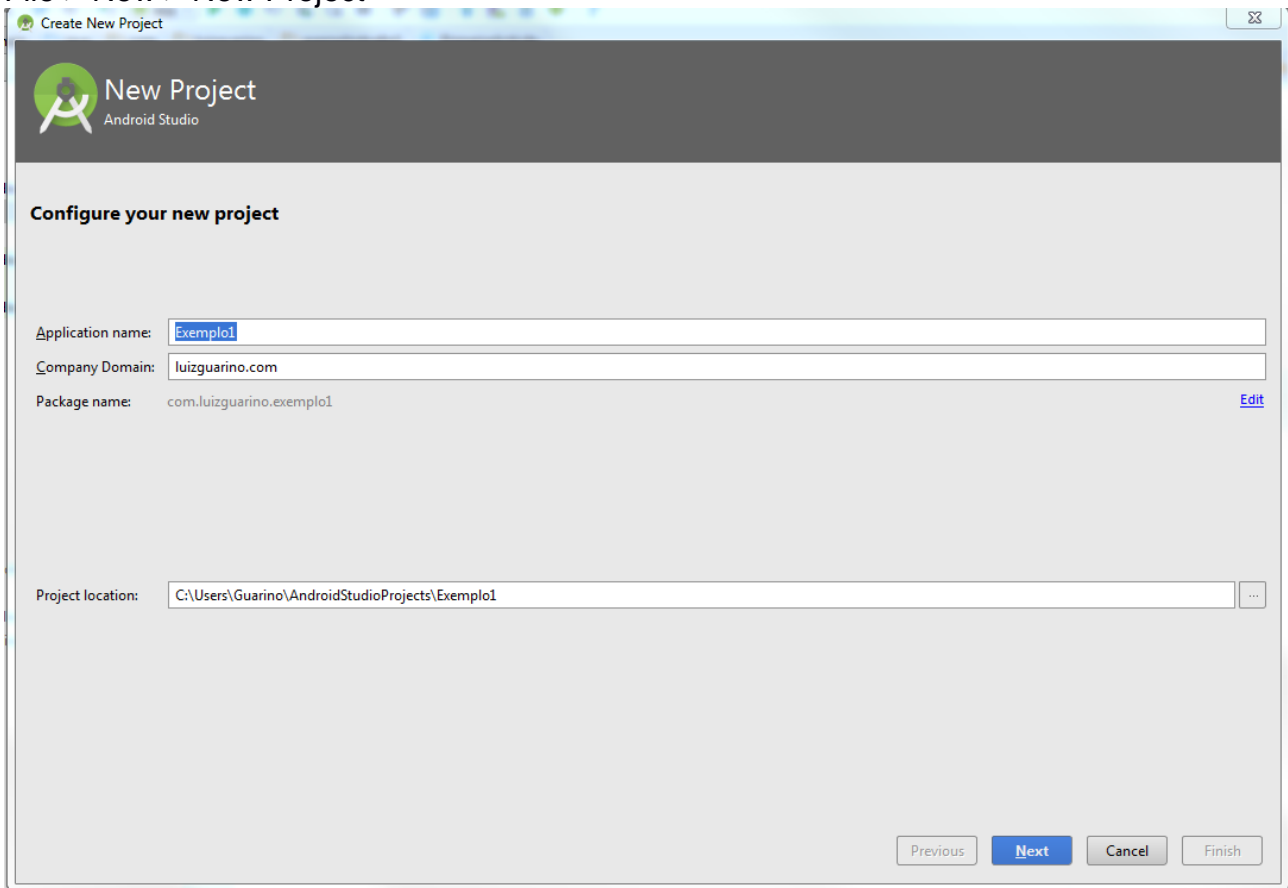


Trabalhando com Mensagens

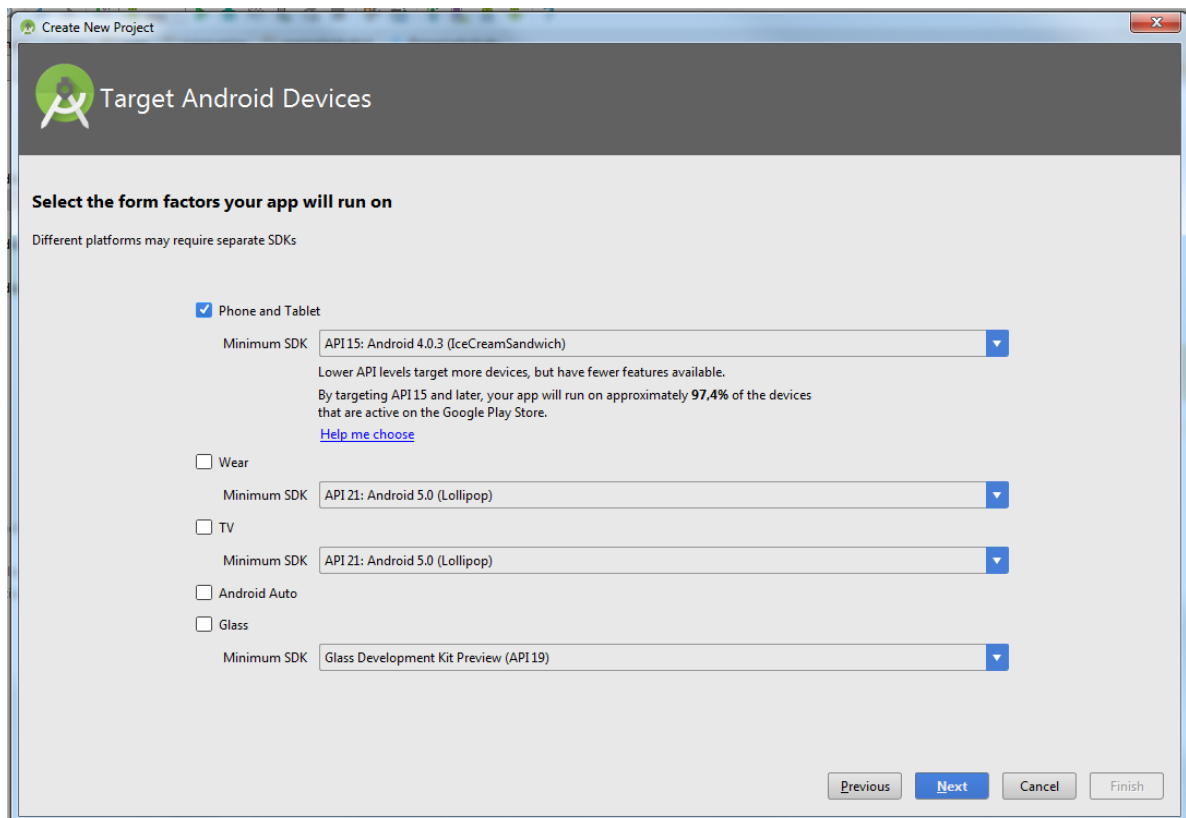
File > New > New Project



The screenshot shows the 'Create New Project' dialog in Android Studio. The title bar reads 'Create New Project'. The main header area contains the Android Studio logo and the text 'New Project Android Studio'. Below this, the heading 'Configure your new project' is displayed. The form includes the following fields:

- Application name:** A text input field containing 'Exemplo1'.
- Company Domain:** A text input field containing 'luizguarino.com'.
- Package name:** A text input field containing 'com.luizguarino.exemplo1'. To the right of this field is a blue 'Edit' link.
- Project location:** A text input field containing 'C:\Users\Guarino\AndroidStudioProjects\Exemplo1'. To the right of this field is a small square icon with three dots.

At the bottom right of the dialog, there are four buttons: 'Previous' (disabled), 'Next' (active), 'Cancel', and 'Finish' (disabled).

