

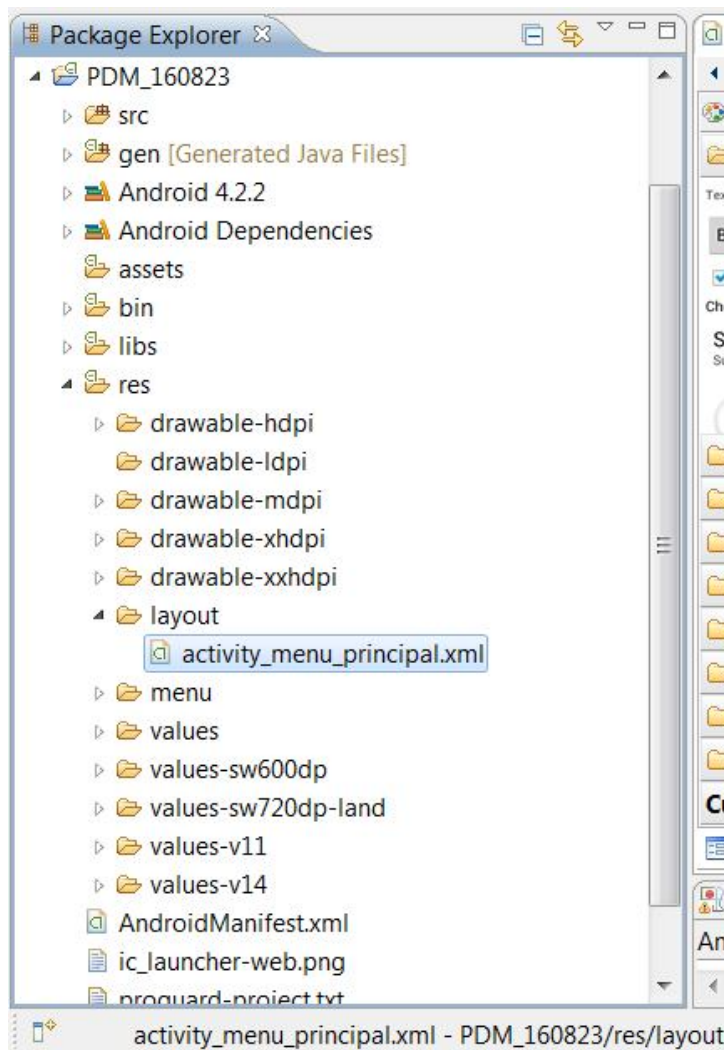
Prof. Menezes	PROGRAMAÇÃO MOBILE Estudo de Caso	Data 23/08/2016
------------------	---	--------------------

O desenvolvimento de software para dispositivos móveis diverge em vários aspectos do tradicional desenvolvimento de aplicações para desktop e Web. Quando no segundo caso não temos a preocupação com o tamanho da tela, no primeiro temos que alternar entre elas para atingirmos o nosso objetivo.

Para exemplificarmos o uso de múltiplas telas , iremos desenvolver nesta etapa o protótipo de um programa para lançamento de venda, o qual será formado por duas telas: menu principal e tela de lançamento.

Na tela principal, o usuário poderá optar por abri a tela de lançamento ou sair do aplicativo. Já no lançamento, o usuário informará os dados da venda de um produto, como o código do produto, quantidade e valor de venda.

Para testarmos, criaremos um projeto Android com o nome Cadastro. A Activity será chamada de *MenuPrincipalActivity.java*, já o arquivo XML da interface gráfica, chamado de *activity_menu_principal.xml*. A estrutura do projeto criado é apresentada abaixo:



Prof. Menezes	PROGRAMAÇÃO MOBILE Estudo de Caso	Data 23/08/2016
------------------	---	--------------------

O passo seguinte consiste na criação da interface gráfica do activity_menu_principal.xml.

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".MenuPrincipalActivity" >
```

```
<Button
    android:id="@+id/btLancamento"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"
    android:layout_marginLeft="18dp"
    android:layout_marginTop="23dp"
    android:text="Lançamento" />
```

```
<Button
    android:id="@+id/btSair"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/btLancamento"
    android:layout_below="@+id/btLancamento"
    android:layout_marginTop="28dp"
    android:text="Sair" />
```

```
</RelativeLayout>
```

Que representa a tela a seguir



Prof. Menezes	PROGRAMAÇÃO MOBILE Estudo de Caso	Data 23/08/2016
------------------	---	--------------------

Para tratamento dos botões , é necessária a codificação da classe Activity, responsável pela interface visual.

