



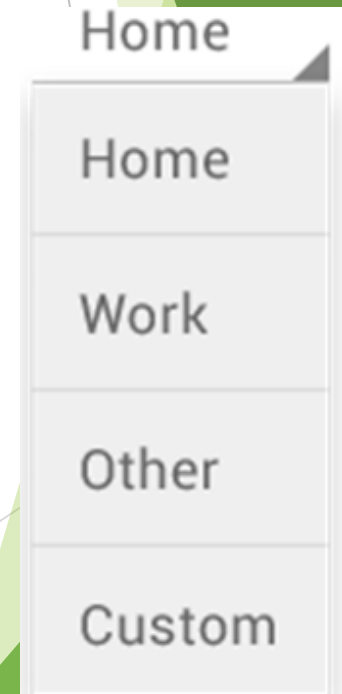
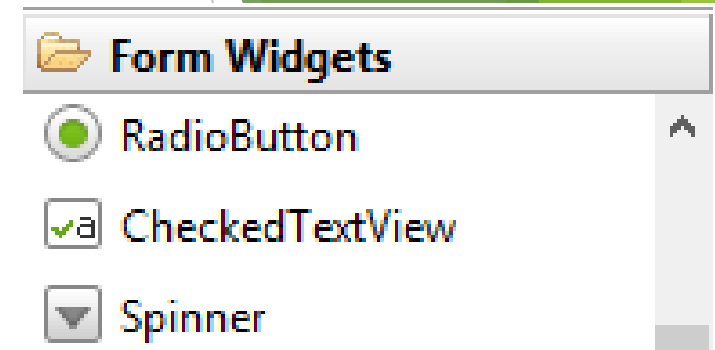
Programação para Android

Aula 05: Spinner e ListView

Spinner

- ▶ Spinners é um widget que permite o usuário selecionar um valor a partir de um conjunto de dados. No estado padrão, um spinner mostra seu valor atualmente selecionado.
- ▶ Um toque no spinner exibe um menu com todos os outros valores disponíveis, a partir do qual o usuário pode selecionar um novo.
- ▶ Na paleta de Widgets é possível incluir um Spinner ou podemos adicioná-lo via código XML

```
<Spinner  
    android:id="@+id/spTimes"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
/>
```



Spinner

- ▶ Podemos preencher um spinner de diversas formas:
 - ▶ Através de um ArrayList,
 - ▶ Através de um Array de string definido em um arquivo resources,
 - ▶ Através de uma consulta ao banco de dados (veremos isto quando estudarmos sobre persistência de dados).
 - ▶ Outros
- ▶ Estudaremos o preenchimento através de um array de string definido em um arquivo resources.
- ▶ No código Java, para preencher o spinner com uma lista de opções, precisamos especificar um ArrayAdapter e depois associa-lo ao spinner desejado.

Spinner - Populando

- 1ª Parte: Criação de um Array de String em res -> values -> strings

```
<string-array name="times">  
    <item>Atlético MG</item>  
    <item>Atlético PR</item>  
    <item>Botafogo</item>  
    <item>Corinthians</item>  
    <item>Cruzeiro</item>  
    <item>Flamengo</item>  
    <item>Fluminense</item>  
    <item>Grêmio</item>  
    <item>Internacional</item>  
    <item>Santos</item>  
    <item>São Paulo</item>  
    <item>Vasco</item>  
    <item>Vitória</item>  
</string-array>
```

Spinner - Populando

- ▶ 2º Passo: associar o Array de String ao spinner.
 - ▶ No método onCreate da activity devemos relacionar a nossa fonte de dados (array de string) a um spinner desejado. Neste caso usaremos um ArrayAdapter.

```
Spinner spTimes = (Spinner)
findViewById(R.id.spTimes);
// Cria um ArrayAdapter usando um array de string e
um layout padrão de spinner
ArrayAdapter adapter =
ArrayAdapter.createFromResource(this,
R.array.times,
android.R.layout.simple_spinner_item);
//alterar a fonte de dados(adapter) do Spinner
spTimes.setAdapter(adapter);
```