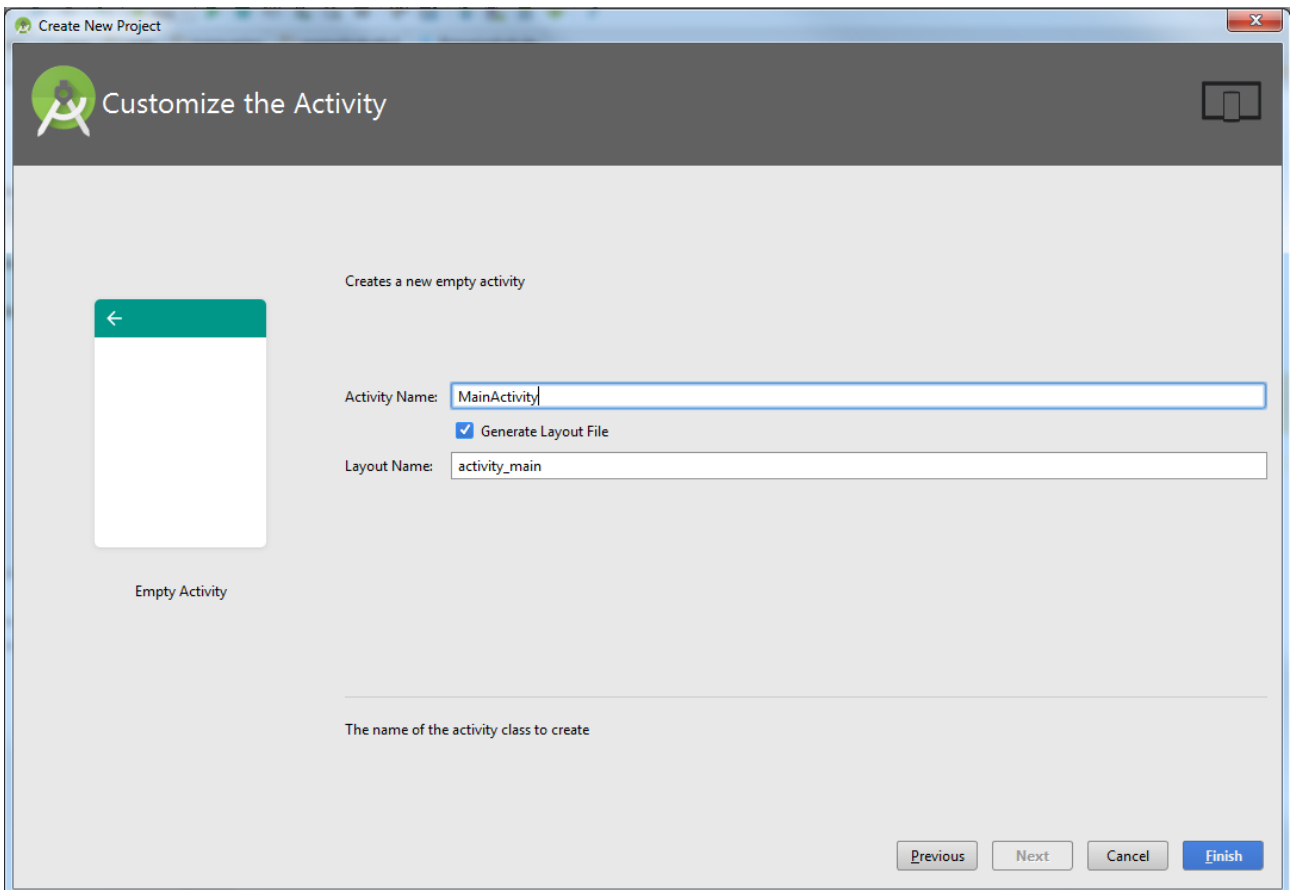
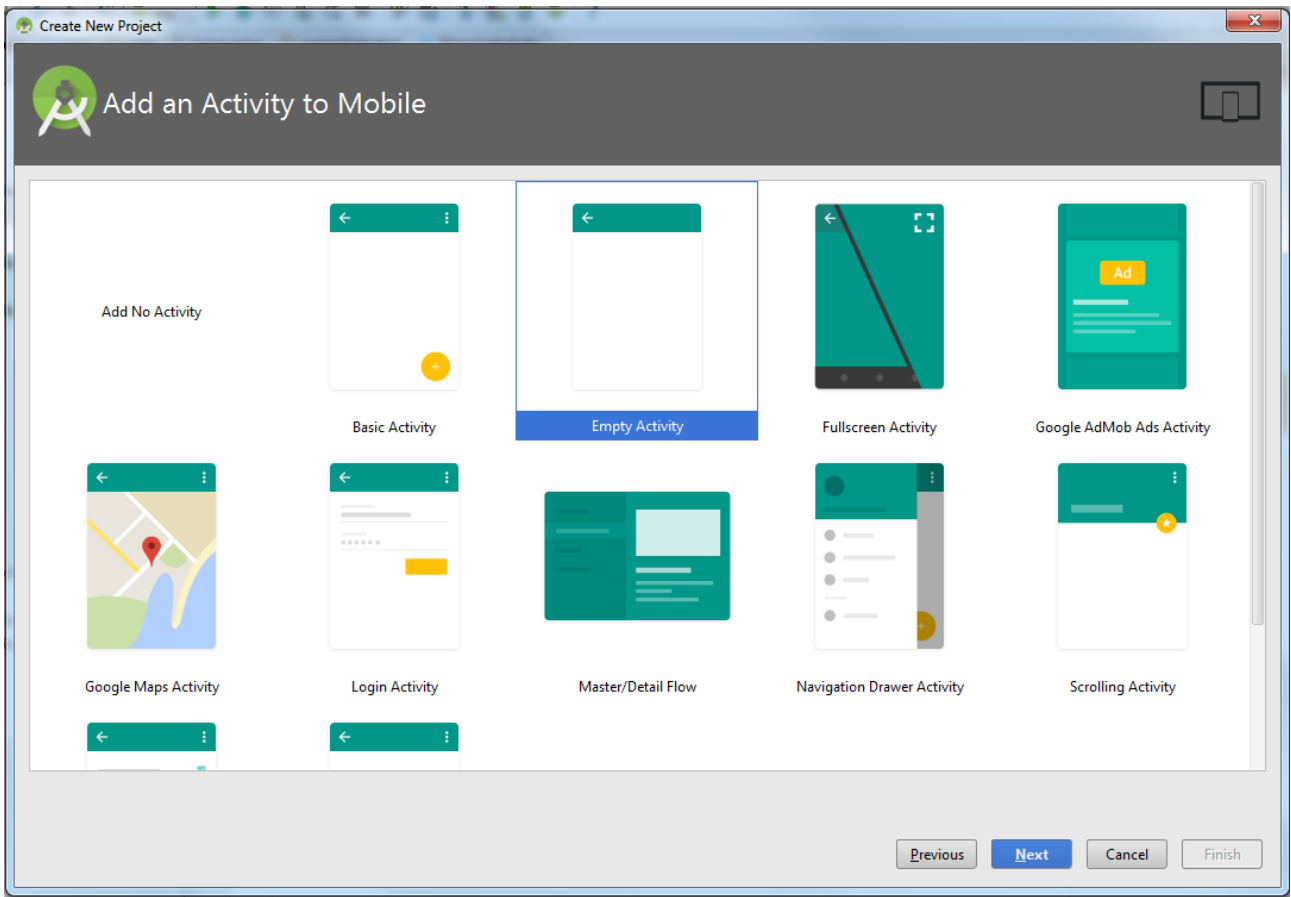


