

File - D:\workspace\android\Programacao-Android\PM_20160328_trabalho_prova\app\src\main\AndroidManifest.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     package="br.com.wgbn.pm_20160328_trabalho_prova">
4
5     <application
6         android:name="com.activeandroid.app.Application"
7         android:allowBackup="true"
8         android:icon="@mipmap/ic_launcher"
9         android:label="@string/app_name"
10        android:supportsRtl="true"
11        android:theme="@style/AppTheme">
12            <activity android:name=".MainActivity" />
13            <activity
14                android:name=".SplashscreenActivity"
15                android:configChanges="orientation|keyboardHidden|
 screenSize"
16                android:label="@string/app_name"
17                android:theme="@style/SplashScreenTheme">
18                    <intent-filter>
19                        <action android:name="android.intent.action.MAIN" />
20
21                        <category android:name="android.intent.category.
LAUNCHER" />
22                    </intent-filter>
23                </activity>
24                <activity android:name=".CadastroActivity" />
25                <activity android:name=".BuscarActivity" />
26
27                <meta-data
28                    android:name="AA_DB_NAME"
29                    android:value="clientes.db" />
30                <meta-data
31                    android:name="AA_DB_VERSION"
32                    android:value="1" />
33                <meta-data
34                    android:name="AA_MODELS"
35                    android:value="br.com.wgbn.pm_20160328_trabalho_prova.
pojo.ClientePojo" />
36
37                <activity android:name=".ListarActivity"></activity>
38            </application>
39
40 </manifest>
41
```

File - D:\workspace\android\Programacao-Android\PM_20160328_trabalho_prova\app\src\main\res\menu\cadastro_menu.xml

```
1 <menu xmlns:android="http://schemas.android.com/apk/res/android"
2     xmlns:app="http://schemas.android.com/apk/res-auto"
3     xmlns:tools="http://schemas.android.com/tools"
4     tools:context="br.com.wgbn.pm_20160222_hello_world.MainActivity"
5     >
6
7     <item android:id="@+id	btnSalvar"
8         android:title="@string/btnSalvar"
9         android:orderInCategory="100"
10        android:icon="@drawable/ic_action_salvar"
11        app:showAsAction="always" />
12
13    <!--item android:id="@+id	btnCancelar"
14        android:title="@string/btnCancelar"
15        android:orderInCategory="100"
16        android:icon="@drawable/ic_action_cancelar"
17        app:showAsAction="always" /-->
18
19 </menu>
```

File - D:\workspace\android\Programacao-Android\PM_20160328_trabalho_prova\app\src\main\res\layout\item_cliente.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     android:orientation="vertical"
5     android:layout_margin="10dp"
6     android:layout_width="match_parent"
7     android:layout_height="wrap_content">
8
9     <TextView
10        android:text="undefined"
11        android:id="@+id/txtNomeCliente"
12        style="@style/nomeCliente"/>
13
14     <LinearLayout
15         android:orientation="horizontal"
16         android:layout_width="match_parent"
17         android:layout_height="wrap_content">
18
19         <TextView
20             android:text="undefined"
21             android:id="@+id/txtCpfCliente"
22             style="@style/dadosCliente"/>
23
24         <TextView
25             android:text="undefined"
26             android:id="@+id/txtTelefoneCliente"
27             android:textAlignment="textEnd"
28             style="@style/dadosCliente"/>
29     </LinearLayout>
30 </LinearLayout>
```

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/
  android"
3   xmlns:tools="http://schemas.android.com/tools"
4   android:layout_width="match_parent"
5   android:layout_height="match_parent"
6   android:orientation="vertical"
7   tools:context="br.com.wgbn.pm_20160328_trabalho_prova.
  MainActivity">
8
9   <ScrollView
10    android:layout_width="match_parent"
11    android:layout_height="match_parent"
12    android:fillViewport="false">
13
14     <LinearLayout
15       android:layout_width="match_parent"
16       android:layout_height="match_parent"
17       android:orientation="vertical"
18       android:paddingBottom="@dimen/activity_vertical_margin"
19       android:paddingLeft="@dimen/activity_horizontal_margin"
20       android:paddingRight="@dimen/activity_horizontal_margin"
21       android:paddingTop="@dimen/activity_vertical_margin">
22
23       <ImageView
24         android:layout_width="match_parent"
25         android:layout_height="wrap_content"
26         android:src="@drawable/ic_logo"
27         android:id="@+id/imgLogo" />
28
29       <LinearLayout
30         android:orientation="horizontal"
31         android:layout_width="match_parent"
32         android:layout_height="wrap_content"
33         android:layout_below="@+id/imgLogo"
34         android:id="@+id/barraMenu1">
35
36         <Button
37           android:text="@string/btnCadastrar"
38           android:id="@+id/btnCadastrar"
39           android:drawableTop="@drawable/ic_action_add"
40           style="@style/menuBtns"
41           android:onClick="cadastrarClick"/>
42
43         <Button
44           android:text="@string/btnBuscar"
45           android:id="@+id/btnBuscar"
46           android:drawableTop="@drawable/ic_action_busca"
47           style="@style/menuBtns"
48           android:onClick="buscarClick"/>
49
50     </LinearLayout>
51
52     <LinearLayout
53       android:orientation="horizontal"
```

File - D:\workspace\android\Programacao-Android\PM_20160328_trabalho_prova\app\src\main\res\layout\layout_activity_main

```
54         android:layout_width="match_parent"
55         android:layout_height="wrap_content"
56         android:layout_below="@+id/barraMenu1"
57         android:id="@+id/barraMenu2">
58
59             <Button
60                 android:text="@string/btnListar"
61                 android:id="@+id/btnListar"
62                 android:drawableTop="@drawable/ic_action_list"
63                 style="@style/menuBtns"
64                 android:onClick="listarClick"/>
65
66             <Button
67                 android:text="@string/btnSair"
68                 android:id="@+id/btnSair"
69                 android:drawableTop="@drawable/ic_action_sair"
70                 style="@style/menuBtns"
71                 android:onClick="sairClick"/>
72
73         </LinearLayout>
74
75     </LinearLayout>
76
77 </ScrollView>
78
79 </LinearLayout>
80
```

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     android:orientation="vertical"
8     tools:context="br.com.wgbn.pm_20160328_trabalho_prova.
9     BuscarActivity">
10
11    <ScrollView
12        android:layout_width="match_parent"
13        android:layout_height="match_parent">
14
15        <LinearLayout
16            android:orientation="vertical"
17            android:layout_width="match_parent"
18            android:layout_height="match_parent"
19            android:paddingBottom="@dimen/activity_vertical_margin"
20            android:paddingLeft="@dimen/activity_horizontal_margin"
21            android:paddingRight="@dimen/activity_horizontal_margin"
22            android:paddingTop="@dimen/activity_vertical_margin">
23
24            <TextView
25                style="@style/legendas"
26                android:text="@string/txtBuscarCpf" />
27
28            <AutoCompleteTextView
29                android:id="@+id/txtBuscarCpf"
30                android:inputType="number"
31                android:layout_marginBottom="34dp"
32                style="@style/campos" />
33
34            <TextView
35                android:id="@+id/lblNome"
36                android:textStyle="bold"
37                style="@style/cliente"/>
38
39            <TextView
40                android:id="@+id/lblTelefone"
41                style="@style/cliente"
42                android:onClick="telefoneClick"/>
43
44            <TextView
45                android:id="@+id/lblCpf"
46                style="@style/cliente"/>
47
48            <TextView
49                android:id="@+id/lblEndereco"
50                style="@style/cliente"/>
51
52            <TextView
53                android:id="@+id/lblSexo"
54                style="@style/cliente"/>
```

File - D:\workspace\android\Programacao-Android\PM_20160328_trabalho_prova\app\src\main\res\layout\layout_activity_busca.xml

```
55         <TextView  
56             android:id="@+id/lblAtivo"  
57             style="@style/cliente"/>  
58     </LinearLayout>  
59 </ScrollView>  
60  
61 </LinearLayout>  
62  
63 </LinearLayout>  
64
```

File - D:\workspace\android\Programacao-Android\PM_20160328_trabalho_prova\app\src\main\res\layout\layout_activity_listar.

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     android:orientation="vertical"
8     tools:context="br.com.wgbn.pm_20160328_trabalho_prova.
ListarActivity">
9
10    <android.support.v7.widget.RecyclerView
11        android:layout_height="match_parent"
12        android:layout_width="match_parent"
13        android:id="@+id/lstClientes"
14        android:divider="@color/colorPrimaryDark"
15        android:dividerHeight="1sp">
16
17    </android.support.v7.widget.RecyclerView>
18 </LinearLayout>
19
```

File - D:\workspace\android\Programacao-Android\PM_20160328_trabalho_prova\app\src\main\res\layout\layout_activity_cada:

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/
  android"
3   xmlns:tools="http://schemas.android.com/tools"
4   android:layout_width="match_parent"
5   android:layout_height="match_parent"
6   android:orientation="vertical"
7   tools:context="br.com.wgbn.pm_20160328_trabalho_prova.
  CadastroActivity">
8
9   <ScrollView
10    android:layout_width="match_parent"
11    android:layout_height="match_parent">
12
13     <LinearLayout
14       android:layout_width="match_parent"
15       android:layout_height="match_parent"
16       android:orientation="vertical"
17       android:paddingBottom="@dimen/activity_vertical_margin"
18       android:paddingLeft="@dimen/activity_horizontal_margin"
19       android:paddingRight="@dimen/activity_horizontal_margin"
20       android:paddingTop="@dimen/activity_vertical_margin">
21
22       <TextView
23         android:text="@string/txtNome"
24         style="@style/legendas"/>
25       <EditText
26         android:id="@+id/txtNome"
27         android:inputType="textCapSentences"
28         style="@style/campos"/>
29
30       <TextView
31         android:text="@string/txtCpf"
32         style="@style/legendas"/>
33       <br.com.wgbn.pm_20160328_trabalho_prova.componentes.
  WgbnMaskEditText
34         android:id="@+id/txtCpf"
35         android:inputType="number"
36         tipo="cpf"
37         style="@style/campos"/>
38
39       <TextView
40         android:text="@string/txtTelefone"
41         style="@style/legendas"/>
42       <br.com.wgbn.pm_20160328_trabalho_prova.componentes.
  WgbnMaskEditText
43         android:id="@+id/txtTelefone"
44         android:inputType="number"
45         tipo="telefone"
46         style="@style/campos"/>
47
48       <TextView
49         android:text="@string/txtEndereco"
50         style="@style/legendas"/>
51       <EditText
```

```
52         android:id="@+id/txtEndereco"
53         style="@style/campos"/>
54
55     <TextView
56         android:text="@string/txtSexo"
57         style="@style/legendas"/>
58     <LinearLayout
59         android:layout_width="match_parent"
60         android:layout_height="wrap_content"
61         android:layout_marginBottom="10dp">
62
63         <RadioGroup
64             android:layout_width="match_parent"
65             android:layout_height="wrap_content"
66             android:orientation="horizontal"
67             android:id="@+id/rdSexo">
68
69             <RadioButton
70                 android:layout_width="match_parent"
71                 android:layout_height="wrap_content"
72                 android:layout_weight="1"
73                 android:text="@string/rdSexoM"
74                 android:id="@+id/rdSexoM" />
75
76             <RadioButton
77                 android:layout_width="match_parent"
78                 android:layout_height="wrap_content"
79                 android:layout_weight="1"
80                 android:text="@string/rdSexoF"
81                 android:id="@+id/rdSexoF" />
82
83         </RadioGroup>
84
85     </LinearLayout>
86
87     <TextView
88         android:text="@string/txtAtivo"
89         style="@style/legendas"/>
90     <LinearLayout
91         android:layout_width="match_parent"
92         android:layout_height="wrap_content"
93         android:layout_marginBottom="10dp">
94
95         <Switch
96             android:layout_width="match_parent"
97             android:layout_height="wrap_content"
98             android:text="@string/onoffAtivo"
99             android:id="@+id/onoffAtivo" />
100
101     </LinearLayout>
102
103 </LinearLayout>
104
105 </ScrollView>
106
```

107 </LinearLayout>

108

```
1 <resources>
2
3     <!-- Declare custom theme attributes that allow changing which
4         styles are
5             used for button bars depending on the API level.
6             ?android:attr/buttonBarStyle is new as of API 11 so this is
7             necessary to support previous API Levels. -->
8     <declare-styleable name="ButtonBarContainerTheme">
9         <attr name="metaButtonBarButtonStyle" format="reference" />
10    </declare-styleable>
11
12 </resources>
13
```

File - D:\workspace\android\Programacao-Android\PM_20160328_trabalho_prova\app\src\main\res\values\colors.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <resources>
3     <color name="colorPrimary">#f44336</color>
4     <color name="colorPrimaryDark">#b71c1c</color>
5     <color name="colorAccent">#4caf50</color>
6
7     <color name="black_overlay">#66000000</color>
8     <color name="branco">#ffffff</color>
9     <color name="cinza">#666666</color>
10 </resources>
11
```

File - D:\workspace\android\Programacao-Android\PM_20160328_trabalho_prova\app\src\main\res\values\dimens.xml

```
1 <resources>
2     <!-- Default screen margins, per the Android Design guidelines.
3         --&gt;
4     &lt;dimen name="activity_horizontal_margin"&gt;16dp&lt;/dimen&gt;
5     &lt;dimen name="activity_vertical_margin"&gt;16dp&lt;/dimen&gt;
6 &lt;/resources&gt;</pre>
```