```
package com. exemplo. pdm_160823;
```

```
import android.os. Bundle;
import android.app. Activity;
import android.view. Menu;
import android.view. View;
import android.widget. Button;
```

```
public class MenuPrincipalActivity extends Activity {
```

```
    private Button btLancamento;
```

```
    private Button btSair;
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
        super. onCreate(savedInstanceState);
        setContentView(R.layout. activity_menu_principal);
```

```
        btLancamento = (Button) findViewById(R.id. btLancamento);
```

```
        btSair = (Button) findViewById(R.id. btSair);
```

```
        btLancamento.setOnClickListener( new View.OnClickListener() {
```

```
            @Override
```

```
            public void onClick(View arg0) {
                btLancamento.onClick();
```

```
            }
```

```
        });
```

```
        btSair.setOnClickListener(new View.OnClickListener() {
```

```
            @Override
```

```
            public void onClick(View arg0) {
                btSair.onClick();
```

```
            }
```

```
        });
```

```
    }
```

```
    @Override
```

```
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is
        present.
```

```
        getMenuInflater().inflate(R.menu. menu_principal, menu);
        return true;
```

```
    }
```

```
    public void btLancamentoonClick() {
```

```
    }
```

Prof. Menezes	PROGRAMAÇÃO MOBILE Estudo de Caso	Data 23/08/2016
------------------	---	--------------------

```
public void btSairOnClick() {  
    }  
}
```

Estamos prontos para inserirmos os eventos click's dos botões.

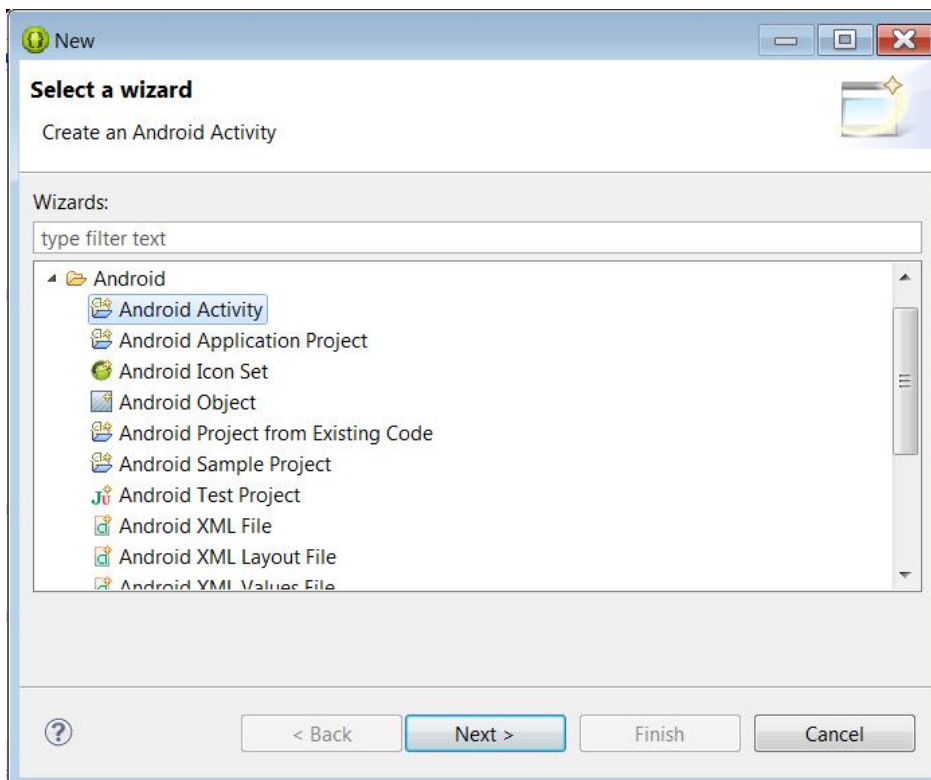
Classes Activity

Activities são classes do Android que possuem a função de realizar alguma atividade ou tarefa dentro da aplicação. Na maioria das vezes (porém nem sempre), as Activities são associadas à tela, ou seja, os dados informados em uma determinada tela são processados pela Activity correspondente a essa tela, assim se tivermos 10 telas, 10 Activities. Veja os comentários sobre a classe Activity em http://mz.pro.br/pdm/pdm_aula02.pdf.

Criando uma segunda tela para a aplicação.

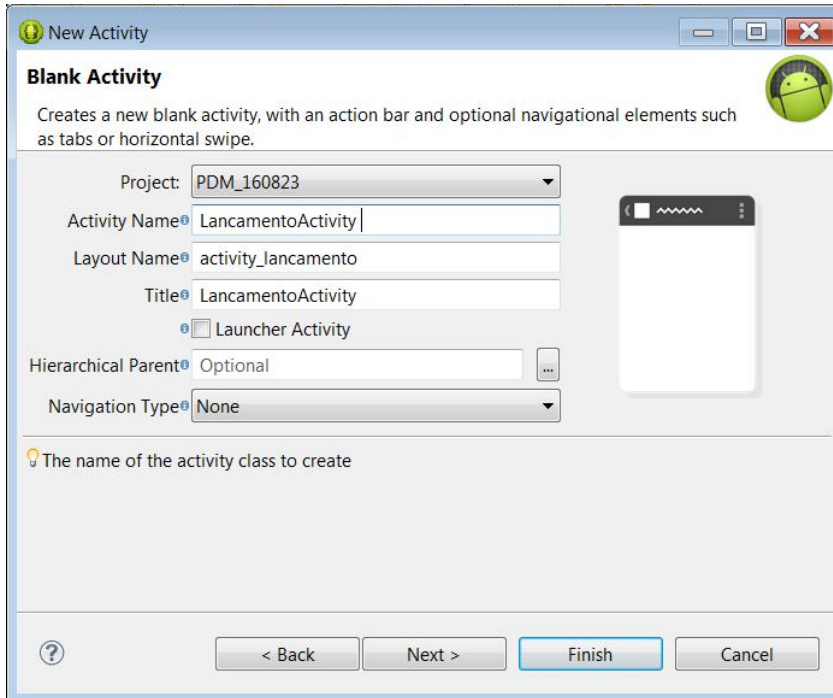
Para a apresentação da tela de lançamento, o primeiro passo é criar uma Activity que tratará a tela de lançamento. Uma Activity é uma classe Android.

Clicar no botão direito no projeto, escolhendo a opção New>>Other. Na janela apresentada, escolhemos a opção Android Activity, a qual se encontra dentro da categoria Android.



Prof. Menezes	PROGRAMAÇÃO MOBILE Estudo de Caso	Data 23/08/2016
------------------	---	--------------------

Na tela seguinte, deve-se selecionar um template para a criação da Activity . Para nosso exemplo selecione New Blank Activity. Na próxima tela, é informado o nome do Activity (.java), bem como o nome da tela (.xml). Para o exemplo, utilizamos LancamentoActivity (Activity Name) e activity_lancamento (Layout name), conforme abaixo:



Para criar uma Activity associada a uma tela, inicialmente, deve-se criar o arquivo XML associado. Seguindo, é necessário criar uma classe Java. Já criado por nós. Por fim, via arquivo manifest.xml (presente na pasta do projeto) deve-se incluir uma entrada informando que a classe Java criada é uma Activity, conforme abaixo.



O código conforme indicado acima, caso não exista, deve ser incluído pelo programador.

Após criar um tela e uma Activity, vamos atualizar o activity_lancamento.xml, conforme abaixo:

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin">
```

