

1. Aula 6

1.1. Arrays

Matrizes são usadas para armazenar vários valores em uma única variável, em vez de declarar variáveis separadas para cada valor.

Para declarar uma matriz, defina o tipo de variável com colchetes:

```
String[] cars;
```

Nós agora declaramos uma variável que contém um array de strings. Para inserir valores, podemos usar um literal de matriz - coloque os valores em uma lista separada por vírgula, dentro de chaves:

```
String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};
```

Você acessa um elemento da matriz referindo-se ao número do índice.

Esta declaração acessa o valor do primeiro elemento em carros:

```
String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};
```

```
System.out.println(cars[0]);
```

```
// Outputs Volvo
```

Os arrays ou matrizes, como são conhecidos pelo Java, fazem parte do pacote java.util, na coleção da API do Java. São objetos de recipientes que contém um número fixo de valores de um único tipo. O comprimento de um array é estabelecido quando criado, sendo que, após a criação, o seu comprimento fica fixo.

Cada item em um array é chamado de elemento, e cada elemento é acessado pelo número, o índice. Abaixo é mostrado como é acessado os seus elementos, lembrando que sempre sua numeração começa em 0.

Opala Comodoro	Chevette	Rural Willys	Golf
0	1	2	3

O nosso exemplo é um array de quatro carros, sendo que vamos exibir o elemento de índice 2.

```
String[] carros={"Opala","Chevette","Corcel II","Rural"};
```

```
EditText resultado1;
```

```
resultado1=(EditText)findViewById(R.id.resultado1);
```

```
resultado1.setText(carros[2]);
```

O resultado deste array é:

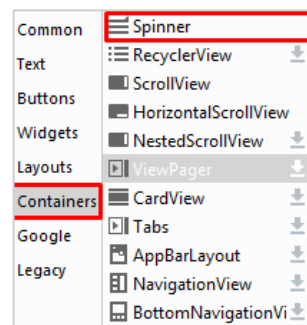
Corcel II

Componente Spinner

O componente Spinner é uma versão do Select (HTML) ou ComboBox.

Muitas vezes, temos a necessidade de personalizar componentes no decorrer do desenvolvimento de algum projeto. Na realidade, é isso que deixa nosso software com um lay-out diferente dos outros. Quando trabalhamos com a programação para Android, é inevitável não nos depararmos com uma situação dessas.

Neste artigo, irei personalizar um dos principais componentes, o “Spinner”. Ele fornece uma maneira rápida de selecionar um valor a partir de um conjunto de valores. No estado padrão, um “Spinner” mostra seu valor atualmente selecionado. Ao selecioná-lo, é exibido um Menu suspenso com todos os outros valores disponíveis, a partir do qual o usuário pode selecionar um novo. Ele se parece com o “Combobox”.



Encontramos o componente na categoria Containers.

Após o componente ser inserido na tela, ele recebe um ID automático. Nesse caso, ficou **spinner2**.

Criando uma variável:

Spinner cidades;

Variável recebendo conteúdo do componente:

cidades=(Spinner)findViewById(R.id.spinner2);

cidades – é a variável.

(Spinner) – é o componente.

findViewById – é para localizar um ID.

ArrayAdapter

No desenvolvimento Android, para exibir uma lista vertical com rolagem em nossa tela, utilizamos o componente ListView, e uma maneira eficiente de popular os dados neste componente é através do Adapter. Uma maneira simples de criar um adapter é usar um ArrayAdapter, pois o adapter converte cada objeto do ArrayList em itens (View) da ListView que iremos popular.

```
ArrayAdapter adapter=ArrayAdapter.createFromResource  
(this,R.array.materiais,android.R.layout.simple_spinner_item);
```

Segundo item do array é a variável “adapter”, que vai receber o conteúdo que vai ser localizado com o createFromResource, é passado o array.

Layout de item – escolhemos o simple_spinner_item, que é um simples TextView, como o layout para cada um dos itens.

setAdapter().

Permite listar o conteúdo do array.

1.2. Activity

A navegação entre telas/páginas é uma necessidade comum da maioria das aplicações, sejam elas mobile, web ou desktop. No Android, as telas da aplicação são representadas por activities.

A primeira situação a ser mostrada é a de navegação entre activities. Com ela, é possível iniciar novas activities, além de poder passar parâmetros para as activities iniciadas.

Veja o exemplo:

```
Intent intent = new Intent(this, NewActivity.class);  
startActivity(intent);
```

Na segunda linha, basta chamar o `startActivity`, passando a `intent` por parâmetro para que seja executada pelo Android a criação da nova activity.