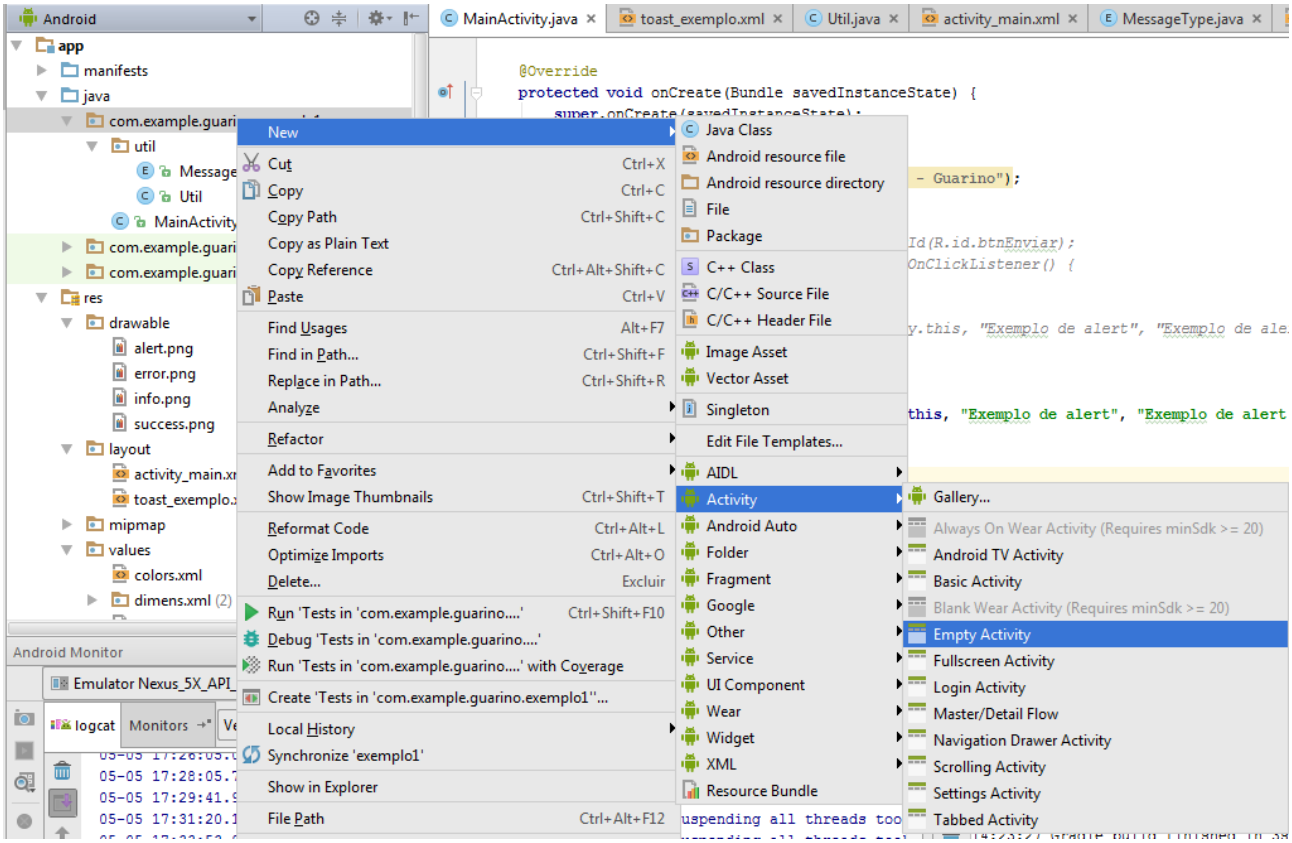
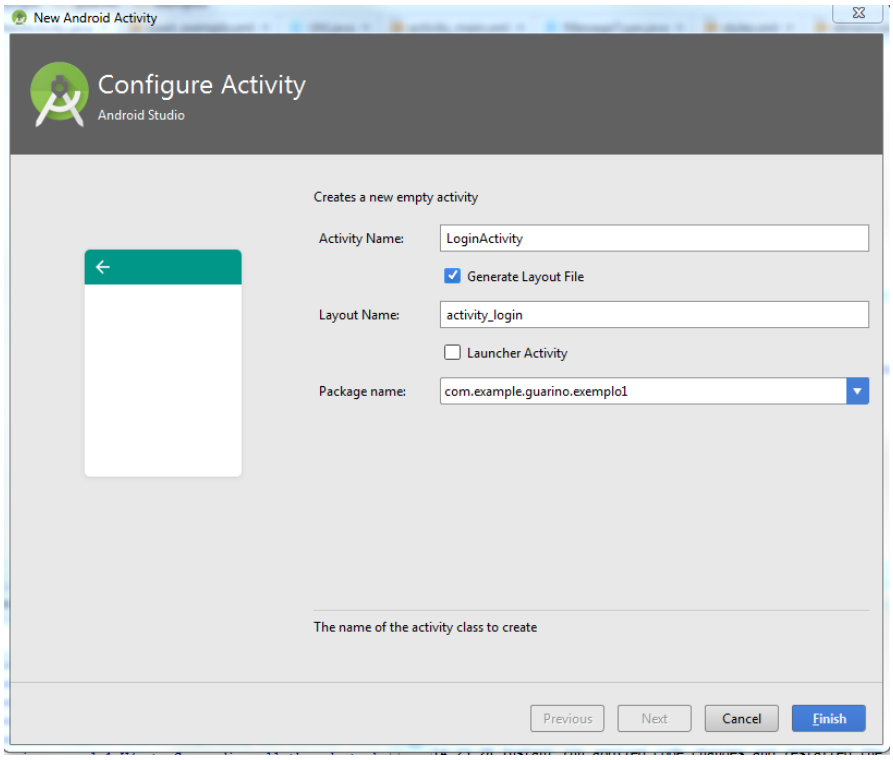
Login