```
1 <resources>
2
3     <!-- Base application theme. -->
4     <style name="AppTheme" parent="Theme.AppCompat.Light.
5         DarkActionBar">
6         <!-- Customize your theme here. -->
7         <item name="colorPrimary">@color/colorPrimary</item>
8         <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
9         <item name="colorAccent">@color/colorAccent</item>
10    </style>
11
12   <style name="menuBtns">
13       <item name="android:layout_width">match_parent</item>
14       <item name="android:layout_height">wrap_content</item>
15       <item name="android:layout_weight">1</item>
16       <item name="android:background">@drawable/verdinho</item>
17       <item name="android:layout_margin">5dp</item>
18       <item name="android:paddingBottom">10dp</item>
19       <item name="android:paddingTop">10dp</item>
20       <item name="android:textColor">#ffffffff</item>
21   </style>
22
23   <style name="legendas">
24       <item name="android:layout_width">match_parent</item>
25       <item name="android:layout_height">wrap_content</item>
26       <item name="android:textStyle">italic</item>
27       <item name="android:textSize">18dp</item>
28       <item name="android:textColor">@color/colorAccent</item>
29   </style>
30
31   <style name="campos">
32       <item name="android:layout_width">match_parent</item>
33       <item name="android:layout_height">wrap_content</item>
34       <item name="android:layout_marginBottom">10dp</item>
35   </style>
36
37   <style name="nomeCliente">
38       <item name="android:layout_width">match_parent</item>
39       <item name="android:layout_height">wrap_content</item>
40       <item name="android:textSize">18dp</item>
41       <item name="android:textColor">@color/colorPrimary</item>
42   </style>
43
44   <style name="dadosCliente">
45       <item name="android:layout_width">match_parent</item>
46       <item name="android:layout_height">wrap_content</item>
47       <item name="android:layout_weight">1</item>
48       <item name="android:textSize">12dp</item>
49       <item name="android:textColor">@color/cinza</item>
50   </style>
51
52   <style name="cliente">
53       <item name="android:layout_width">match_parent</item>
54       <item name="android:layout_height">wrap_content</item>
55       <item name="android:textSize">24dp</item>
```

File - D:\workspace\android\Programacao-Android\PM_20160328_trabalho_prova\app\src\main\res\values\styles.xml

```
55      <item name="android:textColor">@color/colorPrimary</item>
56      <item name="android:textAlignment">center</item>
57      <item name="android:layout_marginBottom">15dp</item>
58  </style>
59
60  <style name="SplashScreenTheme" parent="Theme.AppCompat.
  NoActionBar">
61      <item name="android:windowBackground">@drawable/
  spalsh_background</item>
62  </style>
63
64 </resources>
65
```

```
1 <resources>
2     <string name="app_name">Web Programming</string>
3     <string name="btnCadastrar">Add Client</string>
4     <string name="btnBuscar">Search Client</string>
5     <string name="btnListar">List Clients</string>
6     <string name="btnSair">Exit</string>
7     <string name="txtNome">Client Name</string>
8     <string name="txtCpf">Client Document</string>
9     <string name="txtTelefone">Client Contact Phone</string>
10    <string name="txtEndereco">Client Address</string>
11    <string name="txtSexo">Client Gender</string>
12    <string name="rdSexoM">Male</string>
13    <string name="rdSexoF">Female</string>
14    <string name="txtAtivo">Client Status</string>
15    <string name="onoffAtivo">Active</string>
16    <string name="btnSalvar">Save</string>
17    <string name="btnCancelar">Cancel</string>
18    <string name="tituloCadastrar">Create New Client</string>
19    <string name="tituloListar">Client List</string>
20    <string name="tituloBuscar">Client Search</string>
21    <string name="snackDel">YES</string>
22    <string name="snackDeleteText">Delete</string>
23    <string name="txtBuscarCpf">Search by Document ID</string>
24 </resources>
25
```

File - D:\workspace\android\Programacao-Android\PM_20160328_trabalho_prova\app\src\main\res\drawable\spalsh_backgrou

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <layer-list xmlns:android="http://schemas.android.com/apk/res/android"
  ">
3     <item android:drawable="@color/branco"/>
4     <item>
5         <bitmap
6             android:gravity="center"
7             android:src="@drawable/ic_logo"/>
8     </item>
9 </layer-list>
```

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <resources>
3     <string name="app_name">Programação Web</string>
4     <string name="btnCadastrar">Cad. Cliente</string>
5     <string name="btnBuscar">Buscar Cliente</string>
6     <string name="btnListar">Listar Cliente</string>
7     <string name="btnSair">Sair</string>
8     <string name="txtNome">Nome do Cliente</string>
9     <string name="txtCpf">CPF do Cliente</string>
10    <string name="txtTelefone">Telefone do Cliente</string>
11    <string name="txtEndereco">Endereço do Cliente</string>
12    <string name="txtSexo">Sexo</string>
13    <string name="rdSexoM">Masculino</string>
14    <string name="rdSexoF">Feminino</string>
15    <string name="txtAtivo">Status do Cliente</string>
16    <string name="onoffAtivo">Ativo</string>
17    <string name="btnSalvar">Salvar</string>
18    <string name="btnCancelar">Cancelar</string>
19    <string name="tituloCadastrar">Cadastro de Cliente</string>
20    <string name="tituloListar">Clientes</string>
21    <string name="tituloBuscar">Buscar Clientes</string>
22    <string name="snackDel">SIM</string>
23    <string name="snackDeleteText">Excluir</string>
24    <string name="txtBuscarCpf">Buscar por CPF</string>
25 </resources>
```

```
1 <resources>
2     <!-- Example customization of dimensions originally defined in
3         res/values/dimens.xml
4         (such as screen margins) for screens with more than 820dp of
5         available width. This
6         would include 7" and 10" devices in Landscape (~960dp and ~
7         1280dp respectively). --&gt;
8     &lt;dimen name="activity_horizontal_margin"&gt;64dp&lt;/dimen&gt;
9 &lt;/resources&gt;</pre>
```

```
1 package br.com.wgbn.pm_20160328_trabalho_prova;
2
3 import android.content.Intent;
4 import android.support.v7.app.AppCompatActivity;
5 import android.os.Bundle;
6 import android.view.View;
7
8 public class MainActivity extends AppCompatActivity {
9
10    @Override
11    protected void onCreate(Bundle savedInstanceState) {
12        super.onCreate(savedInstanceState);
13        setContentView(R.layout.layout_activity_main);
14    }
15
16    public void cadastrarClick(View v) {
17        Intent cadastro = new Intent(this, CadastroActivity.class);
18        startActivityForResult(cadastro, 1);
19    }
20
21    public void buscarClick(View v){
22        Intent buscar = new Intent(this, BuscarActivity.class);
23        startActivityForResult(buscar, 2);
24    }
25
26    public void listarClick(View v){
27        Intent listar = new Intent(this, ListarActivity.class);
28        startActivityForResult(listar, 3);
29    }
30
31    public void sairClick(View v){
32        finish();
33    }
34 }
```

