

### Ficha do Aplicativo

#### App03 – Aplicativo Cursos Fatec

##### Resumo

Neste App03 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	layout/activity_main.xml

## Resources

strings.xml

```
<resources>
  <string name="app_name">Cursos Fatec</string>
  <string name="selcurso">Selecione um 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>
  </string-array>
</resources>
```

colors.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <color name="colorPrimary">#3949AB</color>
  <color name="colorPrimaryDark">#00574B</color>
  <color name="colorAccent">#D81B60</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"?>
<androidx.constraintlayout.widget.ConstraintLayout 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"
    tools:context=".MainActivity">

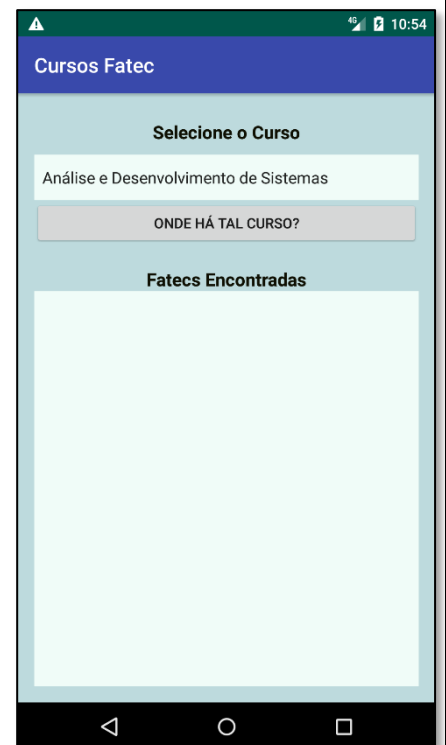
  <LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/corFundo"
    android:orientation="vertical"
    android:padding="16dp">

    <TextView
      android:id="@+id/textView"
      android:layout_width="match_parent"
      android:layout_height="45dp"
      android:gravity="center"
      android:text="@string/selcurso"
      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>

</androidx.constraintlayout.widget.ConstraintLayout>

```

### 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 CursosFatec buscaFatecs = new CursosFatec();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        TextView txtFatecs = findViewById(R.id.txtFatecs);
        txtFatecs.setMovementMethod(new ScrollingMovementMethod());
    }

    public void btnCursoOnClick(View view) {
        TextView txtFatecs = findViewById(R.id.txtFatecs);
        Spinner spnCursos = findViewById(R.id.spnCursos);
        String cursoEscolhido = String.valueOf(spnCursos.getSelectedItem());

        ArrayList<String> listaFatecs = buscaFatecs.getFatecs(cursoEscolhido);

        StringBuilder s = new StringBuilder();
        for(String x : listaFatecs) {
            s.append("- ");
            s.append(x);
            s.append("\n");
        }
        txtFatecs.setText(s);
    }
}

```

```

/* 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 {

    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("Itaquaquecetuba");
            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 {
            fatecs.add("{curso não encontrado}");
        }
        return(fatecs);
    }
}

```

## Notas Técnicas

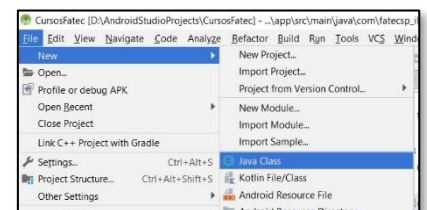
### Uso de Classes Java Puro em projetos Android

Sempre que for necessário, o programador pode criar novas classes em um projeto Android. Para isso deve usar o comando de menu com o caminho File -> New -> Java Class que está exibido na figura ao lado.

Isto abrirá uma janela onde se pode fornecer os parâmetros da classe, tais como Nome, Superclasse, Interface, etc.

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://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://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>