Prof. Menezes	PROGRAMAÇÃO MOBILE Estudo de Caso	Data 23/08/2016
------------------	---	--------------------

```

android:paddingLeft="@dimen/activi ty_hori zontal _margi n"
android:paddingRi ght="@dimen/activi ty_hori zontal _margi n"
android:paddingTop="@dimen/activi ty_verti cal _margi n"
tools:context=". LancamentoActi vi ty" >

```

```
<TextVi ew
```

```

    android: i d="@+i d/textVi ew1"
    android: layout_wi dth="wrap_content"
    android: layout_hei ght="wrap_content"
    android: layout_al i gnParentLeft="true"
    android: layout_al i gnParentTop="true"
    android: layout_margi nLeft="17dp"
    android: layout_margi nTop="16dp"
    android: text="Cod: " />

```

```
<Edi tText
```

```

    android: i d="@+i d/etCod"
    android: layout_wi dth="wrap_content"
    android: layout_hei ght="wrap_content"
    android: layout_al i gnLeft="@+i d/textVi ew1"
    android: layout_bel ow="@+i d/textVi ew1"
    android: ems="10"
    android: i nputType="number" >

```

```
<requestFocus />
```

```
</Edi tText>
```

```
<TextVi ew
```

```

    android: i d="@+i d/textVi ew2"
    android: layout_wi dth="wrap_content"
    android: layout_hei ght="wrap_content"
    android: layout_al i gnLeft="@+i d/etCod"
    android: layout_bel ow="@+i d/etCod"
    android: text="Qtd: " />

```

```
<Edi tText
```

```

    android: i d="@+i d/etQtd"
    android: layout_wi dth="wrap_content"
    android: layout_hei ght="wrap_content"
    android: layout_al i gnLeft="@+i d/textVi ew2"
    android: layout_bel ow="@+i d/textVi ew2"
    android: ems="10"
    android: i nputType="numberDeci mal " />

```

```
<TextVi ew
```

```

    android: i d="@+i d/textVi ew3"
    android: layout_wi dth="wrap_content"
    android: layout_hei ght="wrap_content"
    android: layout_al i gnLeft="@+i d/etQtd"
    android: layout_bel ow="@+i d/etQtd"
    android: text="Val or" />

```

```
<Edi tText
```

```

    android: i d="@+i d/etVal or"
    android: layout_wi dth="wrap_content"
    android: layout_hei ght="wrap_content"

```

Prof. Menezes	PROGRAMAÇÃO MOBILE Estudo de Caso	Data 23/08/2016
------------------	---	--------------------

```

android:layout_alignLeft="@+id/textView3"
android:layout_below="@+id/textView3"
android:ems="10"
android:inputType="numberDecimal" />

```

<Button

```

android:id="@+id/btConfirmar"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignLeft="@+id/etValor"
android:layout_below="@+id/etValor"
android:text="Confirmar" />

```

<Button

```

android:id="@+id/btListarProdutos"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignLeft="@+id/btConfirmar"
android:layout_below="@+id/btConfirmar"
android:text="Listar Produtos" />

```

</RelativeLayout>

Representando a tela abaixo:

