utilizando o método **addChild()**.

Nas linhas:

```

var carregaTexto:URLLoader = new URLLoader();
var request:URLRequest = new URLRequest(
    "http://www.fifacupsoftheworld.com/texto.txt");
carregaTexto.load(request);

```

Criamos uma instância (**carregaTexto**) da classe **URLLoader** utilizando o seu respectivo construtor. Criamos também uma nova instância (**request**) da classe **URLRequest** e definimos a origem do texto a ser carregado através do endereço de um servidor remoto. Em seguida, vinculamos esse endereço à instância **carregaTexto** através do método **load()**, para carregar o texto.

Na linha abaixo:

```
carregaTexto.addEventListener(Event.COMPLETE, cargaCompleta);
```

Utilizamos o evento **COMPLETE** vinculado à instância **carregaTexto**, de forma que quando o texto for totalmente carregado, a função **cargaCompleta** seja executada.

E finalmente, nas linhas a seguir:

```
function cargaCompleta(event:Event) {  
    // Insere o texto carregado no campo de texto.  
    meuTexto.text = carregaTexto.data;  
    // Define meuTexto como destino para a barra de rolagem.  
    scrollbar.scrollTarget = meuTexto;  
}
```

Criamos a função **cargaCompleta**, de forma que quando a mesma for executada:

- Carregue o texto no campo de texto **meuTexto**, definido na instância **carregaTexto**.
- Vincule a barra de rolagem ao campo de texto **meuTexto**.

5. Execute a aplicação e confira o resultado com o mostrado na **Figura 19.4** a seguir:

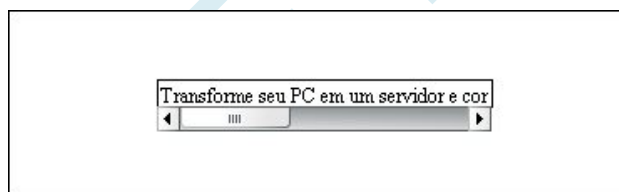


Figura 19.4 – Componente **UI ScrollBar** criado juntamente com um campo de texto dinâmico em tempo de execução.

\*\*\*\*\*

#### Exemplo 4

Nesse exemplo utilizaremos dois componentes **UI ScrollBar**, um na horizontal e outro na vertical vinculado a um campo de texto em tempo de execução. O campo de texto será preenchido automaticamente também em tempo de execução a partir de um servidor remoto. Veja a seguir como fazer isso:

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

```
stop();  
import fl.controls.UI ScrollBar;  
import fl.controls.ScrollBarDirection;  
//  
var url:String = "http://www.fifacupsoftheworld.com/texto2.txt";  
//
```



```

var carregador:URLLoader = new URLLoader(new URLRequest(uri));
carregador.addEventListener(Event.COMPLETE, cargaCompleta);
//
var campoTexto:TextField = new TextField();
campoTexto.width = 320;
campoTexto.height = 100;
campoTexto.x = 50;
campoTexto.y = 50;
campoTexto.border = true;
campoTexto.multiline = true;
campoTexto.wordWrap = false;
addChild(campoTexto);
//
var hScrollBar:UI ScrollBar = new UI ScrollBar();
hScrollBar.direction = ScrollBarDirection.HORIZONTAL;
hScrollBar.move(campoTexto.x, campoTexto.y + campoTexto.height);
hScrollBar.width = campoTexto.width;
hScrollBar.scrollTarget = campoTexto;
addChild(hScrollBar);
//
var vScrollBar:UI ScrollBar = new UI ScrollBar();
vScrollBar.direction = ScrollBarDirection.VERTICAL;
vScrollBar.move(campoTexto.x + campoTexto.width, campoTexto.y);
vScrollBar.height = campoTexto.height;
vScrollBar.scrollTarget = campoTexto;
addChild(vScrollBar);
//
function cargaCompleta(event:Event):void {
    campoTexto.text = event.target.data;
    hScrollBar.update();
    vScrollBar.update();
}

```

### Vejam os comentários sobre o código:

Nas linhas:

```

import fl.controls.UI ScrollBar;
import fl.controls.ScrollBarDirection;

```

Importamos as classes necessárias do respectivo componente para que possamos utilizá-lo adequadamente.

