

TUTORIAIS FLASH

O Componente List

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 List

Introdução

Esse componente é encontrado com muita frequência em quase todos os Web sites, geralmente como parte de formulários, para preenchimento de cadastros, compra de produtos, entre outras finalidades. Ele é utilizado para a criação de listas dos mais variados assuntos, tais como a lista dos meses do ano, lista de produtos, lista de profissões, entre outras. Nesses casos, o usuário seleciona um ou mais itens da lista, alternados ou não, dependendo do caso, e a partir daí ocorre uma ação em resposta dessa seleção.

Na lista de itens do **List** podem ser incluídos gráficos, e até mesmo outros componentes. Cada item da lista possui um valor correspondente, chamado de **index** (ou índice), começando com **zero** (para o primeiro item da lista). Quando um item é adicionado ou excluído da lista, utilizando-se as propriedades e métodos da classe **List**, o índice de cada elemento da lista é automaticamente alterado.

Interação do usuário com o componente List

Você poderá configurar uma lista para que os usuários possam fazer seleções simples ou múltiplas. Podemos citar como exemplo uma lista de produtos a serem adquiridos pelo usuário, onde ele deverá escolher um ou mais itens de uma lista.

Um componente **List** recebe foco quando você clica nele ou usa a tecla **TAB**. Quando ele está com o foco, você poderá utilizar as seguintes teclas para controlá-lo:

Tecla	Finalidade
Teclas alfanuméricas	Seleciona o item da lista cujo primeiro caractere foi pressionado.
Control	Juntamente com o mouse permite selecionar ou deselecionar vários itens de uma lista, alternados ou não.
Seta para baixo	A seleção é movida um item para baixo.
Home	A seleção é movida para o topo da lista.
Page Down	A seleção é movida uma página para baixo.
Page Up	A seleção é movida uma página para cima.
Shift	Permite selecionar itens contínuos.
Seta para cima	A seleção é movida um item para cima.

Para utilizar corretamente esse componente, ele dispõe de alguns parâmetros que devem ser preenchidos adequadamente, conforme mostrados abaixo.

Os parâmetros do componente List

Para uma melhor utilização do componente **List**, você poderá configurar os seguintes parâmetros disponíveis no **Inspetor de Propriedades** ou no painel **Component Inspector** para cada instância desse componente.

Parâmetro	Descrição
allowMultipleSelection	Nesse parâmetro você deverá definir um valor booleano, do tipo true ou false , possibilitando-lhe selecionar vários itens (true) ou apenas um item (false) da lista. O valor padrão é false .
dataProvider	Nesse parâmetro você deverá criar uma matriz de valores para serem agregados aos dados da lista. O valor padrão é uma matriz vazia ("").
enabled	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 true .
horizontalLineScrollSize	Indica o número de unidades que uma barra de rolagem horizontal é movida cada vez que o botão da seta é clicado. O valor padrão é 4 .
horizontalPageScrollSize	Indica o número de unidades que uma barra de rolagem horizontal é movida cada vez que a trilha da barra é clicada. O valor padrão é 0 .
horizontalScrollPolicy	Mostra ou não a barra de rolagem horizontal. Esse valor poderá ser on , off ou auto . O valor padrão é: auto .
verticalLineScrollSize	Indica o número de unidades que uma barra de rolagem vertical é movida cada vez que o botão da seta é clicado. O valor padrão é 4 .
verticalPageScrollSize	Indica o número de unidades que uma barra de rolagem vertical é movida cada vez que a trilha da barra é clicada. O valor padrão é 0 .
verticalScrollPolicy	Mostra ou não a barra de rolagem vertical. Esse valor poderá ser on , off ou auto . O valor padrão é auto .
visible	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: true .

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 e criar aplicações mais interativas, conforme descritas abaixo:

As propriedades do componente List

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

Propriedade	Descrição
iconField	É uma string que representa o campo do item que fornece o ícone para o item. O valor padrão é: icon .
iconFunction	Define uma função a ser usada para obter o ícone para o item. O valor padrão é: null .
labelField	É uma string que representa o nome do campo no objeto dataProvider a ser mostrado como o rótulo do campo TextInput da lista drop-down. O valor padrão é: label .
labelFunction	Configura a função a ser usada para obter o rótulo para o item. O valor padrão é: null .
rowCount	Define o número de linhas que deverão ser mostradas parcialmente em uma lista de um componente List .
rowHeight	É um número que define a altura de cada linha em uma lista, em pixels. O valor padrão é: 20 .

