

Ficha do Aplicativo

App05 – Aplicativo Cursos Fatec

Resumo

Neste App05 o usuário escolhe um curso dentre uma lista de cursos disponíveis nas unidades Fatecs e o aplicativo apresenta em quais Fatecs tal curso é oferecido.

Neste aplicativo vamos ver como criar e usar uma classe Java puro em nosso projeto Android.

Objetivos de Aprendizagem

1. Consolidar a organização de layout responsivo através do uso do LinearLayout.
2. Uso da classe Spinner.
3. Uso de resources contendo construção array de strings (string-array).
4. Uso de classes Java puro para construção de partes de uma aplicação.
5. Uso da classe ArrayList
6. Uso da classe StringBuilder

Dinâmica do Aplicativo

O usuário escolherá um curso de uma lista contida em um componente Spinner. Uma vez escolhido o curso, haverá um botão que quando clicado construirá uma lista de Fatecs que o oferecem. Essa lista de Fatecs será apresentada em um componente da classe TextView, que terá capacidade de rolagem

Lista de Activities do Aplicativo

Nome	Layout
MainActivity.java	activity_main.xml

Resources

strings.xml

```
<resources>
  <string name="app_name">Cursos Fatec</string>
  <string name="selecurso">Selecione o Curso</string>
  <string name="vercurso">Onde há tal Curso?</string>
  <string name="lblfatec">Fatecs Encontradas</string>
  <string-array name="cursos">
    <item>Análise e Desenvolvimento de Sistemas</item>
    <item>Jogos Digitais</item>
    <item>Segurança da Informação</item>
    <item>Gestão Comercial</item>
    <item>Comércio Exterior</item>
    <item>Secretariado</item>
    <item>Mecânica de Precisão</item>
  </string-array>
</resources>
```

colors.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <color name="purple_200">#FFBB86FC</color>
  <color name="purple_500">#FF6200EE</color>
  <color name="purple_700">#FF3700B3</color>
  <color name="teal_200">#FF03DAC5</color>
  <color name="teal_700">#FF018786</color>
  <color name="black">#FF000000</color>
  <color name="white">#FFFFFFFF</color>
  <color name="corFundo">#BDDADD</color>
  <color name="corVerTexto">#0E0F02</color>
  <color name="corBranca">#F0FCF8</color>
</resources>
```

Detalhamento da Activity

Layout: activity_main.xml

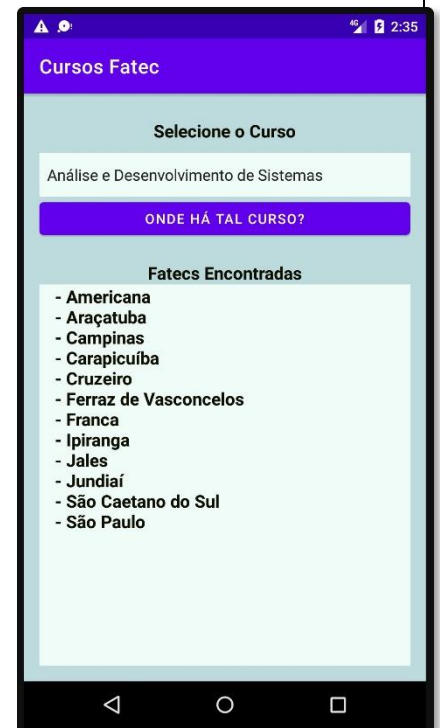
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:app="http://schemas.android.com/apk/res-auto"
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:background="@color/corFundo"
  android:orientation="vertical"
  android:padding="16dp"
  tools:context=".MainActivity">

  <TextView
    android:id="@+id/textView"
    android:layout_width="match_parent"
    android:layout_height="45dp"
    android:gravity="center"
    android:text="@string/selecurso"
    android:textColor="@color/corVerTexto"
    android:textSize="18sp"
    android:textStyle="bold" />

  <Spinner
    android:id="@+id/spnCursos"
    android:layout_width="match_parent"
    android:layout_height="45dp"
    android:background="@color/corBranca"
    android:entries="@array/cursos" />

  <Button
    android:id="@+id/btnCurso"
    android:layout_width="match_parent"
    android:layout_height="45dp"
    android:onClick="btnCursoOnClick"
    android:text="@string/vercurso" />

  <TextView
    android:id="@+id/lblFatec"
    android:layout_width="match_parent"
```



```
        android:layout_height="45dp"
        android:gravity="bottom|center_horizontal"
        android:text="@string/lblfatec"
        android:textColor="@color/corVerTexto"
        android:textSize="18sp"
        android:textStyle="bold" />

<TextView
    android:id="@+id/txtFatecs"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/corBranca"
    android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:scrollbarStyle="outsideInset"
    android:scrollbars="vertical"
    android:text=""
    android:textColor="@color/corVerTexto"
    android:textSize="18sp"
    android:textStyle="bold" />

</LinearLayout>
```