Na linha:

```

var url:String = "http://www.fifacupsoftheworld.com/texto2.txt";

```

Criamos um variável string (**url**) para armazenar o endereço absoluto do arquivo que queremos importar do servidor (**texto2.txt**).

Nas linhas seguintes:

```

var carregador:URLLoader = new URLLoader(new URLRequest(uri));
carregador.addEventListener(Event.COMPLETE, cargaCompleta);

```

Criamos uma instância (**carregador**) da classe **URLLoader** cujo parâmetro é o endereço do arquivo texto que queremos importar. Em seguida, utilizamos o evento **COMPLETE** vinculado à instância **carregador**, de forma que quando o texto for totalmente carregado, a função **cargaCompleta** seja executada.

Nas linhas a seguir:

```

var campoTexto:TextField = new TextField();
campoTexto.width = 320;

```

```
campoTexto.height = 100;
campoTexto.x = 50;
campoTexto.y = 50;
campoTexto.border = true;
campoTexto.multiline = true;
campoTexto.wordWrap = false;
addChild(campoTexto);
```

Criamos uma instância da classe **TextField** chamada **campoTexto**, e em seguida definimos os seguintes parâmetros para ela:

- A largura.
- A altura.
- As coordenadas **X** e **Y**, ou seja, a posição no palco.
- A borda.
- A possibilidade de aceitar múltiplas linhas (**multiline=true**).
- A possibilidade de não quebrar as linhas quando atingir a margem do campo de texto (**wordWrap=false**).

Em seguida, utilizamos o método **addChild()** para inserir a instância no palco.

No bloco de código a seguir:

```
var hScrollBar:UI ScrollBar = new UI ScrollBar();
hScrollBar.direction = ScrollBarDirection.HORIZONTAL;
hScrollBar.move(campoTexto.x, campoTexto.y + campoTexto.height);
hScrollBar.width = campoTexto.width;
hScrollBar.scrollTarget = campoTexto;
addChild(hScrollBar);
```

Criamos uma instância do componente **UI ScrollBar** chamada **hScrollBar**. Depois estabelecemos os seguintes parâmetros para ela:

- A posição da barra de rolagem (horizontal).
- A posição no palco, que nesse caso é igual à posição **X** do campo de texto e de sua posição **Y** adicionado de sua altura.
- A largura, que nesse caso é igual à largura do campo de texto.
- O objeto a ser vinculado, que no caso, também é o mesmo campo de texto.

Em seguida, utilizamos o método **addChild()** para inserir a instância no palco.

No bloco de código:

```
var vScrollBar:UI ScrollBar = new UI ScrollBar();
vScrollBar.direction = ScrollBarDirection.VERTICAL;
vScrollBar.move(campoTexto.x + campoTexto.width, campoTexto.y);
vScrollBar.height = campoTexto.height;
vScrollBar.scrollTarget = campoTexto;
addChild(vScrollBar);
```

Criamos uma instância do componente **UI ScrollBar** chamada **vScrollBar**. Depois estabelecemos os seguintes parâmetros para ela:

- A posição da barra de rolagem (vertical).
- A posição no palco, que nesse caso é igual à posição **X** do campo de texto adicionado de sua largura, e sua posição **Y**.
- A altura, que nesse caso é igual à altura do campo de texto.
- O objeto a ser vinculado, que no caso, também é o mesmo campo de texto.

Em seguida, utilizamos o método **addChild()** para inserir a instância no palco.

Na linha seguinte:

```

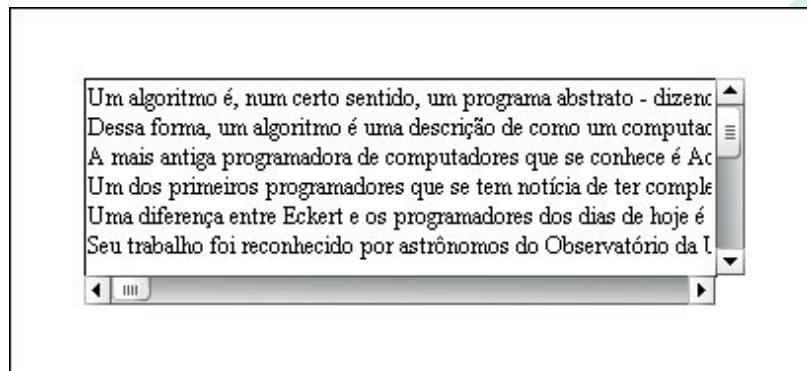
function cargaCompleta(event:Event):void {
    campoTexto.text = event.target.data;
    hScrollBar.update();
    vScrollBar.update();
}

```

Criamos uma função chamada **cargaCompleta**, de forma que quando a mesma for executada:

- Preencha o campo de texto com o texto importado do servidor.
- Atualize ambas as barras de rolagem.