```
1 package br.com.wgbn.pm_20160328_trabalho_prova;
2
3 import android.content.Intent;
4 import android.net.Uri;
5 import android.support.v7.app.AppCompatActivity;
6 import android.os.Bundle;
7 import android.util.Log;
8 import android.view.MenuItem;
9 import android.view.View;
10 import android.widget.AdapterView;
11 import android.widget.AutoCompleteTextView;
12 import android.widget.TextView;
13
14 import java.util.ArrayList;
15 import java.util.Iterator;
16
17 import br.com.wgbn.pm_20160328_trabalho_prova.pojo.ClientePojo;
18 import br.com.wgbn.pm_20160328_trabalho_prova.utils.
CpfAutoCompleteAdapter;
19 import br.com.wgbn.pm_20160328_trabalho_prova.utils.Utils;
20
21 public class BuscarActivity extends AppCompatActivity {
22
23     private ArrayList<ClientePojo> clientes;
24     private ArrayList<String> cpfs;
25     private AutoCompleteTextView busca;
26     private ClientePojo selecionado;
27
28     private TextView lblNome;
29     private TextView lblTelefone;
30     private TextView lblCpf;
31     private TextView lblEndereco;
32     private TextView lblSexo;
33     private TextView lblAtivo;
34
35     @Override
36     protected void onCreate(Bundle savedInstanceState) {
37         super.onCreate(savedInstanceState);
38         setContentView(R.layout.layout_activity_buscar);
39
40         getSupportActionBar().setTitle(R.string.tituloBuscar);
41         getSupportActionBar().setDisplayHomeAsUpEnabled(true);
42
43         carregaViews();
44         busca.setOnItemSelectedListener(new AdapterView.
OnItemSelectedListener() {
45
46             @Override
47             public void onItemClick(AdapterView<?> parent, View view,
48             int pos, long id) {
49                 selecionado = ClientePojo.getByCpf(busca.getText().
toString());
50                 carregaCliente();
51             }
51         });
51     }
52 }
```

```
52
53     getClienteDaLista();
54
55     carregaListas();
56 }
57
58     private void voltaMain(){
59         Intent intent = new Intent(getApplicationContext(),
60         MainActivity.class);
61         setResult(2, intent);
62         finish();
63     }
64
65     private void carregaViews(){
66         lblNome = (TextView) findViewById(R.id.lblNome);
67         lblTelefone = (TextView) findViewById(R.id.lblTelefone);
68         lblCpf = (TextView) findViewById(R.id.lblCpf);
69         lblEndereco = (TextView) findViewById(R.id.lblEndereco);
70         lblSexo = (TextView) findViewById(R.id.lblSexo);
71         lblAtivo = (TextView) findViewById(R.id.lblAtivo);
72         busca = (AutoCompleteTextView) findViewById(R.id.
73             txtBuscarCpf);
74     }
75
76     private void carregaCliente(){
77         lblNome.setText(selecionado.getNome());
78         lblTelefone.setText(selecionado.getTelefone());
79         lblCpf.setText(selecionado.getCpf());
80         lblEndereco.setText(selecionado.getEndereco());
81         lblSexo.setText(selecionado.getSexo() == 'M' ? "Masculino":"
82             Feminino");
83         lblAtivo.setText(selecionado.isAtivo() ? "Cliente Ativo":"
84             Cliente Inativo");
85     }
86
87     private void carregaListas(){
88         cpfs = new ArrayList<String>();
89         clientes = (ArrayList<ClientePojo>) ClientePojo.getAll();
90
91         Iterator<ClientePojo> iterator = clientes.iterator();
92
93         while (iterator.hasNext()){
94             ClientePojo c = iterator.next();
95             cpfs.add(c.getCpf());
96         }
97
98         CpfAutoCompleteAdapter adapter = new CpfAutoCompleteAdapter(
99             this, android.R.layout.simple_dropdown_item_1line, cpfs);
100         busca.setAdapter(adapter);
101     }
102
103     private void getClienteDaLista(){
104         Intent intent = getIntent();
105
106         selecionado = intent.getParcelableExtra("cliente");
```

```
102
103     if (selecionado != null) {
104         carregaCliente();
105         busca.setText(selecionado.getCpf());
106     }
107 }
108
109 public void telefoneClick(View v){
110     Uri uri = Uri.parse("tel:" + selecionado.getTelefone());
111     uri.toString().replaceAll("[^0-9]+","");
112     Intent ligar = new Intent(Intent.ACTION_DIAL, uri);
113     startActivity(ligar);
114 }
115
116 @Override
117 public boolean onOptionsItemSelected(MenuItem item) {
118     int id = item.getItemId();
119
120     if (id == android.R.id.home){
121         voltaMain();
122         return true;
123     }
124
125     return super.onOptionsItemSelected(item);
126 }
127
128 @Override
129 public void onBackPressed() {
130     voltaMain();
131 }
132
133 @Override
134 public void onSaveInstanceState(Bundle savedInstanceState) {
135     if (selecionado != null)
136         savedInstanceState.putParcelable("cliente", selecionado)
137 ;
138
139     super.onSaveInstanceState(savedInstanceState);
140 }
141
142 public void onRestoreInstanceState(Bundle savedInstanceState) {
143     super.onRestoreInstanceState(savedInstanceState);
144     selecionado = savedInstanceState.getParcelable("cliente");
145     if (selecionado != null)
146         carregaCliente();
147 }
```

```
1 package br.com.wgbn.pm_20160328_trabalho_prova;
2
3 import android.content.Intent;
4 import android.support.v7.app.AppCompatActivity;
5 import android.os.Bundle;
6 import android.support.v7.widget.DefaultItemAnimator;
7 import android.support.v7.widget.LinearLayoutManager;
8 import android.support.v7.widget.RecyclerView;
9 import android.util.Log;
10 import android.view.MenuItem;
11
12 import java.util.ArrayList;
13 import java.util.Iterator;
14
15 import br.com.wgbn.pm_20160328_trabalho_prova.componentes.
16 DividerItemDecoration;
17 import br.com.wgbn.pm_20160328_trabalho_prova.pojo.ClientePojo;
18 import br.com.wgbn.pm_20160328_trabalho_prova.utils.ClientesAdapter;
19
20 public class ListarActivity extends AppCompatActivity {
21
22     private RecyclerView lista;
23     private ArrayList<ClientePojo> clientes;
24
25     @Override
26     protected void onCreate(Bundle savedInstanceState) {
27         super.onCreate(savedInstanceState);
28         setContentView(R.layout.layout_activity_listar);
29
30         getSupportActionBar().setTitle(R.string.tituloListar);
31         getSupportActionBar().setDisplayHomeAsUpEnabled(true);
32
33         lista = (RecyclerView) findViewById(R.id.lstClientes);
34         lista.addItemDecoration(new DividerItemDecoration(
35             getResources().getDrawable(R.drawable.abc_list_divider_mtrl_alpha)));
36         lista.setItemAnimator(new DefaultItemAnimator());
37         lista.setLayoutManager(new LinearLayoutManager(this));
38
39         carregaClientes();
40     }
41
42     private void voltaMain(){
43         Intent intent = new Intent(getApplicationContext(),
44             MainActivity.class);
45         setResult(3, intent);
46         finish();
47     }
48
49     private void carregaClientes(){
50         clientes = (ArrayList<ClientePojo>) ClientePojo.getAll();
51         lista.setAdapter(new ClientesAdapter(this, clientes));
52     }
53
54     @Override
55     public boolean onOptionsItemSelected(MenuItem item) {
```

File - D:\workspace\android\Programacao-Android\PM_20160328_trabalho_prova\app\src\main\java\br\com\wgbn\pm_20160328

