**Trabalhando com layouts XML em Android**

17 de abril de 2010 [**27 comentários**](http://www.felipesilveira.com.br/2010/04/trabalhando-com-layouts-xml-em-android/#comments)

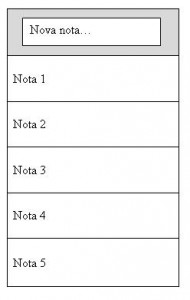
Olá pessoal, hoje vou falar sobre a definição do**layout de sua aplicação Android usando XML**.

Definir os layouts através dessa técnica é a forma mais comum de se começar o desenvolvimento de uma aplicação.

Para exemplificar os estudos que faremos a partir de agora, usaremos uma aplicação de exemplo, que construiremos juntos, a partir dos próximos posts.

A idéia é fazer uma aplicação onde o usuário poderá entrar com algumas anotações, e visualizar as últimas entradas.

O layout será mais ou menos assim:

[](http://www.felipesilveira.com.br/wp-content/uploads/2010/04/esboco.jpg)

esboço do layout

Criando o main.xml

Para novos projetos android, o arquivo main.xml já é automaticamente criado. Ele fica no diretório res/layout, com o conteúdo:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | <?xml version="1.0" encoding="utf-8"?>  <LinearLayout xmlns:android="<http://schemas.android.com/apk/res/android>"   android:orientation="vertical"   android:layout\_width="fill\_parent"   android:layout\_height="fill\_parent"  >  <TextView   android:layout\_width="fill\_parent"   android:layout\_height="wrap\_content"   android:text="@string/hello"   />  </LinearLayout> |

Neste arquivo temos contato com os primeiros elementos de um arquivo de layout XML:

* *LinearLayout*, que é apenas um container.
* *TextView*, que é um elemento de texto. Nesse caso está imprimindo a string cujo id é @string/hello. (Não se preocupe, falaremos sobre strings e seus ids à frente nesse curso)

Para criar um layout parecido com o rascunho do início do post, iremos inserir outros dois elementos:

* *EditText* – uma caixa de texto onde o usuário irá entrar com as anotações;
* *ListView* – uma lista de anotações previamente submetidas.

Assim, o nosso novo XML:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20 | <?xml version="1.0" encoding="utf-8"?>  <LinearLayout xmlns:android="<http://schemas.android.com/apk/res/android>"   android:orientation="vertical"   android:layout\_width="fill\_parent"   android:layout\_height="fill\_parent"  >   <EditText   android:id="@+id/edit\_box"   android:layout\_width="fill\_parent"   android:layout\_height="wrap\_content"   android:text="Nova nota..."  >   </EditText>   <ListView   android:id="@+id/notes\_list"   android:layout\_width="fill\_parent"   android:layout\_height="wrap\_content"  >   </ListView>  </LinearLayout> |

Carregando o arquivo XML na aplicação

Para que a nossa aplicação tenha o layout definido pelo arquivo XML, é preciso carregá-lo.

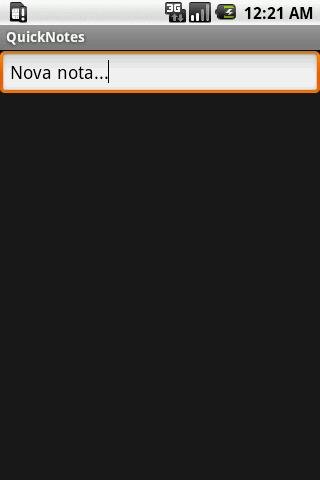
Isso é feito através da função setContentView(), como no código abaixo:

|  |  |
| --- | --- |
| 1  2  3  4  5 | public void onCreate(Bundle savedInstanceState) {      super.onCreate(savedInstanceState);      setContentView(R.layout.main);  } |

O parâmetro R.layout.main indica que o arquivo de layout a ser carregado é o main.xml. (Se o se arquivo se chamar abobrinha.xml, o parâmetro deverá ser R.layout.abobrinha)

É possível utilizar mais de um arquivo XML para uma mesma tela, para formar layouts mais sofisticados. Trataremos disso à frente nesse curso.

Compilando o nosso projeto e rodando no emulador, temos o seguinte resultado:

[](http://www.felipesilveira.com.br/wp-content/uploads/2010/04/device.png)

QuickNotes rodando no emulador