5. Execute a aplicação e confira o resultado com o mostrado na **Figura 19.5** a seguir:



**Figura 19.5 – Utilização de dois componentes UI ScrollBar simultaneamente com arquivo de texto importado de um servidor remoto.**

\*\*\*\*\*

## Exemplo 5

Nesse exemplo utilizaremos um componente **UI ScrollBar** vertical vinculado a um campo de texto dinâmico preenchido automaticamente em tempo de execução, cujo conteúdo é proveniente de um arquivo texto local. Além disso, utilizamos também algumas tags do HTML no próprio texto e alteraremos a cor de fundo do texto através de suas propriedades. Veja a seguir como fazer isso:

1. Crie um novo documento (ActionScript 3.0);
2. Crie um campo de texto dinâmico com as seguintes dimensões: **300 x 25** pixels. Crie a seguinte instância para ele: **meuTexto**. E na opção **Line type**, selecione **Multiline no wrap**, para que o texto seja mostrado em apenas uma única linha;
3. Abra o painel **Components (Ctrl + F7)**, arraste um componente **UI ScrollBar** e posicione-o no interior do campo de texto. Ele deverá se ajustar ao campo de texto automaticamente. Observe que a propriedade **scrollTargetName** será alterada automaticamente para **meuTexto**;
4. Nomeie a camada atual para: **componentes**;
5. Crie uma nova camada e chame-a de: **ações**;
6. No primeiro frame dessa camada insira o seguinte código:

```

stop();
import fl.controls.UI ScrollBar;
import fl.controls.ScrollBarDirection;
//
var url:String = "texto3.txt";
//
var carregador:URLLoader = new URLLoader(new URLRequest(url));
carregador.addEventListener(Event.COMPLETE, cargaCompleta);

```

```
//
var campoTexto:TextField = new TextField();
campoTexto.width = 320;
campoTexto.height = 100;
campoTexto.x = 50;
campoTexto.y = 50;
campoTexto.border = true;
campoTexto.multiline = true;
campoTexto.wordWrap = true;
campoTexto.background = true;
campoTexto.backgroundColor = 0x0000ff;
addChild(campoTexto);
//
var vScrollBar:UI ScrollBar = new UI ScrollBar();
vScrollBar.direction = ScrollBarDirection.VERTICAL;
vScrollBar.move(campoTexto.x + campoTexto.width, campoTexto.y);
vScrollBar.height = campoTexto.height;
vScrollBar.scrollTarget = campoTexto;
addChild(vScrollBar);
//
function cargaCompleta(event:Event):void {
    campoTexto.htmlText = event.target.data;
}
}
```

### Vejamos os comentários sobre o código:

#### Nas linhas:

```
import fl.controls.UI ScrollBar;
import fl.controls.ScrollBarDirection;
```

Importamos as classes necessárias do respectivo componente para que possamos utilizá-lo adequadamente.

#### Na linha:

```
var url:String = "texto3.txt";
```

Criamos uma variável (**url**) do tipo **String** para armazenar o texto a ser carregado.

#### Na linha seguinte:

```
var carregador:URLLoader = new URLLoader(new URLRequest(url));
```

Criamos uma instância (**carregador**) da classe **URLLoader** cujo parâmetro é um novo construtor da classe **URLRequest** que por sua vez armazena a variável (**url**) do texto que será carregado.

#### Na linha:

```
var campoTexto:TextField = new TextField();
```

Criamos uma instância da classe **TextField** chamada **campoTexto** utilizando o seu respectivo construtor.

#### Nas linhas seguintes:

```
campoTexto.width = 320;
campoTexto.height = 100;
campoTexto.x = 50;
campoTexto.y = 50;
campoTexto.border = true;
campoTexto.multiline = true;
campoTexto.wordWrap = true;
```

```
campoTexto.background = true;
campoTexto.backgroundColor = 0x0000ff;
addChild(campoTexto);
```

Uma vez criado o novo campo de texto, definimos os seguintes parâmetros para ele:

- A largura.
- A altura
- A posição do componente no palco através das suas coordenadas **X** e **Y**.
- A borda.
- A possibilidade de aceitar múltiplas linhas.
- A possibilidade de quebrar as linhas quando as mesmas atingirem a margem direita do campo de texto.
- A possibilidade de aceitar uma cor de fundo.
- A cor de fundo (azul).