Os métodos do componente List

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

Método	Descrição
List	Cria uma nova instância do componente List .
itemToLabel	Recupera a string que é mostrada para o objeto de dados dado baseado nas propriedades labelField e labelFunction .
scrollToIndex	Rola a lista para o item no índice especificado. Se o índice estiver fora da faixa, a posição do scroll não será alterada.

Os eventos do componente List

Embora o **List** 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 List

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

Exemplo 1

O preenchimento dos itens manualmente, ou seja, utilizando o painel **Properties** é ideal para uma pequena quantidade de itens, uma vez que o processo é um pouco trabalhoso, principalmente quando se quer acrescentar ou retirar algum item, além de não ser prático e muito limitado. Nesse exemplo mostraremos como preencher um componente **List** manualmente. Vejamos como fazer isso:

Suponha a lista abaixo de alguns estados do Brasil e suas respectivas siglas:

- Acre - AC
- Amazonas - AM
- Bahia - BA
- Ceará - CE
- Goiás - GO
- Paraíba - PB
- Pernambuco - PE
- Rio Grande do Norte - RN
- Rio Grande do Sul - RS
- Rio de Janeiro - RJ
- São Paulo - SP

Para preencher o componente utilizando esse método, faça o seguinte:

1. Crie um novo documento (ActionScript 3.0);
2. Abra o painel **Components (Ctrl + F7)** e arraste um componente **List** para o palco. Centralize-o no palco e dimensione-o para o tamanho 140 x 160 pixels;
3. Nomeie a camada atual para: **componentes**;
4. Selecione o componente, e no painel **Component Inspector (Shift + F7)**, na guia **Parameters**, clique no parâmetro **dataProvider**, e em seguida, clique no ícone localizado na margem direita conforme mostra a **Figura 9.1** abaixo;

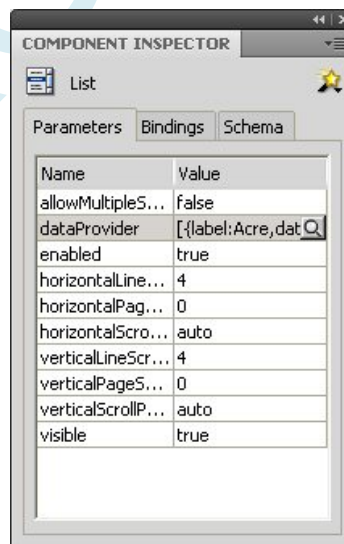


Figura 9.1 – Painel Component Inspector (guia Parameters).

5. Será mostrada uma janela, conforme mostra a **Figura 9.2** a seguir:

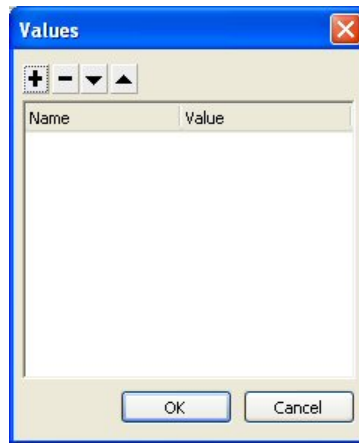


Figura 9.2 – Janela para preenchimento dos dados.

6. Para iniciar o preenchimento, clique no sinal de adição (+), e em seguida, na linha onde está escrito **label** preencha com os nomes dos estados, e na linha **data**, preencha com as siglas dos estados. Para adicionar um novo item clique novamente no sinal de adição (+), e assim por diante até concluir a lista. Veja exemplo na **Figura 9.3** abaixo:

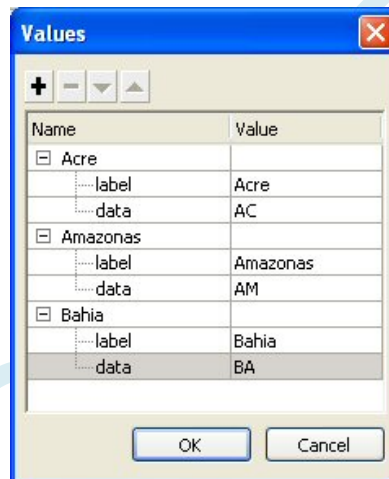


Figura 9.3 – Janela para preenchimento das informações.

7. Clique em **OK** quando terminar;
8. Execute a aplicação e confira o resultado com o mostrado na **Figura 9.4** abaixo:



Figura 9.4 – Componente List em execução.

9. Tente selecionar mais de um item da lista. Isso não será possível porque o parâmetro **allowMultipleSelection** está configurado para **false**.

Nota: O componente **List** não coloca os itens em ordem alfabética automaticamente. Para fazer isso, você poderá seleccionar o item desejado utilizando os botões das setas 'para cima' e 'para baixo' para colocá-los na devida ordem.

Exemplo 2

Nesse exemplo mostraremos como preencher um componente **List** através do ActionScript em tempo de execução utilizando o método **addItem()** para inserir um item de cada vez, ou seja, cada item em uma linha de código individual. Vejamos como fazer isso:

1. Crie um novo documento (ActionScript 3.0);
2. Abra o painel **Components** e arraste um componente **List** para o palco. Posicione-o em algum lugar e dimensione-o para **100 x 160** pixels). Isso pode ser feito a qualquer momento;
3. Selecione o componente, e no **Inspetor de Propriedades**, crie um nome de instância para o componente. Por exemplo: **listaMeses**;
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();  
//  
listaMeses.addItem({label:"Janeiro"});  
listaMeses.addItem({label:"Fevereiro"});  
listaMeses.addItem({label:"Março"});  
listaMeses.addItem({label:"Abril"});  
listaMeses.addItem({label:"Maio"});  
listaMeses.addItem({label:"Junho"});  
listaMeses.addItem({label:"Julho"});  
listaMeses.addItem({label:"Agosto"});  
listaMeses.addItem({label:"Setembro"});  
listaMeses.addItem({label:"Outubro"});  
listaMeses.addItem({label:"Novembro"});  
listaMeses.addItem({label:"Dezembro"});
```

7. Execute a aplicação e confira o resultado com o da **Figura 9.5** abaixo:

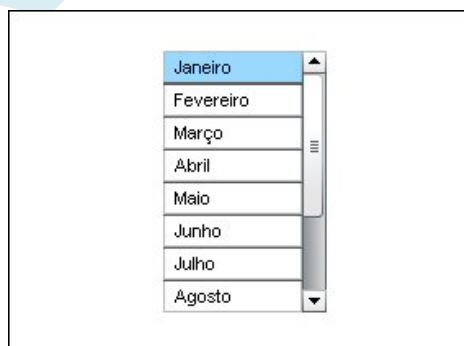


Figura 9.5 – Componente List preenchido em tempo de execução.

Exemplo 3

Utilizando o método do exemplo anterior o código fica muito extenso caso a lista de itens seja muito grande. Por isso, ao invés de utilizar um item por linha, nesse exemplo

mostraremos como preencher esse componente utilizando uma matriz com os nomes dos meses. Dessa forma o código fica mais reduzido e mais fácil de alterar qualquer informação. Vejamos como ficaria o exemplo anterior utilizando uma matriz para armazenar os dados.

1. Crie um novo documento (ActionScript 3.0);
2. Abra o painel **Components** e arraste um componente **List** para o palco. Posicione-o em algum lugar e dimensione-o de forma que todos os itens da lista possam ser vistos totalmente;
3. Selecione o componente;
4. Abra o painel **Properties** e crie um nome para a instância do componente na caixa de texto **Instance Name**. Por exemplo: **listaMeses**;
5. Nomeie a camada atual para: **componentes**;
6. Crie uma nova camada e chame-a de: **ações**;
7. No primeiro frame dessa camada insira o seguinte código:

```
stop();  
//  
var meses:Array = new Array("Janeiro", "Fevereiro", "Março", "Abril", "Maio", "Junho",  
"Julho", "Agosto", "Setembro", "Outubro", "Novembro", "Dezembro");  
for (x=0; x<meses.length; x++) {  
    listaMeses.addItem({label:meses[x]});  
}
```

Vejamos os comentários desse código:

Na linha:

```
meses = ["Janeiro", "Fevereiro", "Março", "Abril", "Maio", "Junho", "Julho", "Agosto",  
"Setembro", "Outubro", "Novembro", "Dezembro"];
```

Criamos uma matriz para os meses do ano e armazenamos na variável '**meses**'.

No bloco de código abaixo:

```
for (x=0; x<meses.length; x++) {  
    listaMeses.addItem(meses[x]);  
}
```

Criamos um loop variando de 0 até o tamanho da matriz '**meses**', para preencher o componente **List** utilizando o método **addItem()**, já comentado anteriormente.

8. Execute a aplicação e confira se está igual com o da **Figura 9.6** acima.



Figura 9.6 – Componente List preenchido em tempo de execução.

Exemplo 4

Além dos métodos de preenchimento utilizados para um componente **List** discutidos anteriormente, você também poderá utilizar dados externos ao Flash, como um arquivo texto ou um banco de dados. Nesse exemplo mostraremos como preencher esse componente utilizando um arquivo texto externo no formato **.as** criado no próprio Flash. Esse método reduz muito o tamanho do arquivo original, já que a fonte de toda a informação se encontra em um arquivo separado. É bom lembrar que esse arquivo texto deve estar no mesmo diretório do arquivo da aplicação, a menos que você indique o endereço absoluto do mesmo. Vejamos como fazer isso:

1. Inicialmente crie um novo arquivo no Flash através do menu: **File > New** ou (**Ctrl + N**);
2. Na janela apresentada **New Document**, selecione **ActionScript file** para criar um arquivo texto com a extensão **.as**. Clique em **OK** para confirmar;
3. Será criada uma janela de edição com o nome **Script-1**;
4. Salve esse arquivo como: **sites.as**;
5. Nessa janela, digite o seguinte código:

```
stop();
// Matriz para a relação de sites.
var lista:List = new List();
addChild(lista);
//
lista.move(30,30);
lista.setSize(120,140);
//
var lista_sites:Array = new Array("Copas do Mundo", "Yahoo", "Google", "Submarino",
"Americanas", "Globo", "Fox Films", "Kirupa", "Flashkit", "Adobe");
lista_sites.sortOn("label");
for (var i=0; i<lista_sites.length; i++) {
    lista.addItem({label:lista_sites[i]});
}
```

Vejamos os comentários sobre esse código:

Nas linhas:

```
var lista:List = new List();
addChild(lista);
```

Criamos uma nova instância da classe **List** com o seu respectivo construtor e armazenamos na variável chamada **lista**. Em seguida inserimos o componente no palco através do método **addChild()**.

Nas linhas seguintes:

```
lista.move(30,30);
lista.setSize(120,140);
```

Posicionamos o componente **List** no palco através do método **move()** e definimos o seu tamanho através do método **setSize()**.

Na linhas abaixo:

```
var lista_sites:Array = new Array("Copas do Mundo", "Yahoo", "Google", "Submarino",
"Americanas", "Globo", "Fox Films", "Kirupa", "Flashkit", "Adobe");
lista_sites.sortOn("label");
```

Criamos uma matriz chamada **lista_sites** para armazenar os nomes de alguns sites. Em seguida colocamos os elementos dessa matriz em ordem alfabética através do método **sortOn()**.

No bloco de código:


```
// Loop para preenchimento do componente List.
for (var i=0; i<lista_sites.length; i++) {
    lista.addItem({label:lista_sites[i]});
}
```

Criamos um loop variando de 0 (índice inicial da matriz **lista_sites**) até o tamanho da mesma, para preencher o componente **List** utilizando o método **addItem()**.

Agora criaremos o arquivo FLA no Flash:

1. Crie um novo documento (ActionScript 3.0);
2. Nomeie a camada atual para: **ações**;
3. No primeiro frame dessa camada insira o seguinte código:

```
stop();
//
include "sites.as"
import fl.controls.List;
```

Vejamos os comentários desse código:

Na linha:

```
include "sites.as"
```

Incluímos a diretiva **include** cuja finalidade é anexar o arquivo especificado no documento principal.

Na linha seguinte:

```
import fl.controls.List;
```

Importamos a classe do componente **List** tendo em vista que estamos utilizando-o em tempo de execução.

6. Execute a aplicação e confira se está igual com o da **Figura 9.7** acima.

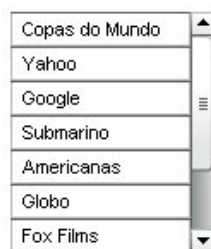


Figura 9.7 – Componente List preenchido em tempo de execução através de um arquivo externo no formato .as.

Exemplo 5

Nesse exemplo utilizaremos um componente **List** juntamente com um componente **Slider** em tempo de execução, de forma que quando o valor do **Slider** for alterado, a quantidade de itens visíveis no **List** seja igual a esse valor. Os dados do **List** podem ser provenientes de quaisquer fontes. Vejamos como fazer isso:

1. Crie um novo documento (ActionScript 3.0);

2. Abra o painel **Components** e arraste um componente **List** e um componente **Slider** para a biblioteca;
3. Nomeie a camada atual para: **ações**;
4. Abra o painel **Actions**, e digite o seguinte código no primeiro frame dessa camada:

```
stop();
//
import fl.controls.Slider;
import fl.controls.List;
import fl.data.DataProvider;
import fl.events.SliderEvent;
//
var dp:DataProvider = new DataProvider();
var i:int;
//
for (i=0; i<9; i++) {
    dp.addItem({label:"Item " + i});
}
//
var meuSlider:Slider = new Slider();
meuSlider.move(30, 30);
meuSlider.tickInterval = 1;
meuSlider.snapInterval = 1;
meuSlider.minimum = 2;
meuSlider.maximum = 9;
meuSlider.liveDragging = true;
addChild(meuSlider);
//
meuSlider.addEventListener(SliderEvent.CHANGE, alteraLista);
//
var lista:List = new List();
lista.dataProvider = dp;
lista.move(meuSlider.x, meuSlider.y + 20);
lista.setSize(100, 50);
addChild(lista);
//
function alteraLista(event:SliderEvent):void {
    lista.rowCount = meuSlider.value;
}
```

Vejamos como funciona o código:

Nas linhas:

```
import fl.controls.Slider;
import fl.controls.List;
import fl.data.DataProvider;
import fl.events.SliderEvent;
```

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

Nas linhas a seguir:

```
var dp:DataProvider = new DataProvider();
var i:int;
```

Criamos uma nova instância da classe **DataProvider** que deverá armazenar os dados que preencherão o **List**. Em seguida, criamos uma variável **i** do tipo inteiro, para ser utilizada no loop de preenchimento das informações armazenadas em **dp**.

O bloco de código a seguir:

```
for (i=0; i<9; i++) {
```

```
    dp.addItem({label:"Item " + i});
}
```

Criamos um loop utilizando a variável **i**, declarada anteriormente, para armazenar os itens em **dp** utilizando o método **addItem()**.

No bloco de código seguinte:

```
var meuSlider:Slider = new Slider();
meuSlider.move(30, 30);
meuSlider.tickInterval = 1;
meuSlider.snapInterval = 1;
meuSlider.minimum = 2;
meuSlider.maximum = 9;
meuSlider.liveDragging = true;
addChild(meuSlider);
```

Criamos uma nova instância do componente **Slider** de nome **meuSlider**, e em seguida definimos os seguintes parâmetros para ela:

- A sua posição no palco.
- O valor do espaçamento entre os valores.
- O intervalo dos valores.
- O valor mínimo do slider
- O valor máximo do slider.

Feito isso, configuramos a propriedade **liveDragging** para **true** para que as alterações a serem executadas durante a mudança dos valores do **Slider** ocorram em tempo real, e logo após, inserimos o componente no palco utilizando o método **addChild()**.

Na linha seguinte:

```
meuSlider.addEventListener(SliderEvent.CHANGE, alteraLista);
```

Utilizamos o evento **CHANGE** vinculado à instância **meuSlider**, de forma que quando o valor do **Slider** for alterado, a função **alteraLista** seja executada.

No bloco de código abaixo:

```
var lista:List = new List();
lista.move(meuSlider.x, meuSlider.y + 20);
lista.setSize(100, 50);
lista.dataProvider = dp;
addChild(lista);
```

Criamos uma nova instância do componente **List** de nome **lista**, e em seguida definimos os seguintes parâmetros para ela:

- A sua posição no palco em relação ao componente **Slider**.
- O seu tamanho através do método **setSize()**.

Feito isso, configuramos a propriedade **dataProvider** para preencher as informações que foram armazenadas em **dp** quando da execução do loop, e logo após, inserimos o componente no palco utilizando o método **addChild()**.

Finalmente, nas linhas:

```
function alteraLista(event:SliderEvent):void {
    lista.rowCount = meuSlider.value;
}
```

Criamos a função **alteraLista** que deverá alterar o número de itens do **List** de acordo com o valor selecionado no **Slider** em tempo real quando a mesma for executada.

5. Execute a aplicação e confira se está igual com o da **Figura 9.8** a seguir:

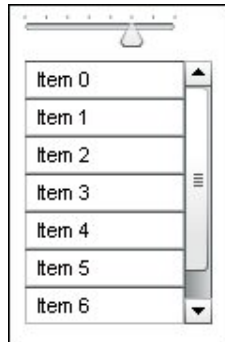


Figura 9.8 – Componentes List em execução.

Exercícios de Fixação

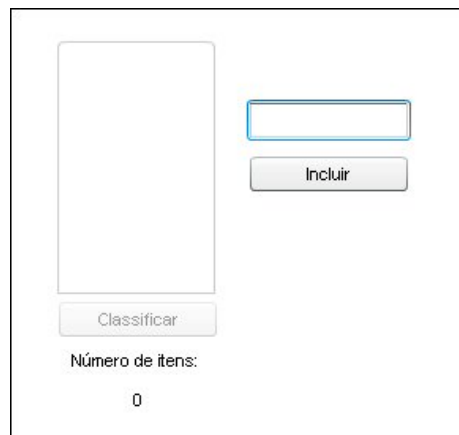
1. Que tecla ou combinação de teclas abaixo utilizamos juntamente com o mouse para selecionarmos vários itens alternados em um componente **List**?
 - a) Shift
 - b) Ctrl
 - c) Alt
 - d) Ctrl + Shift
 - e) Ctrl + Alt
2. Qual dos métodos abaixo é utilizado para adicionar um item em um componente **List**?
 - a) insertItem()
 - b) createItem()
 - c) includeItem()
 - d) addIndex()
 - e) addItem()
3. Qual das alternativas abaixo cria um componente **List** em tempo de execução, cuja instância se chama '**lista**'?
 - a) `var lista:List = new.List();`
 - b) `var lista.List = newList();`
 - c) `var lista-List = new List();`
 - d) `var lista:List = newList();`
 - e) `var lista:List = new List();`
4. Qual das opções abaixo altera a cor do texto dos itens de um componente **List**, cuja instância se chama **lista**?
 - a) `lista.setStyle("textColor", "0x0000ff");`
 - b) `lista.setText ("color", "0x0000ff");`
 - c) `lista.setStyleColor("text", "0x0000ff");`
 - d) `lista.setStyle("color", "0x0000ff");`
 - e) `lista.setStyleText("color", "0x0000ff");`
5. Que tecla ou combinação de teclas abaixo utilizamos juntamente com o mouse para selecionarmos vários itens contíguos em um componente **List**?

- a) Alt + Shift
 - b) Alt + Ctrl
 - c) Alt
 - d) Ctrl
 - e) Shift
6. Qual o valor padrão do parâmetro **rowHeight** de um componente **List**?
- a) 10
 - b) 15
 - c) 20
 - d) 25
 - e) 30
7. Qual a combinação de teclas que podemos utilizar para selecionar diretamente o primeiro item de um componente **List**?
- a) Ctrl + End
 - b) Alt + Home
 - c) Ctrl + Home
 - d) Shift + End
 - e) Alt + End