Atletico MG

Atletico MG

Atletico MG

Atletico PR

Botafogo

Corinthians

Cruzeiro

Flamengo

Fluminense

Grêmio

Internacional

Santos

São Paulo

Vasco

Vitória

Spinner

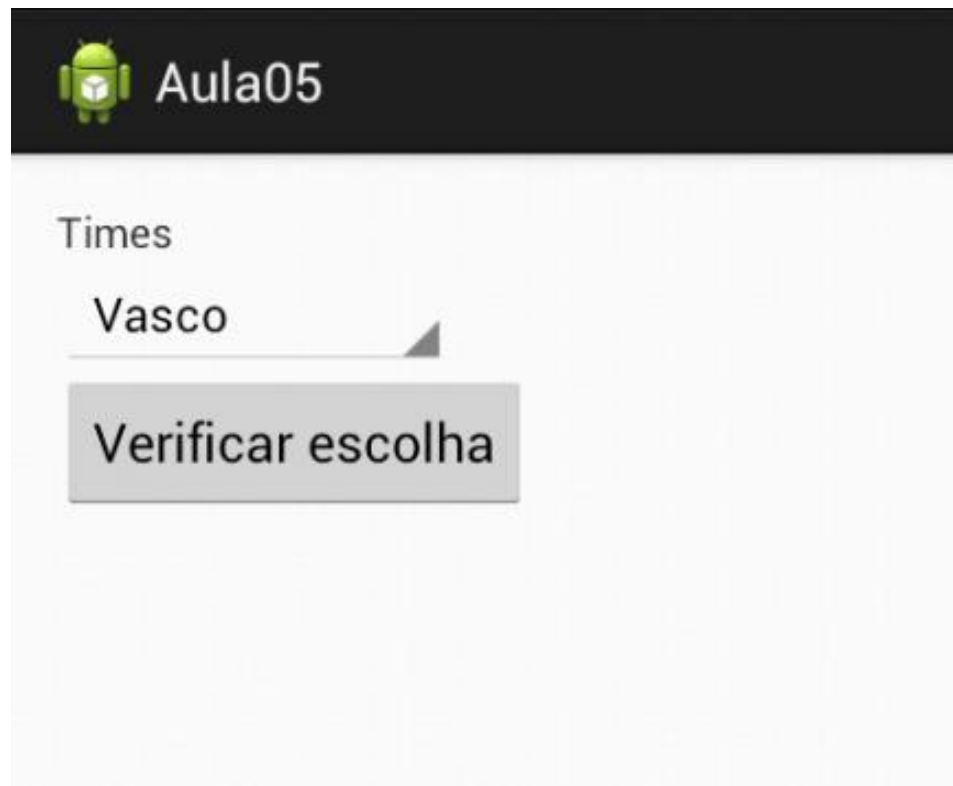
- ▶ Para capturar o texto selecionado usamos o método `getSelectedItem()`

```
String texto;  
texto = spTimes.getSelectedItem().toString();
```

- ▶ Para capturar a posição(número do item selecionado - começando de 0) usamos o método `getSelectedItemPosition()`

```
int a;  
a = spTimes.getSelectedItemPosition();
```

Spinner - Exemplo 01



Spinner - Listener onItemSelected

- ▶ Para manipular o evento de um Spinner quando o usuário alterar o item selecionado, devemos criar um listener específico para esta função..
- ▶ Neste caso usamos um listener para um AdapterView

```
AdapterView.OnItemSelectedListener escolha = new AdapterView.OnItemSelectedListener() {
```

- ▶ Em seguida, programaremos o evento relacionado ao item selecionado. **OBS:** Ao incluir o listener (`AdapterView.OnItemSelectedListener`) o Netbeans irá incluir um código automaticamente relacionando ao item selecionado. Devemos programar nosso código no método:

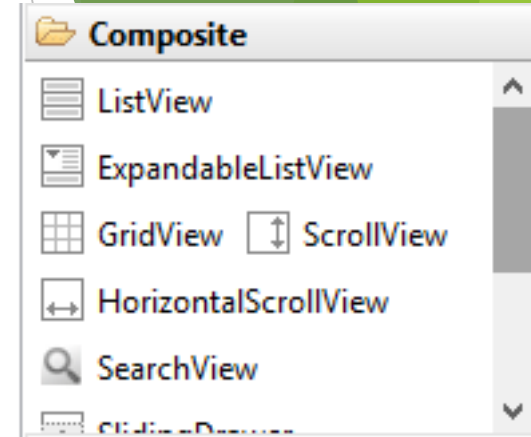
```
public void onItemSelected
```

- ▶ Após programar o clique é necessário alterar a propriedade `setOnItemSelectedListener` do Spinner:

```
spTimes.setOnItemSelectedListener(escolha);
```


ListView

- ▶ Um ListView é um Widget que mostra itens em uma lista de rolagem vertical.
- ▶ Na paleta de Widgets (Composite) é possível incluir um ListView ou podemos adiciona-lo via código XML



```
<ListView  
    android:id="@+id/lvTimes"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content" />
```