Em seguida, utilizamos o método **addChild()** para inserirmos a instância no palco.

No bloco de código:

```
var vScrollBar:UI ScrollBar = new UI ScrollBar();
vScrollBar.direction = ScrollBarDirection.VERTICAL;
vScrollBar.move(campoTexto.x + campoTexto.width, campoTexto.y);
vScrollBar.height = campoTexto.height;
vScrollBar.scrollTarget = campoTexto;
addChild(vScrollBar);
```

Criamos uma instância do componente **UI ScrollBar** chamada **vScrollBar**. Depois estabelecemos os seguintes parâmetros para ela:

- A posição da barra de rolagem (vertical).
- A posição no palco, que nesse caso é igual à posição **X** do campo de texto adicionado de sua largura, e sua posição **Y**.
- A altura, que nesse caso é igual à altura do campo de texto.
- O objeto a ser vinculado, que no caso, também é o mesmo campo de texto.

Em seguida, utilizamos o método **addChild()** para inserir a instância no palco.

Nas linhas seguintes:

```
carregador.addEventListener(Event.COMPLETE, cargaCompleta);
function cargaCompleta(event:Event):void {
    campoTexto.htmlText = event.target.data;
}
```

Utilizamos o evento **COMPLETE** vinculado à instância **carregador**, de forma que quando o texto for totalmente carregado, a função **cargaCompleta** seja executada. Em seguida, criamos a função chamada **cargaCompleta**, que deverá preencher o campo de texto com o texto definido na string **url**, quando a mesma for executada.

7. Execute a aplicação e confira o resultado com o mostrado na **Figura 19.6** a seguir:

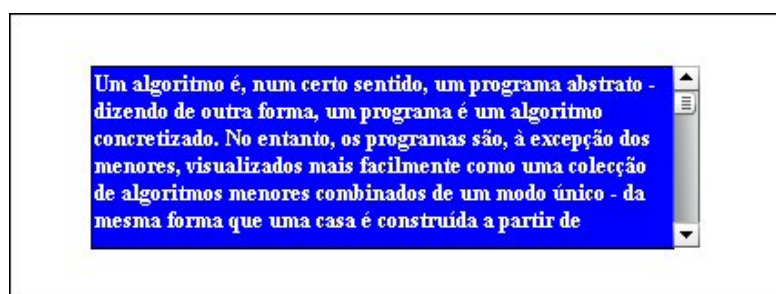


Figura 19.6 – Utilização do componente **UI ScrollBar** com barra vertical utilizando recursos do HTML.

\*\*\*\*\*

### Exercícios de Fixação

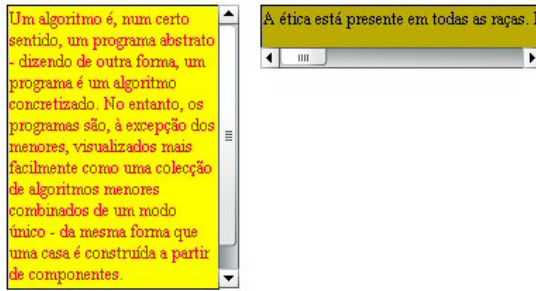
1. Em que direção(ões) é permitido utilizar um componente **UI ScrollBar** em um campo de texto?
  - a) Somente na horizontal
  - b) Somente na vertical
  - c) Em ambas as direções simultaneamente
  - d) Somente na horizontal se o campo de texto for estático.
  - e) Somente na vertical se o campo de texto for estático.
2. Que método devemos utilizar para criarmos uma nova instância de um componente **UI ScrollBar**?
  - a) verticalScrollBar
  - b) horizontalUI ScrollBar
  - c) newUI ScrollBar
  - d) UI ScrollBar
3. Que tipo de texto podemos utilizar juntamente com um componente **UI ScrollBar**?
  - a) estático
  - b) dinâmico
  - c) label
  - d) textArea
4. Qual a alternativa correta para se criar uma nova instância (de nome: **barra**) de um componente **UI ScrollBar**?
  - a) var barra: UI ScrollBar = new UI ScrollBar();
  - b) var barra: UI ScrollBar = new UI ScrollBar();
  - c) var barra: UI ScrollBar = new UI ScrollBar();
  - d) var barra: UI ScrollBar = new UI ScrollBar();
  - e) var barra: UI ScrollBar = new UI ScrollBar();
5. Que classe precisamos importar para criarmos uma instância de um componente **UI ScrollBar** em tempo de execução?
  - a) fl.control.UI ScrollBar;
  - b) fl.controls.UI ScrollBar;
  - c) fl.controlsUI ScrollBar;
  - d) flash.UI ScrollBar;
6. O que acontecerá a um componente **UI ScrollBar** se o campo de texto vinculado a ele for alterado o seu tamanho em tempo de execução?
  - a) O componente ficará com o mesmo tamanho do campo de texto.
  - b) O componente permanecerá do mesmo tamanho.
  - c) Depende da posição do componente.
  - d) O componente ficará desabilitado.