Criar uma nova Activity



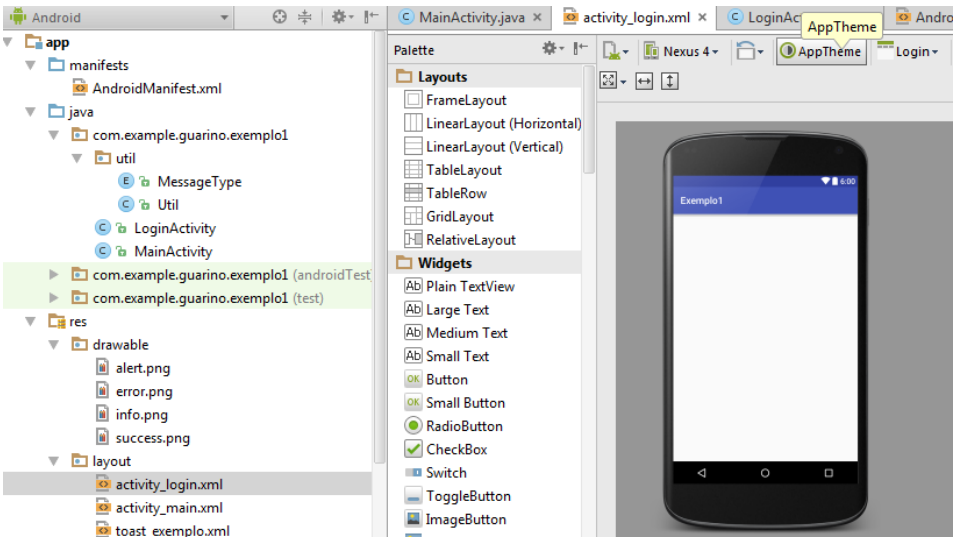


Observe no AndroidManifest.xml que foi criada a segunda Activity.