8. Que classe devemos importar para que possamos criar uma instância de um componente **List** em tempo de execução?
- a) flash:controls.List;
 - b) fl.controls:List;
 - c) fl.control.List;
 - d) fl.controls.List;
 - e) flash.controls.List;
9. O parâmetro **allowMultipleSelection** de um componente **List** estando configurado para **true**, indica que o usuário poderá selecionar vários itens alternados da lista:
- a) utilizando as teclas Alt + Shift e o mouse.
 - b) utilizando a tecla Alt e o mouse.
 - c) utilizando as teclas Ctrl + Alt e o mouse.
 - d) utilizando a tecla Shift e o mouse.
 - e) utilizando a tecla Ctrl e o mouse.
10. Qual o tamanho padrão de um componente **List**?
- a) 50 x 100 pixels
 - b) 100 x 50 pixels
 - c) 50 x 50 pixels
 - d) 50 x 150 pixels
 - e) 100 x 100 pixels

Exercícios Propostos

1. Crie um documento com um componente **List**, dois componentes **Button**, um componente **TextInput** e dois componentes **Label**. Sendo um botão para incluir o texto digitado no **TextInput** no componente **List**, um botão para classificar a lista, um **Label** para informar a quantidade de itens da lista e um outro **Label** para a mensagem. O

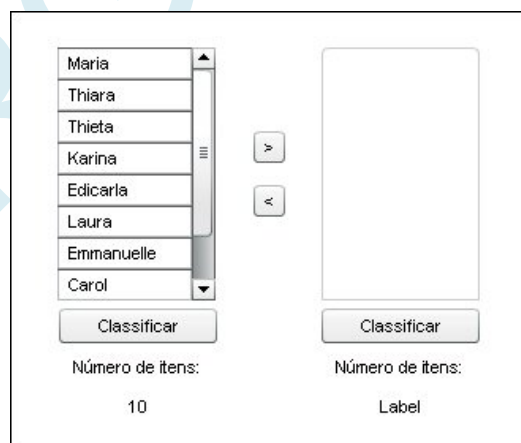
botão **Classificar** deverá iniciar desabilitado, e somente será habilitado quando houver pelo menos um item na lista. Veja layout sugerido na figura a seguir:



2. Crie um documento com dois componentes **List**, quatro componentes **Button** e quatro componentes **Label**. Sendo que o primeiro componente **List** deverá ser preenchido com 10 itens provenientes de uma matriz, de forma que:

- Quando a aplicação for executada, os botões ">" e "<", juntamente com o botão **Classificar** da segunda lista deverão ficar desabilitados. O botão ">" só deverá ficar habilitado quando algum item da primeira lista for selecionado. Da mesma forma, os botões "<" e **Classificar** da segunda lista só deverão ficar habilitados quando houver pelo menos um item nessa lista.
- Quando o usuário selecionar um item da primeira lista e clicar no botão ">", esse item seja removido da primeira lista e incluído na segunda lista, e vice-versa, e os rótulos que informam a quantidade de itens de cada lista sejam atualizados em cada operação.
- Quando o usuário clicar em um dos botões **Classificar**, a respectiva lista deverá ser classificada em ordem alfabética crescente.

Veja uma sugestão de layout para esse exercício:



3. Crie um documento com um componente **List**, um componente **TextInput**, dois componentes **Button** e dois componentes **Label**. De forma que:

- O componente **List** deverá ser preenchido com informações digitadas pelo usuário através do **TextInput**.
- Os dois botões deverão classificar a lista em ordem alfabética crescente ou decrescente respectivamente.
- Os dois labels servirão para informar o número de itens incluídos na lista:

Veja uma sugestão de layout para esse exercício:

Maria
Joana

Classificar (A-Z)
Classificar (Z-A)

Número de itens:
2

JORGE ELDER