7. Como será mostrado um componente **UI ScrollBar** cujo conteúdo do campo de texto a ele vinculado não preencha totalmente a área estabelecida?
- a) Aparecerá mas não serão mostradas as setas da barra de rolagem.
  - b) O componente não será mostrado.
  - c) O componente aparecerá totalmente, incluindo as setas, mas desabilitado.
  - d) Aparecerá mas não será mostrado o controle deslizante da barra de rolagem.
8. Se a propriedade **wordWrap** de um campo de texto dinâmico vinculado a um componente **UI ScrollBar** for configurada para **false**, e a direção da barra de rolagem tiver sido originalmente configurada para **vertical**, o que acontecerá quando a aplicação for executada?
- a) A barra de rolagem será automaticamente alterada para horizontal e ficará abaixo do campo de texto.
  - b) A barra de rolagem permanecerá na mesma posição.
  - c) A barra de rolagem será automaticamente alterada para horizontal mas ficará fora do campo de texto.
  - d) A propriedade do campo de texto **wordWrap** será automaticamente configurada para **true**.
9. Que propriedade devemos utilizar para vincularmos uma instância de um componente **UI ScrollBar** a um campo de texto dinâmico?
- a) `getStyleDefinition`
  - b) `scrollTargetName`
  - c) `setScrollProperties`
  - d) `setScrollName`
10. Qual o valor padrão da propriedade **direction** de um componente **UI ScrollBar**?
- a) Horizontal
  - b) Vertical

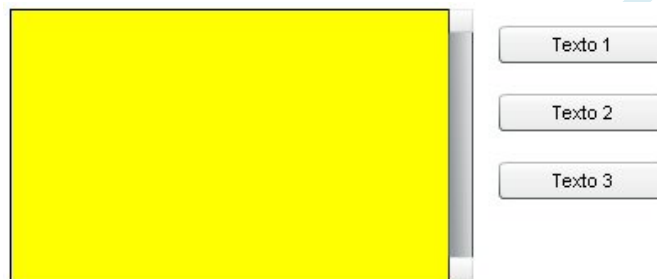
\*\*\*\*\*

## Exercícios Propostos

1. Desenvolva uma aplicação utilizando um componente **UI ScrollBar** vertical e um campo de texto dinâmico **TextField** em tempo de execução, com fundo amarelo e caracteres vermelhos. O texto deverá ser carregado a partir de um servidor remoto (qualquer um de sua preferência).
2. Desenvolva uma aplicação utilizando dois componentes **UI ScrollBar** sendo um horizontal e outro vertical vinculados a dois campos de textos dinâmicos **TextField** em tempo de execução, de forma que sejam carregados dois textos diferentes de um diretório local, nos respectivos **TextFields** quando a aplicação for executada. Veja na figura abaixo uma sugestão de layout para esse exercício.



3. Desenvolva uma aplicação utilizando um componente **UI ScrollBar** vertical vinculado a um campo de texto dinâmico **TextField** e três componentes **Button** em tempo de execução, de forma que cada um dos botões carregue um texto diferente no **TextField**. Os textos podem ser carregados tanto a partir de um servidor remoto (qualquer um de sua preferência) quanto de um diretório local, ou ambos. Veja na figura abaixo uma sugestão de layout para esse exercício.



\*\*\*\*\*