Nesse arquivo:
- defini-la como a primeira a ser executada (recortar o intent-filter da MainActivity para a LoginActivity)

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
xmlns:android="http://schemas.android.com/apk/res/android"
  package="com.example.guarino.exemplo1">
  <application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:supportsRtl="true"
    android:theme="@style/AppTheme">
    <activity android:name=".MainActivity">

    </activity>

    <activity android:name=".LoginActivity">

      <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
      </intent-filter>

    </activity>

  </application>
</manifest>
```

Abrir a activity_login.xml no modo Design e alterar o **AppTheme** para **AppCompat.Light**



Acrescentar as strings em strings.xml

```
<string name="lblLogin">Login</string>
<string name="lblSenha">Senha</string>
<string name="btnAutenticar">Autenticar</string>
```

Acrescentar cor em colors.xml

```
<color name="txt">#000000</color>
<color name="lbl">#000000</color>
```

Acrescentar dimensão em dimens.xml

```
<dimen name="txt">16sp</dimen>
<dimen name="lbl">16sp</dimen>
```

Criar um estilo em styles.xml

```
<style name="lbl" >
  <item name="android:textColor">@color/lbl</item>
  <item name="android:textSize">@dimen/lbl</item>
  <item
name="android:layout_marginBottom">@dimen/margin_widgets</item>
</style>

<style name="txt" >
  <item name="android:textColor">@color/txt</item>
  <item name="android:textSize">@dimen/txt</item>
  <item
name="android:layout_marginBottom">@dimen/margin_widgets</item>
</style>
```

Alterar a activity_login.xml

- Para LinearLayout
- Orientação: vertical
- Adicionar 2 TextView
- Adicionar 2 EditText
- Adicionar 1 Button

```
<?xml version="1.0" encoding="utf-8"?>

<LinearLayout
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:padding="@dimen/margin_widgets"
  tools:context="com.example.guarino.exemplo1.LoginActivity"
  android:orientation="vertical">
```

```
 <TextView
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
    style="@style/lbl"
  android:text="@string/lblLogin"/>

  <EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/txtLogin"
    style="@style/txt"
    android:inputType="text"
    />

  <TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    style="@style/lbl"
    android:text="@string/lblSenha"/>

  <EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/txtSenha"
    android:inputType="textPassword"
    />

  <Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/btnLogar"
    android:text="@string/btnAutenticar"
    />

</LinearLayout>
```

Na LoginActivity.java
- Deixar invisível a ActionBar

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

  getSupportActionBar().hide();

}
```

Executar o app
Será exibida a tela de Login.

Adicionar uma imagem acima do TextView em activity_login.xml
Iremos usar uma imagem já existente.

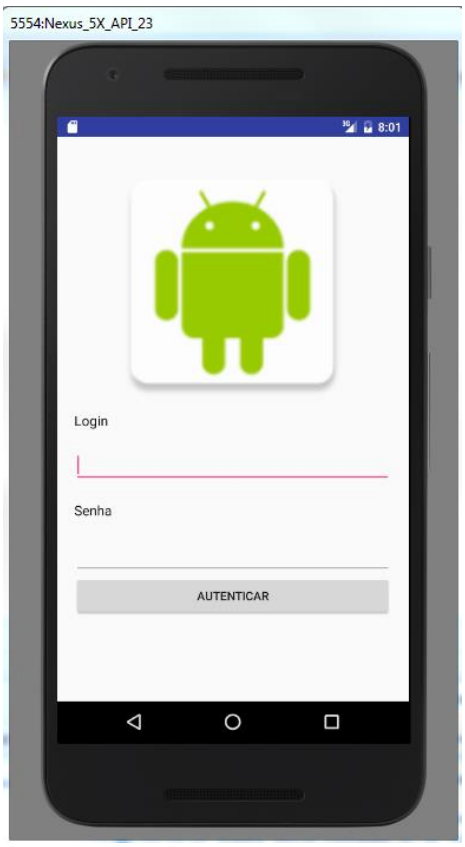
```
<ImageView
  android:layout_width="match_parent"
  android:layout_height="100dp"
  android:src="@mipmap/logo"
  android:layout_gravity="center"
  />
```