- ▶ OBS: Uma propriedade importante da ListView é a choiceMode. Usamos este atributo quando queremos transformar uma ListView em uma lista de seleção.

```
android:choiceMode="singleChoice"
```

ListView

- ▶ Podemos preencher um ListView de diversas formas:
 - ▶ Através de um ArrayList,
 - ▶ Através de um Array de string definido em um arquivo resources,
 - ▶ Através de uma consulta ao banco de dados (veremos isto quando estudarmos sobre persistência de dados).
 - ▶ Outros
- ▶ Estudaremos o preenchimento através de um array de string definido em um arquivo resources.
- ▶ No código Java, para preencher a ListView com uma lista de opções, precisamos especificar um ArrayAdapter e depois associa-lo a ListView desejada.
- ▶ OBS 1: Neste Exemplo, usaremos a string de arrays já criada para popular o Spinner relacionado aos times de futebol.

ListView - Populando dados

► Populando dados a partir de um Array de String.

```
// Cria um ArrayAdapter usando um array de string e um
layout padrão de spinner
// Neste caso, o layout padrão adotado possui a opção do
usuário CHECAR um item
ArrayAdapter timesSerieA =
ArrayAdapter.createFromResource(this,
R.array.times, android.R.layout.simple_list_item_checked);
timesSerieA.setDropDownViewResource(android.R.layout.simple_list_item_checked);

//Cria um ListView e relaciona com o Widget criado no XML
ListView lvTimes;
lvTimes = (ListView)findViewById(R.id.lvTimes);
//Associa a fonte de dados timesSerieA - Array de string
lvTimes.setAdapter(timesSerieA);
```

Times série A - Brasileirão

Atletico MG ✓

Atletico PR ✓

Botafogo ✓

Corinthians ✓

Cruzeiro ✓

Flamengo ✓

ListView

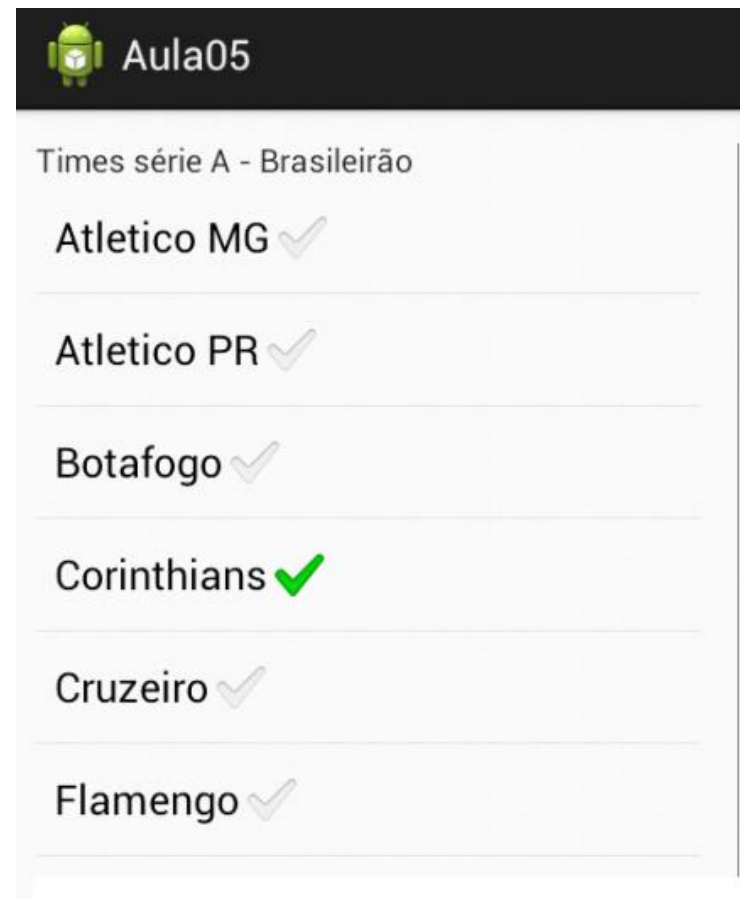
- ▶ Para capturar a posição de um elemento selecionado em uma ListView com a opção ChoiceMode configurada, usamos o método `getCheckedItemPosition()`

```
int a;  
a = lvTimes.getCheckedItemPosition();
```

- ▶ Para capturar o texto de um item selecionado usamos o método `getItemAtPosition` de uma posição selecionada..

```
String texto;  
texto =  
lvTimes.getItemAtPosition(a).toString();
```

ListView - Exemplo 02



Na próxima aula...

- Menus, Action Bar e Activitys