Código: MainActivity.java

```
/* Observação:  
O nome do package está omitido, porque ao fazer o seu projeto esse nome será outro.  
Os imports também estão omitidos. O motivo para isso é que a cada nova versão do Android Studio pode  
haver mudanças nos nomes dos caminhos da biblioteca. */  
  
public class MainActivity extends AppCompatActivity {  
  
    private Spinner spnCursos;  
    private TextView txtFatecs;  
    private CursosFatec cursosFatec;  
    private String cursoEscolhido;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        spnCursos = findViewById(R.id.spnCursos);  
        txtFatecs = findViewById(R.id.txtFatecs);  
        txtFatecs.setMovementMethod(new ScrollingMovementMethod());  
        cursosFatec = new CursosFatec();  
    }  
  
    @Override  
    protected void onSaveInstanceState(Bundle savedInstanceState) {  
        super.onSaveInstanceState(savedInstanceState);  
        savedInstanceState.putString("curso", cursoEscolhido);  
    }  
  
    @Override  
    protected void onRestoreInstanceState(Bundle savedInstanceState) {  
        super.onRestoreInstanceState(savedInstanceState);  
        cursoEscolhido = savedInstanceState.getString("curso");  
        carregaFatecs();  
    }  
  
    public void btnCursoOnClick(View view) {  
        cursoEscolhido = spnCursos.getSelectedItem().toString();  
        // cursoEscolhido = String.valueOf(spnCursos.getSelectedItem()); // também pode ser usado esta forma  
        carregaFatecs();  
    }  
  
    private void carregaFatecs() {  
        if (cursoEscolhido != null) {  
            ArrayList<String> listaFatecs = cursosFatec.getFatecs(cursoEscolhido);  
            StringBuilder s = new StringBuilder();  
            for (String fatec : listaFatecs) {  
                s.append(" - ");  
                s.append(fatec);  
                s.append("\n");  
            }  
            txtFatecs.setText(s);  
        }  
    }  
}
```

Código: CursosFatec.java – Classe Java puro

```
/* Observação:  
O nome do package está omitido, porque ao fazer o seu projeto esse nome será outro.  
Os imports também estão omitidos. O motivo para isso é que a cada nova versão do Android Studio pode  
haver mudanças nos nomes dos caminhos da biblioteca. */  
  
public class CursosFatec {  
    public ArrayList<String> getFatecs(String curso) {  
        ArrayList<String> fatecs = new ArrayList<>();  
        if (curso.equals("Análise e Desenvolvimento de Sistemas")) {  
            fatecs.add("Americana");  
            fatecs.add("Araçatuba");  
            fatecs.add("Campinas");  
            fatecs.add("Carapicuíba");  
            fatecs.add("Cruzeiro");  
            fatecs.add("Ferraz de Vasconcelos");  
            fatecs.add("Franca");  
            fatecs.add("Ipiranga");  
            fatecs.add("Jales");  
            fatecs.add("Jundiaí");  
            fatecs.add("São Caetano do Sul");  
            fatecs.add("São Paulo");  
        }  
        else if (curso.equals("Jogos Digitais")) {  
            fatecs.add("Americana");  
            fatecs.add("Carapicuíba");  
            fatecs.add("Ourinhos");  
            fatecs.add("São Caetano do Sul");  
        }  
        else if (curso.equals("Segurança da Informação")) {  
            fatecs.add("Americana");  
            fatecs.add("Ourinhos");  
            fatecs.add("São Caetano do Sul");  
        }  
        else if (curso.equals("Gestão Comercial")) {  
            fatecs.add("Adamantina");  
            fatecs.add("Araraquara");  
            fatecs.add("Assis");  
            fatecs.add("Guaratinguetá");  
            fatecs.add("Itaquaquetuba");  
            fatecs.add("Santana de Parnaíba");  
            fatecs.add("São Roque");  
        }  
        else if (curso.equals("Comércio Exterior")) {  
            fatecs.add("Barueri");  
            fatecs.add("Indaiatuba");  
            fatecs.add("Itapetininga");  
            fatecs.add("Praia Grande");  
            fatecs.add("Santo André");  
            fatecs.add("São Caetano do Sul");  
        }  
        else if (curso.equals("Secretariado")) {  
            fatecs.add("São Paulo");  
        }  
        else {  
            fatecs.add("{curso não encontrado}");  
        }  
        return fatecs;  
    }  
}
```

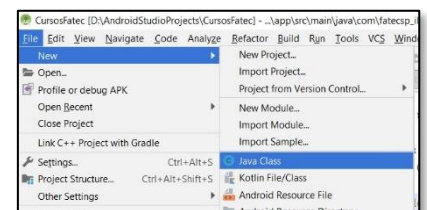
Notas Técnicas

Uso de Classes Java Puro em projetos Android

Sempre que for necessário, o programador pode criar novas classes em um projeto Android. Para isso, selecione a aba do seu código Java e use o comando de menu com o caminho File -> New -> Java Class que está exibido na figura ao lado.

Isto abrirá uma janela onde você irá fornecer um nome para a sua classe. Digite o nome CursosFatec e aperte Enter para finalizar. Pronto.

Uma vez criada a Classe, deve-se codificá-la normalmente seguindo as regras da Programação Orientada a Objetos e da linguagem Java. O código completo acima, nesta página, é a nossa classe para este projeto. Nela implementamos o método getFatecs() que tem o seguinte cabeçalho:



```
ArrayList<String> getFatecs(String curso)
```

Este método recebe um String com o nome do curso escolhido e retorna um vetor dinâmico, contendo os nomes das Fatecs que ministram esse curso. O vetor dinâmico será da classe `ArrayList<String>`.

O que é a classe ArrayList

Os vetores na linguagem Java têm tamanho fixo que deve ser especificado no momento em que a variável é instanciada. A classe `ArrayList` é um recurso criado para oferecer aos programadores Java uma ferramenta que permite trabalhar com vetores dinâmicos, ou seja, com variável quantidade de elementos.

A classe `ArrayList` se encarrega de criar internamente um vetor com um tamanho suficiente para copiar seus elementos correntes e, no caso de alteração de tais elementos, faz o redimensionamento do vetor interno sempre que necessário. O vetor interno da classe `ArrayList` é recriado quando ocorre uma dessas operações:

- adição de elemento no fim da lista além da capacidade dimensionada;
- adição de elemento no meio da lista;
- remoção de elemento;

Como a operação de recriação do vetor é custosa em termos de tempo de processamento, é recomendado que o programador procure dimensionar previamente o tamanho da lista, para tal, utilize o construtor sobrecarregado `ArrayList(int)` enviando como argumento a capacidade inicial da lista. A capacidade inicial padrão de um objeto `ArrayList` é de 10 elementos.

Manipular vetores estáticos em Java é muito trabalhoso para o programador.

Em C, pode-se utilizar as funções `malloc` e `realloc` e com pouco trabalho e algumas funções espertas pode-se redimensionar um vetor com muita facilidade. As linguagens mais modernas, como Python, possuem implementações totalmente flexíveis e poderosas de recursos como listas. Em Java, por outro lado, o trabalho é muito grande, pois não há listas prontas suportadas nativamente e não há comando para realocação de memória como o `realloc` de C.

Devido a essas dificuldades, com o tempo, foram sendo criadas bibliotecas voltadas para implementação de coleções de dados. Para conferir poder e flexibilidade foi criada a interface `List<>`. Em Java as interfaces são templates abstratos para a criação de classes. Interfaces contém apenas cabeçalhos de métodos e atributos constantes (variáveis definidas como `static` e `final`). Com isso a interface pré determina os métodos que a classe derivada deve obrigatoriamente implementar.

Dito isto, o `ArrayList<>` é uma implementação da interface `List<>`.

Note que introduzimos um novo símbolo: `<>`

Este é o chamado operador diamante (diamond operator) e serve para especificar o tipo de elemento que estará contido no `ArrayList`. Todos os elementos de um `ArrayList` serão sempre do mesmo tipo e será o tipo que for especificado dentro do operador diamante. Para saber mais acesse os links a seguir.

Sobre interfaces

https://www.w3schools.com/java/java_interface.asp

<https://docs.oracle.com/javase/tutorial/java/concepts/interface.html>

Sobre `List<>`

<https://docs.oracle.com/javase/tutorial/collections/interfaces/list.html>

<https://www.geeksforgeeks.org/list-interface-java-examples/>

Sobre `ArrayList<>`

https://www.w3schools.com/java/java_arraylist.asp

<https://docs.oracle.com/javase/8/docs/api/java/util/ArrayList.html>

A exibição da lista de Fatecs para o curso escolhido

A classe Java puro deste aplicativo – denominada `CursosFatec` – não chega a ser brilhante, mas tem o propósito de exemplificar de forma simples o uso de uma classe independente criada pelo programador. Essa classe contém o método `getFatecs()` que recebe o nome do curso e retorna um `ArrayList<String>` – ou seja, um vetor dinâmico cujos elementos são objetos `String` – com os nomes das Fatecs que ministram o curso.

O objeto `buscaFatecs` da classe `CursosFatec` está definido e instanciado logo no início da classe `MainActivity`. Isso significa que estará disponível para uso em todos os métodos implementados a seguir. No método `btnCursoOnClick()` é feita a chamada ao método `getFatecs()` e a lista é obtida. O passo seguinte é transferir para o componente `TextView` os nomes das Fatecs retornados por esse chamado. Para essa transferência foi utilizado um `StringBuilder`. Como o objetivo era concatenar é concatenar um grupo de `Strings`, um objeto da classe `StringBuider` é adequado pois realiza a tarefa de maneira menos custosa para o processador.

Para saber mais sobre `StringBuilder`:

<https://www.devmedia.com.br/a-classe-stringbuilder-em-java/25609>

<https://docs.oracle.com/javase/7/docs/api/java/lang/StringBuilder.html>