```
53         int id = item.getItemId();
54
55         if (id == android.R.id.home){
56             voltaMain();
57             return true;
58         }
59
60         return super.onOptionsItemSelected(item);
61     }
62
63     @Override
64     public void onBackPressed() {
65         voltaMain();
66     }
67 }
68
```

```
1 package br.com.wgbn.pm_20160328_trabalho_prova;
2
3 import android.content.Intent;
4 import android.support.v7.app.AppCompatActivity;
5 import android.os.Bundle;
6 import android.util.Log;
7 import android.view.Menu;
8 import android.view.MenuItem;
9 import android.view.View;
10 import android.widget.EditText;
11 import android.widget.RadioButton;
12 import android.widget.RadioGroup;
13 import android.widget.Switch;
14 import android.widget.Toast;
15
16 import br.com.wgbn.pm_20160328_trabalho_prova.componentes.WgbnMaskEditText;
17 import br.com.wgbn.pm_20160328_trabalho_prova.pojo.ClientePojo;
18
19 public class CadastroActivity extends AppCompatActivity {
20
21     private EditText txtNome;
22     private WgbnMaskEditText txtCpf;
23     private WgbnMaskEditText txtTelefone;
24     private EditText txtEndereco;
25     private RadioGroup rdSexo;
26     private Switch onoffAtivo;
27
28     @Override
29     protected void onCreate(Bundle savedInstanceState) {
30         super.onCreate(savedInstanceState);
31         setContentView(R.layout.layout_activity_cadastro);
32
33         txtNome = (EditText)findViewById(R.id.txtNome);
34         txtCpf = (WgbnMaskEditText)findViewById(R.id.txtCpf);
35         txtTelefone = (WgbnMaskEditText)findViewById(R.id.txtTelefone);
36     };
37         txtEndereco = (EditText)findViewById(R.id.txtEndereco);
38         rdSexo = (RadioGroup)findViewById(R.id.rdSexo);
39         onoffAtivo = (Switch)findViewById(R.id.onoffAtivo);
40
41         getSupportActionBar().setTitle(R.string.tituloCadastrar);
42         getSupportActionBar().setDisplayHomeAsUpEnabled(true);
43     }
44
45     private boolean validarCampos(){
46         if (!txtNome.getText().toString().isEmpty() &&
47             !txtCpf.getText().toString().isEmpty() &&
48             !txtTelefone.getText().toString().isEmpty() &&
49             !txtEndereco.getText().toString().isEmpty() &&
50             rdSexo.getCheckedRadioButtonId() > -1
51         )
52             return true;
53         return false;
54     }
```

```
54
55     private void salvarCliente(){
56         if (validarCampos()){
57             char sexo = getResources().getResourceEntryName(rdSexo.
58                 getCheckedRadioButtonId()).equals("rdSexoM") ? 'M':'F';
59             ClientePojo cliente = new ClientePojo(onoffAtivo.
60                 isChecked(),
61                     txtCpf.getText().toString(),
62                     txtEndereco.getText().toString(),
63                     txtNome.getText().toString(),
64                     sexo,
65                     txtTelefone.getText().toString());
66
67             cliente.save();
68
69             if (cliente.getId() > 0) {
70                 Toast.makeText(this, "Cliente cadastrado com sucesso",
71                     Toast.LENGTH_LONG).show();
72                 voltaMain();
73             } else
74                 Toast.makeText(this, "Houve um erro ao salvar o
75 cliente.", Toast.LENGTH_LONG).show();
76             } else {
77                 Toast.makeText(this, "Por favor, preencha todos os
78 campos.", Toast.LENGTH_LONG).show();
79             }
80         }
81
82         private void voltaMain(){
83             Intent intent = new Intent(getApplicationContext(),
84             MainActivity.class);
85             setResult(1, intent);
86             finish();
87         }
88
89         @Override
90         public boolean onCreateOptionsMenu(Menu menu) {
91             getMenuInflater().inflate(R.menu.cadastro_menu, menu);
92             return true;
93         }
94
95         @Override
96         public boolean onOptionsItemSelected(MenuItem item) {
97             int id = item.getItemId();
98
99             if (id == R.id.btnSalvar) {
100                 salvarCliente();
101                 return true;
102             } else if (id == android.R.id.home){
103                 voltaMain();
104                 return true;
105             }
106
107             return super.onOptionsItemSelected(item);
108         }
109     }
110 }
```

```
103
104     @Override
105     public void onBackPressed() {
106         voltaMain();
107     }
108 }
109
```

File - D:\workspace\android\Programacao-Android\PM_20160328_trabalho_prova\app\src\main\java\br\com\wgbn\pm_20160328_trabalho_prova\SplashscreenActivity.java

```
1 package br.com.wgbn.pm_20160328_trabalho_prova;
2
3 import android.content.Intent;
4 import android.support.v7.app.AppCompatActivity;
5 import android.os.Bundle;
6
7 public class SplashscreenActivity extends AppCompatActivity {
8
9     @Override
10    protected void onCreate(Bundle savedInstanceState) {
11        super.onCreate(savedInstanceState);
12
13        Intent intent = new Intent(this, MainActivity.class);
14        startActivity(intent);
15        finish();
16    }
17 }
18 }
```

```
1 package br.com.wgbn.pm_20160328_trabalho_prova.pojo;
2
3 import android.os.Parcel;
4 import android.os.Parcelable;
5
6 import com.activeandroid.Model;
7 import com.activeandroid.annotation.Column;
8 import com.activeandroid.annotation.Table;
9 import com.activeandroid.query.Select;
10
11 import java.util.List;
12
13 @Table(name = "clientes")
14 public class ClientePojo extends Model implements Parcelable {
15
16     @Column(index = true)
17     private String cpf;
18
19     @Column
20     private String nome;
21
22     @Column
23     private String endereco;
24
25     @Column
26     private String telefone;
27
28     @Column
29     private char sexo;
30
31     @Column
32     private boolean ativo;
33
34     public ClientePojo(){
35         super();
36     }
37
38     public ClientePojo(boolean ativo, String cpf, String endereco,
39                         String nome, char sexo, String telefone) {
40         super();
41         this.ativo = ativo;
42         this.cpf = cpf;
43         this.endereco = endereco;
44         this.nome = nome;
45         this.sexo = sexo;
46         this.telefone = telefone;
47     }
48
49     public boolean isAtivo() {
50         return ativo;
51     }
52
53     public void setAtivo(boolean ativo) {
54         this.ativo = ativo;
55     }
56 }
```

```
55
56     public String getCpf() {
57         return cpf;
58     }
59
60     public void setCpf(String cpf) {
61         this.cpf = cpf;
62     }
63
64     public String getEndereco() {
65         return endereco;
66     }
67
68     public void setEndereco(String endereco) {
69         this.endereco = endereco;
70     }
71
72     public String getNome() {
73         return nome;
74     }
75
76     public void setNome(String nome) {
77         this.nome = nome;
78     }
79
80     public char getSexo() {
81         return sexo;
82     }
83
84     public void setSexo(char sexo) {
85         this.sexo = sexo;
86     }
87
88     public String getTelefone() {
89         return telefone;
90     }
91
92     public void setTelefone(String telefone) {
93         this.telefone = telefone;
94     }
95
96     @Override
97     public int describeContents() {
98         return 0;
99     }
100
101    @Override
102    public void writeToParcel(Parcel dest, int flags) {
103        dest.writeString(this.cpf);
104        dest.writeString(this.nome);
105        dest.writeString(this.endereco);
106        dest.writeString(this.telefone);
107        dest.writeInt(sexo);
108        dest.writeByte(ativo ? (byte) 1 : (byte) 0);
109    }
```

```
110
111     protected ClientePojo(Parcel in) {
112         this.cpf = in.readString();
113         this.nome = in.readString();
114         this.endereco = in.readString();
115         this.telefone = in.readString();
116         this.sexo = (char) in.readInt();
117         this.ativo = in.readByte() != 0;
118     }
119
120     public static final Parcelable.Creator<ClientePojo> CREATOR =
121         new Parcelable.Creator<ClientePojo>() {
122             public ClientePojo createFromParcel(Parcel source) {
123                 return new ClientePojo(source);
124             }
125
126             public ClientePojo[] newArray(int size) {
127                 return new ClientePojo[size];
128             }
129         };
130
131     public static List<ClientePojo> getAll() {
132         return new Select()
133             .from(ClientePojo.class)
134             .orderBy("nome ASC")
135             .execute();
136     }
137
138     public static ClientePojo getByCpf(String cpf) {
139         return (ClientePojo) new Select()
140             .from(ClientePojo.class)
141             .where("cpf = ?", cpf)
142             .executeSingle();
143     }
144 }
```

```

1 package br.com.wgbn.pm_20160328_trabalho_prova.utils;
2
3
4 import java.util.Set;
5 import java.util.regex.Matcher;
6 import java.util.regex.Pattern;
7
8 public abstract class Utils {
9
10    public static String[] REPLACES = {"a", "e", "i", "o", "u", "c"};
11    public static Pattern[] PATTERNS = null;
12
13    public static String unmask(String s, Set<String> replaceSymbols)
14    {
15        for (String symbol : replaceSymbols)
16            s = s.replaceAll("[+"+symbol+"]","");
17
18        return s;
19    }
20
21    public static String mask(String format, String text){
22        String maskedText="";
23        int i =0;
24        for (char m : format.toCharArray()) {
25            if (m != '#') {
26                maskedText += m;
27                continue;
28            }
29            try {
30                maskedText += text.charAt(i);
31            } catch (Exception e) {
32                break;
33            }
34            i++;
35        }
36        return maskedText;
37    }
38
39    public static void compilePatterns() {
40        PATTERNS = new Pattern[REPLACES.length];
41        PATTERNS[0] = Pattern.compile("[âãáàä]", Pattern.
CASE_INSENSITIVE);
42        PATTERNS[1] = Pattern.compile("[éèêë]", Pattern.
CASE_INSENSITIVE);
43        PATTERNS[2] = Pattern.compile("[íìîï]", Pattern.
CASE_INSENSITIVE);
44        PATTERNS[3] = Pattern.compile("[óòôõö]", Pattern.
CASE_INSENSITIVE);
45        PATTERNS[4] = Pattern.compile("[úùûü]", Pattern.
CASE_INSENSITIVE);
46        PATTERNS[5] = Pattern.compile("[ç]", Pattern.CASE_INSENSITIVE
);
47    }
48

```

```
49     public static String removeAcentos(String text) {  
50         if (PATTERNS == null) {  
51             compilePatterns();  
52         }  
53  
54         String result = text;  
55         for (int i = 0; i < PATTERNS.length; i++) {  
56             Matcher matcher = PATTERNS[i].matcher(result);  
57             result = matcher.replaceAll(REPLACES[i]);  
58         }  
59         return result.toUpperCase();  
60     }  
61 }
```

```
1 package br.com.wgbn.pm_20160328_trabalho_prova.utils;
2
3 import android.content.Context;
4 import android.content.Intent;
5 import android.graphics.Color;
6 import android.support.design.widget.Snackbar;
7 import android.view.LayoutInflater;
8 import android.view.View;
9 import android.view.ViewGroup;
10
11 import java.util.ArrayList;
12
13 import br.com.wgbn.pm_20160328_trabalho_prova.BuscarActivity;
14 import br.com.wgbn.pm_20160328_trabalho_prova.R;
15 import br.com.wgbn.pm_20160328_trabalho_prova.pojo.ClientePojo;
16
17 public class ClientesAdapter extends android.support.v7.widget.
    RecyclerView.Adapter<ClienteViewHolder> {
18
19     private Context context;
20     private ArrayList<ClientePojo> itens;
21     private ClientePojo selecionado;
22
23     public ClientesAdapter(Context context, ArrayList<ClientePojo>
    itens) {
24         this.context = context;
25         this.itens = itens;
26     }
27
28     @Override
29     public ClienteViewHolder onCreateViewHolder(ViewGroup parent, int
    viewType) {
30         View view = LayoutInflater.from(context).inflate(R.layout.
    item_cliente, parent, false);
31         ClienteViewHolder clienteViewHolder = new ClienteViewHolder(
    view, context);
32         return clienteViewHolder;
33     }
34
35     @Override
36     public void onBindViewHolder(final ClienteViewHolder holder,
    final int position) {
37         ClientePojo cliente = itens.get(position);
38         holder.txtNomeCliente.setText(cliente.getNome());
39         holder.txtCpfCliente.setText(String.valueOf(cliente.getCpf()));
40         holder.txtTelefoneCliente.setText(String.valueOf(cliente.
    getTelefone()));
41
42         holder.itemView.setOnLongClickListener(new View.
    OnLongClickListener() {
43             @Override
44             public boolean onLongClick(View v) {
45                 selecionado = itens.get(position);
46                 Snackbar snack = Snackbar.make(v, context.
```

```
File - D:\workspace\android\Programacao-Android\PM_20160328_trabalho_prova\app\src\main\java\br\com\wgbn\pm_20160328\AdapterListagem.java
46     getResources().getText(R.string.snackDeltext)+" "+selecionado.
        getNome()+"?", Snackbar.LENGTH_LONG)
47             .setAction(R.string.snackDel, new View.
        OnClickListener() {
48                 @Override
49                     public void onClick(View v) {
50                         selecionado.delete();
51                         itens.remove(position);
52                         notifyDataSetChanged();
53                     }
54                 })
55             .setActionTextColor(Color.RED);
56         snack.show();
57         return true;
58     }
59 });
60
61     holder.itemView.setOnClickListener(new View.OnClickListener(
62     ) {
63         @Override
64             public void onClick(View v) {
65                 selecionado = itens.get(position);
66                 Intent intent = new Intent(context, BuscarActivity.
67                     class);
68                 intent.putExtra("cliente", selecionado);
69                 holder.itemView.getContext().startActivity(intent);
70             }
71     }
72     @Override
73     public int getItemCount() {
74         return itens.size();
75     }
76 }
77 }
```

File - D:\workspace\android\Programacao-Android\PM_20160328_trabalho_prova\app\src\main\java\br\com\wgbn\pm_20160328_trabalho_prova\utils\ClienteViewHolder.java

```
1 package br.com.wgbn.pm_20160328_trabalho_prova.utils;
2
3 import android.content.Context;
4 import android.support.v7.widget.RecyclerView;
5 import android.view.View;
6 import android.widget.TextView;
7
8 import br.com.wgbn.pm_20160328_trabalho_prova.R;
9
10 public class ClienteViewHolder extends RecyclerView.ViewHolder {
11
12     private Context context;
13
14     public TextView txtNomeCliente;
15     public TextView txtCpfCliente;
16     public TextView txtTelefoneCliente;
17
18     public ClienteViewHolder(View itemView, final Context context) {
19         super(itemView);
20         this.context = context;
21         txtNomeCliente = (TextView) itemView.findViewById(R.id.
22             txtNomeCliente);
23         txtCpfCliente = (TextView) itemView.findViewById(R.id.
24             txtCpfCliente);
25         txtTelefoneCliente = (TextView) itemView.findViewById(R.id.
26             txtTelefoneCliente);
27     }
28 }
```

```
1 package br.com.wgbn.pm_20160328_trabalho_prova.utils;
2
3 import android.content.Context;
4 import android.widget.ArrayAdapter;
5 import android.widget.Filter;
6 import android.widget.Filterable;
7
8 import java.util.ArrayList;
9 import java.util.List;
10
11 public class Cpf.AutoCompleteAdapter extends ArrayAdapter<String>
12     implements Filterable {
13
14     private List<String> listaCompleta;
15     private List<String> resultados;
16     private Filter meuFiltro;
17
18     public Cpf.AutoCompleteAdapter(Context context, int layout, List<
19         String> cpfs) {
20         super(context, layout, cpfs);
21         this.listaCompleta = cpfs;
22         this.resultados = listaCompleta;
23         this.meuFiltro = new MeuFiltro();
24     }
25
26     @Override
27     public int getCount() {
28         return resultados.size();
29     }
30
31     @Override
32     public String getItem(int position) {
33         if (resultados != null && resultados.size() > 0 && position <
34             resultados.size()) {
35             return resultados.get(position);
36         } else {
37             return null;
38         }
39     }
40
41     public Filter getFilter() {
42         return meuFiltro;
43     }
44
45     private class MeuFiltro extends Filter {
46
47         @Override
48         protected FilterResults performFiltering(CharSequence constraint) {
49             FilterResults filterResults = new FilterResults();
50
51             if (constraint != null) {
52                 Page 1 of 2
```

```
File - D:\workspace\android\Programacao-Android\PM_20160328_trabalho_prova\app\src\main\java\br\com\wgbn\pm_20160328\src\main\java\br\com\wgbn\pm_20160328\FilterResults.java
52             String term = Utils.removeAcentos(constraint.
53                 toString().trim().toLowerCase());
54             String placeStr;
55             for (String p : listaCompleta) {
56                 placeStr = Utils.removeAcentos(p.toLowerCase());
57
58                 if (placeStr.indexOf(term) > -1) {
59                     temp.add(p);
60                 }
61             }
62         }
63         filterResults.values = temp;
64         filterResults.count = temp.size();
65         return filterResults;
66     }
67
68     @SuppressWarnings("unchecked")
69     @Override
70     protected void publishResults(CharSequence constraint,
71         FilterResults filterResults) {
72         resultados = (ArrayList<String>) filterResults.values;
73         notifyDataSetChanged();
74     }
75 }
76
```

```
1 package br.com.wgbn.pm_20160328_trabalho_prova.componentes;
2
3 import android.content.Context;
4 import android.util.AttributeSet;
5 import android.util.Log;
6 import android.widget.EditText;
7
8 import java.util.ArrayList;
9 import java.util.HashMap;
10
11 public class WgbnMaskEditText extends EditText {
12     private HashMap<String, String> tipos;
13     private AttributeSet atributos;
14     private MaskEditTextChangedListener mascaraListener;
15     private String mascara;
16
17     public WgbnMaskEditText(Context context) {
18         super(context);
19         init();
20     }
21
22     public WgbnMaskEditText(Context context, AttributeSet attrs) {
23         super(context, attrs);
24         this.atributos = attrs;
25         init();
26     }
27
28     public WgbnMaskEditText(Context context, AttributeSet attrs, int defStyleAttr) {
29         super(context, attrs, defStyleAttr);
30         this.atributos = attrs;
31         init();
32     }
33
34     public WgbnMaskEditText(Context context, AttributeSet attrs, int defStyleAttr, int defStyleRes) {
35         super(context, attrs, defStyleAttr, defStyleRes);
36         this.atributos = attrs;
37         init();
38     }
39
40     private void setMascara(String mascara) {
41         this.mascara = mascara;
42     }
43
44     private void init(){
45         carregaTipos();
46         carregaMascara();
47         mascaraListener = new MaskEditTextChangedListener(mascara,
48             this);
49         this.addTextChangedListener(mascaraListener);
50     }
51
52     private void carregaMascara(){
53         String _mascara = "###.###.###-##";
```

```
53         if (atributos.getAttributeCount() > 0){
54             for (int i = 0; i < atributos.getAttributeCount(); i++){
55                 if (atributos.getAttributeName(i).equals("mascara")){
56                     _mascara = atributos.getAttributeValue(i).
57                     toString();
58                     Log.v("wgbn", atributos.getAttributeValue(i).
59                     toString());
60                 }
61                 if (atributos.getAttributeName(i).equals("tipo")){
62                     _mascara = tipos.get(atributos.getAttributeValue
63                     (i).toString());
64                     Log.v("wgbn", tipos.get(atributos.
65                     getAttributeValue(i).toString()));
66                 }
67             }
68         private void carregaTipos(){
69             tipos = new HashMap<>();
70             tipos.put("telefone", "(##) #####-#####");
71             tipos.put("cpf", "###.###.###-##");
72             tipos.put("cnpj", "##.###.###/###-##");
73             tipos.put("creditcard", "####-####-####-####");
74             tipos.put("data", "##/##/####");
75             tipos.put("hora", "##:##:#");
76             tipos.put("datahora", "##/##/#### ##:##:#");
77             tipos.put("cep", "##.###-##");
78         }
79     }
80 }
```

```
1 package br.com.wgbn.pm_20160328_trabalho_prova.componentes;
2
3 import android.content.Context;
4 import android.content.res.TypedArray;
5 import android.graphics.Canvas;
6 import android.graphics.Rect;
7 import android.graphics.drawable.Drawable;
8 import android.support.v7.widget.LinearLayoutManager;
9 import android.support.v7.widget.RecyclerView;
10 import android.util.AttributeSet;
11 import android.view.View;
12
13 /*
14 * Copyright (C) 2014 The Android Open Source Project
15 *
16 * Licensed under the Apache License, Version 2.0 (the "License");
17 * you may not use this file except in compliance with the License.
18 * You may obtain a copy of the License at
19 *
20 *      http://www.apache.org/licenses/LICENSE-2.0
21 *
22 * Unless required by applicable law or agreed to in writing,
23 * software
24 * distributed under the License is distributed on an "AS IS" BASIS,
25 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
26 * implied.
27 * See the License for the specific language governing permissions
28 * and
29 * limitations under the License.
30 */
31
32 public class DividerItemDecoration extends RecyclerView.ItemDecoration {
33     private Drawable mDivider;
34
35     public DividerItemDecoration(Context context, AttributeSet attrs) {
36         final TypedArray a = context.obtainStyledAttributes(attrs,
37             new int[] { android.R.attr.listDivider });
38         mDivider = a.getDrawable(0);
39         a.recycle();
40     }
41
42     public DividerItemDecoration(Drawable divider) { mDivider =
43         divider; }
44
45     @Override
46     public void getItemOffsets (Rect outRect, View view, RecyclerView
47         parent, RecyclerView.State state) {
48         super.getItemOffsets(outRect, view, parent, state);
49         if (mDivider == null) return;
50         if (parent.getChildPosition(view) < 1) return;
51
52         if (getOrientation(parent) == LinearLayoutManager.VERTICAL)
53             outRect.top = mDivider.getIntrinsicHeight();
```

```
47         else outRect.left = mDivider.getIntrinsicWidth();
48     }
49
50     @Override
51     public void onDrawOver(Canvas c, RecyclerView parent) {
52         if (mDivider == null) { super.onDrawOver(c, parent); return;
53     }
54
55         if (getOrientation(parent) == LinearLayoutManager.VERTICAL)
56     {
57             final int left = parent.getPaddingLeft();
58             final int right = parent.getWidth() - parent.
59             getPaddingRight();
60             final int childCount = parent.getChildCount();
61
62             for (int i=1; i < childCount; i++) {
63                 final View child = parent.getChildAt(i);
64                 final RecyclerView.LayoutParams params = (
65                     RecyclerView.LayoutParams) child.getLayoutParams();
66                 final int size = mDivider.getIntrinsicHeight();
67                 final int top = child.getTop() - params.topMargin;
68                 final int bottom = top + size;
69                 mDivider.setBounds(left, top, right, bottom);
70                 mDivider.draw(c);
71             }
72         } else { //horizontal
73             final int top = parent.getPaddingTop();
74             final int bottom = parent.getHeight() - parent.
75             getPaddingBottom();
76             final int childCount = parent.getChildCount();
77
78             for (int i=1; i < childCount; i++) {
79                 final View child = parent.getChildAt(i);
80                 final RecyclerView.LayoutParams params = (
81                     RecyclerView.LayoutParams) child.getLayoutParams();
82                 final int size = mDivider.getIntrinsicWidth();
83                 final int left = child.getLeft() - params.leftMargin
84 ;
85                 final int right = left + size;
86                 mDivider.setBounds(left, top, right, bottom);
87                 mDivider.draw(c);
88             }
89         }
90     }

91     private int getOrientation(RecyclerView parent) {
92         if (parent.getLayoutManager() instanceof LinearLayoutManager
93 ) {
94             LinearLayoutManager layoutManager = (LinearLayoutManager
95 ) parent.getLayoutManager();
96             return layoutManager.getOrientation();
97         } else throw new IllegalStateException(
98             "DividerItemDecoration can only be used with a LinearLayoutManager.");
99     }

```

```
91  
92 }  
93
```

```
1 package br.com.wgbn.pm_20160328_trabalho_prova.componentes;
2
3 import android.text.Editable;
4 import android.text.TextWatcher;
5 import android.widget.EditText;
6 import java.util.HashSet;
7 import java.util.Set;
8 import br.com.wgbn.pm_20160328_trabalho_prova.utils.Utils;
9
10 public class MaskEditTextChangedListener implements TextWatcher{
11
12     private String mMask;
13     private EditText mEditText;
14     private Set<String> symbolMask = new HashSet<String>();
15     private boolean isUpdating;
16     private String old = "";
17
18     public MaskEditTextChangedListener(String mask, EditText editText
19 ) {
20         mMask = mask;
21         mEditText = editText;
22         initSymbolMask();
23     }
24
25     private void initSymbolMask(){
26         for (int i=0; i < mMask.length(); i++){
27             char ch = mMask.charAt(i);
28             if (ch != '#')
29                 symbolMask.add(String.valueOf(ch));
30         }
31     }
32
33     @Override
34     public void onTextChanged(CharSequence s, int start, int before,
35 int count) {
36         String str = Utils.unmask(s.toString(), symbolMask);
37         String mascara = "";
38
39         if (isUpdating) {
40             old = str;
41             isUpdating = false;
42             return;
43         }
44
45         if(str.length() > old.length())
46             mascara = Utils.mask(mMask, str);
47         else
48             mascara = s.toString();
49
50         isUpdating = true;
51
52         mEditText.setText(mascara);
53         mEditText.setSelection(mascara.length());
54     }
55 }
```

File - D:\workspace\android\Programacao-Android\PM_20160328_trabalho_prova\app\src\main\java\br\com\wgbn\pm_20160328

```
54     @Override
55     public void beforeTextChanged(CharSequence s, int start, int
56         count, int after) {
57     }
58
59
60     @Override
61     public void afterTextChanged(Editable s) {
62     }
63
64 }
```