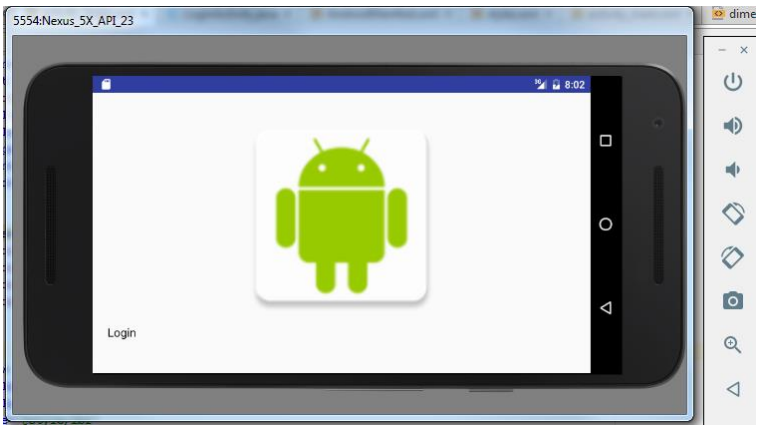
Executar o app

Será exibida a tela de Login.

Adicionando ScrollView
Dependendo da quantidade de componentes na tela, será necessário um ScrollView.

Altere a imagem da tela de login para ter `android:layout_height="300dp"`





Mesmo com a tela ruim, ainda não tem ScrollView.

Alterar a activity_login.xml

- O primeiro componente deve ser o ScrollView. Lembre de fechar a tag ScrollView ao final do arquivo.

- O primeiro elemento deve ter o xmlns

- Deve ser colocado também o layout_width e layout_height

- Do LinearLayout deve ser removido o xmlns

```
<?xml version="1.0" encoding="utf-8"?>

<ScrollView
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">

  <LinearLayout
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:padding="@dimen/margin_widgets"
  tools:context="com.example.guarino.exemplo1.LoginActivity"
  android:orientation="vertical">

  <ImageView
    android:layout_width="match_parent"
    android:layout_height="300dp"
    android:src="@mipmap/logo"
    android:layout_gravity="center"
    />
  <TextView
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
    style="@style/lbl"
```

```
  android:text="@string/lblLogin"/>
  <EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/txtLogin"
    style="@style/txt"
    android:inputType="text"
    />
  <TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    style="@style/lbl"
    android:text="@string/lblSenha"/>
  <EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/txtSenha"
    android:inputType="textPassword"
    />
  <Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/btnLogar"
    android:text="@string/btnAutenticar"
    />
</LinearLayout>
</ScrollView>
```

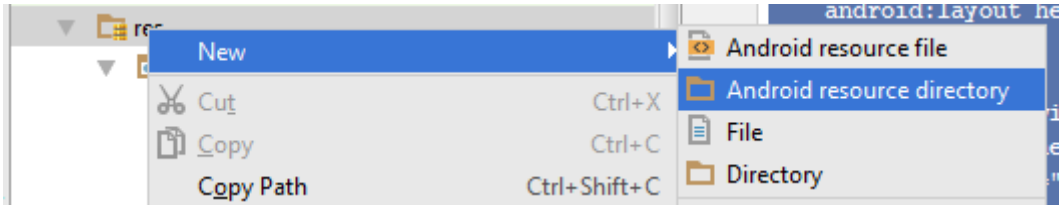
Executar o app

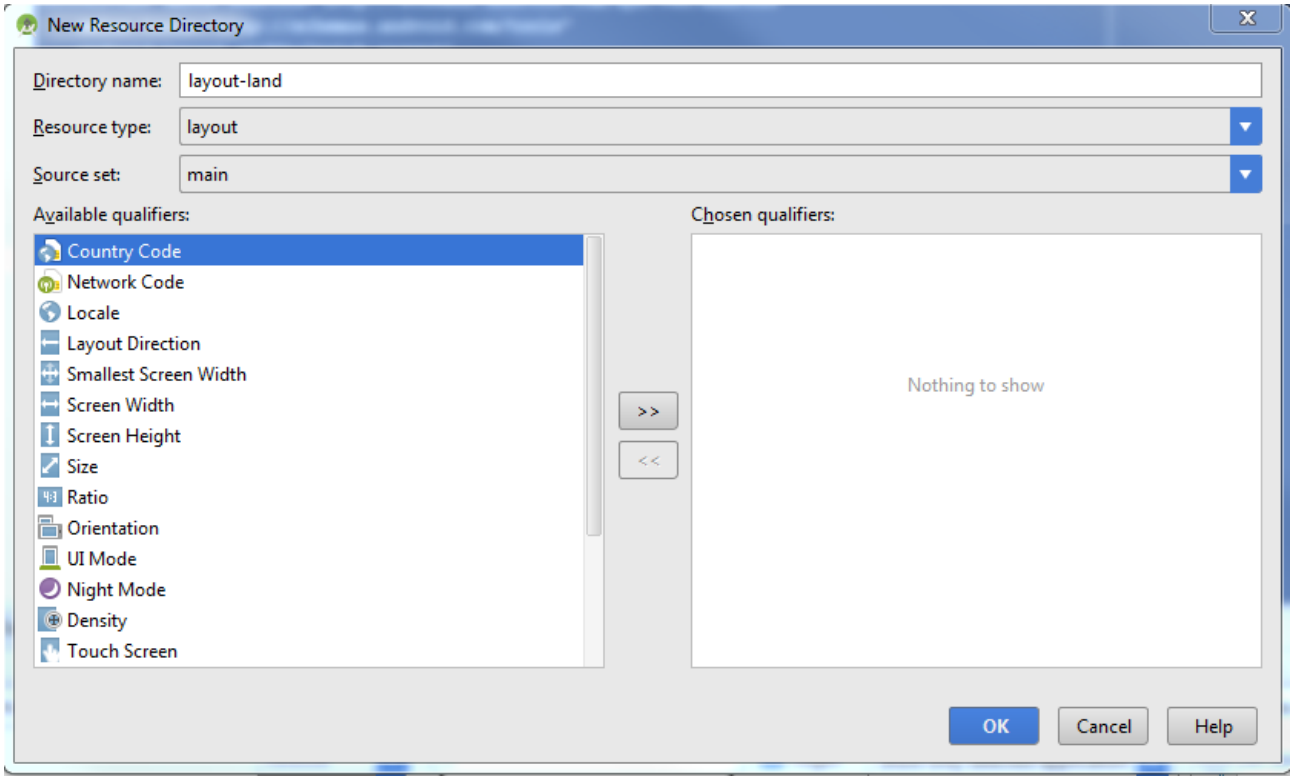
No layout Portrait está OK. Ao rotacionar a tela (Ctrl+F11) ou pelo emulador, no layout landscape aparece um Scroll.

Ainda não é a melhor solução. A melhor é ter um tela para Portrait e uma para Landscape.

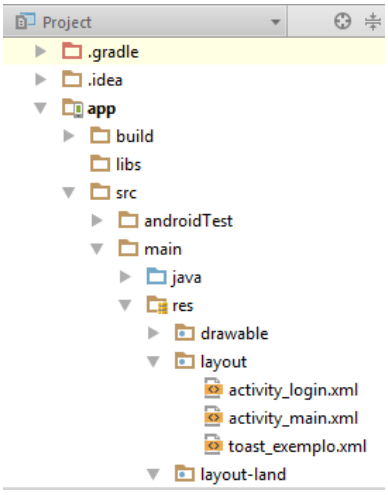
Adicionando outro diretório

Na pasta res > New > Android Resource Directory



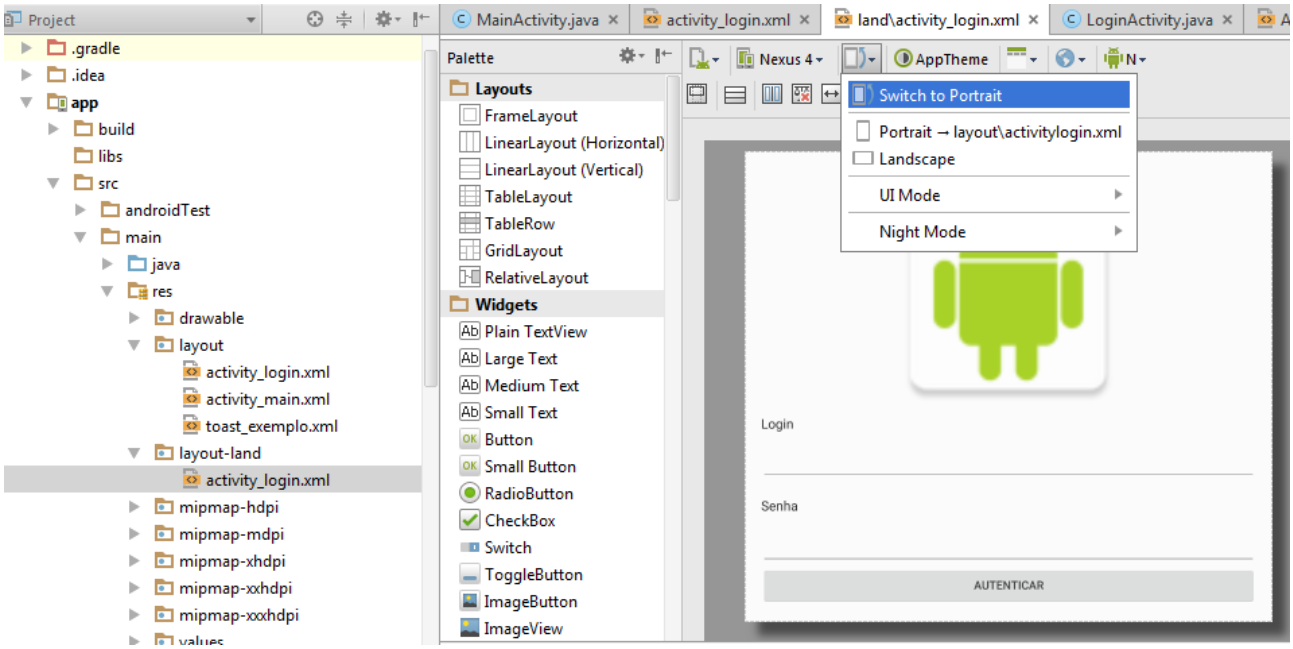


Selecione a visão Project



Na pasta layout, copie o arquivo activity_login.xml
Cole esse arquivo na pasta layout-land. O arquivo deve estar com o mesmo nome.

Automaticamente o Android Studio irá defini-la como Landspace (pois a outra Activity já está como Portrait)

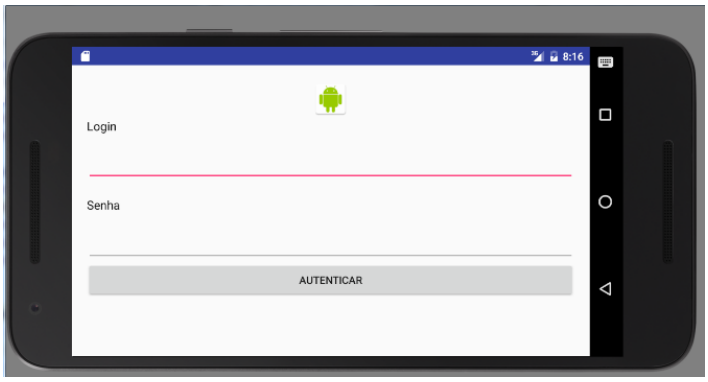


Na layout-land/activity_login.xml, altere a altura da imagem para apenas 50dp

```
<ImageView
  android:layout_width="match_parent"
  android:layout_height="50dp"
  android:src="@mipmap/logo"
  android:layout_gravity="center"
  />
```

Execute a app

No modo Portrait aparece corretamente.
Rotacione a tela para Landscape. A imagem será diminuída.



Obs: Para visualização portrait, o Android busca direto na pasta layout mas poderia ser a pasta layout-port.
É importante que os ids dos componentes estejam iguais porque se não na LoginActivity.java vai ocorrer uma exceção.

Validando os campos

Na LoginActivity.java

Acrescentar as linhas em destaque.

```
package com.example.guarino.exemplo1;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import com.example.guarino.exemplo1.util.Util;

public class LoginActivity extends AppCompatActivity {

  private EditText txtLogin;
  private EditText txtSenha;
  private Button btnLogar;

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    setContentView(R.layout.activity_login);
    getSupportActionBar().hide();

    txtLogin = (EditText) findViewById(R.id.txtLogin);
    txtSenha = (EditText) findViewById(R.id.txtSenha);
    btnLogar = (Button) findViewById(R.id.btnLogar);

    btnLogar.setOnClickListener(new View.OnClickListener() {

      @Override
      public void onClick(View v) {

        String login = txtLogin.getText().toString();
        String senha = txtSenha.getText().toString();
        boolean isValido = validarLogin(login, senha);

        if (!isValido)
        {
          Util.showMessageToast(LoginActivity.this, "Preencha os
campos");

        }
      }
    });

  }
```

```
  private boolean validarLogin(String login, String senha)
  {
    if (login == null || "".equals(login))
    {
      //Util.showMessageToast(LoginActivity.this, "Login
obrigatório");
      return false;
    }
    else
    {
      if (senha == null || "".equals(senha))
      {
        //Util.showMessageToast(LoginActivity.this, "Senha
obrigatória");
        return false;
      }
      else
      {
        return true;
      }
    }
  }
}
```

Execute o app

Verifique se a mensagem aparece corretamente para o não preenchimento dos campos.
Caso queira exibir uma mensagem diferente para cada campo, descomente as linhas comentadas no método validarLogin.

Uma outra forma de validar.
Acrescente uma mensagem de erro em cada campo. Dessa forma, não é necessário o uso do Toast.

```
private boolean validarLogin(String login, String senha)
{
  if (login == null || "".equals(login))
  {
    txtLogin.setError("Campo obrigatório");
    //Util.showMessageToast(LoginActivity.this, "Login
obrigatório");
    return false;
  }
  else
  {
    if (senha == null || "".equals(senha))
    {
      txtSenha.setError("Campo obrigatório");
      //Util.showMessageToast(LoginActivity.this, "Senha
obrigatória");
      return false;
    }
    else
    {
      return true;
    }
  }
}
```

Execute o app
Verifique se a mensagem aparece corretamente para o não preenchimento de cada campo.

Refatorando o código e validando os campos ao mesmo tempo

Crie o método isPreenchido.
Reescreva o método validarLogin.
Crie o método validarSenha.
Reescreva a chamada dos métodos.

```
public class LoginActivity extends AppCompatActivity {

  private EditText txtLogin;
  private EditText txtSenha;
  private Button btnLogar;

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_login);
    getSupportActionBar().hide();
    txtLogin = (EditText) findViewById(R.id.txtLogin);
    txtSenha = (EditText) findViewById(R.id.txtSenha);
    btnLogar = (Button) findViewById(R.id.btnLogar);

    btnLogar.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View v) {
        String login = txtLogin.getText().toString();
        String senha = txtSenha.getText().toString();

        boolean loginValido = validarLogin(login);
        boolean senhaValida = validarSenha(senha);

        if (loginValido && senhaValida)
        {
          //Util.showMessageToast(LoginActivity.this, "Preencha os
campos");
        }

      }
    });
  }

  private boolean validarLogin(String login)
  {
    if (!isPreenchido(login))
    {
      txtLogin.setError("Campo obrigatório");
      //Util.showMessageToast(LoginActivity.this, "Login
obrigatório");
      return false;
    }
    return true;
  }
```

```
  private boolean validarSenha(String senha)
  {
      if (!isPreenchido(senha))
      {
        txtSenha.setError("Campo obrigatório");
        //Util.showMessageToast(LoginActivity.this, "Senha
obrigatória");
        return false;
      }
      return true;
  }

  private boolean isPreenchido(String campo)
  {
    if (campo == null || "".equals(campo)) {
      return false;
    }
    return true;
  }

}
```

Execute o app
Verifique se o funcionamento continua correto.
Agora os 2 campos são validados no clique no botão.

Quantidade mínima de caracteres
Criar um novo método

```
private boolean hasTamanhoMinimo(String campo)
{
  if (campo.length() > 2) {
    return false;
  }
  return true;
}
```

Alterar o método validarLogin

```
private boolean validarLogin(String login)
{
  if (!isPreenchido(login))
  {
    txtLogin.setError("Campo obrigatório");
    //Util.showMessageToast(LoginActivity.this, "Login
obrigatório");
    return false;
  }
  else if (hasTamanhoMinimo(login)) {

    txtLogin.setError("Campo deve ter pelo menos 3 caracteres");
    return false;

  }
  return true;
}
```

Execute o app
Verifique se o funcionamento está correto.