The screenshot shows the 'Target Android Devices' dialog in Android Studio. The title bar reads 'Create New Project'. The main header area contains the Android Studio logo and the text 'Target Android Devices'. Below this, the heading 'Select the form factors your app will run on' is displayed, followed by the sub-heading 'Different platforms may require separate SDKs'.

The form includes the following options:

- Phone and Tablet**
 - Minimum SDK: A dropdown menu showing 'API 15: Android 4.0.3 (IceCreamSandwich)'.
 - Lower API levels target more devices, but have fewer features available.
 - By targeting API 15 and later, your app will run on approximately **97.4%** of the devices that are active on the Google Play Store.
 - [Help me choose](#)
- Wear**
 - Minimum SDK: A dropdown menu showing 'API 21: Android 5.0 (Lollipop)'.
- TV**
 - Minimum SDK: A dropdown menu showing 'API 21: Android 5.0 (Lollipop)'.
- Android Auto**
- Glass**
 - Minimum SDK: A dropdown menu showing 'Glass Development Kit Preview (API 19)'.

At the bottom right of the dialog, there are four buttons: 'Previous' (disabled), 'Next' (active), 'Cancel', and 'Finish' (disabled).



Alterando o Título da ActionBar

Alterar strings.xml

```
<resources>
  <string name="app_name">Exemplo1</string>
  <string name="app_title">Exemplo1 - Guarino</string>
</resources>
```

Alterar MainActivity.java

A MainActivity deve herdar Activity ou AppCompatActivity (compatibilidade com versões anteriores)

```
public class MainActivity extends AppCompatActivity
```

- onCreate – executado assim que a tela abrir
- getSupportActionBar – por estarmos usando AppCompatActivity. Se estivéssemos herdando apenas a Activity, poderia usar getSupportActionBar.

```
package com.example.guarino.exemplo1;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
```

```
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        getSupportActionBar().setTitle(R.string.app_title);
    }
}
```

Executar a app no AVD

Exibindo uma mensagem

```
package com.example.guarino.exemplo1;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        getSupportActionBar().setTitle(R.string.app_title);

        //Mensagem rápida - Toda activity é um context
        Toast.makeText(MainActivity.this, "Teste de Exemplo app 1.0",
        Toast.LENGTH_SHORT).show();
    }
}
```

Duas opções:

- Toast.**LENGTH_SHORT**
- **Toast.LENGTH_LONG**

Executar a app no AVD – usando Toast.LENGTH_SHORT** e depois **Toast.LENGTH_LONG****

Será exibida uma mensagem na posição padrão do Toast – inferior da tela

Toast – o sistema controla a abertura e fechamento da mensagem. É uma mensagem simples suportada pelo Android.

Mais informações sobre Toast

<http://developer.android.com/intl/pt-br/guide/topics/ui/notifiers/toasts.html>

Customizando a mensagem

Pode customizar o Toast mas muitas vezes não compensa fazer isso pois, por exemplo, não dá pra dar a opção do usuário fechar a mensagem.

Posicionando a mensagem no centro

```
package com.example.guarino.exemplo1;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.Gravity;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        getSupportActionBar().setTitle(R.string.app_title);

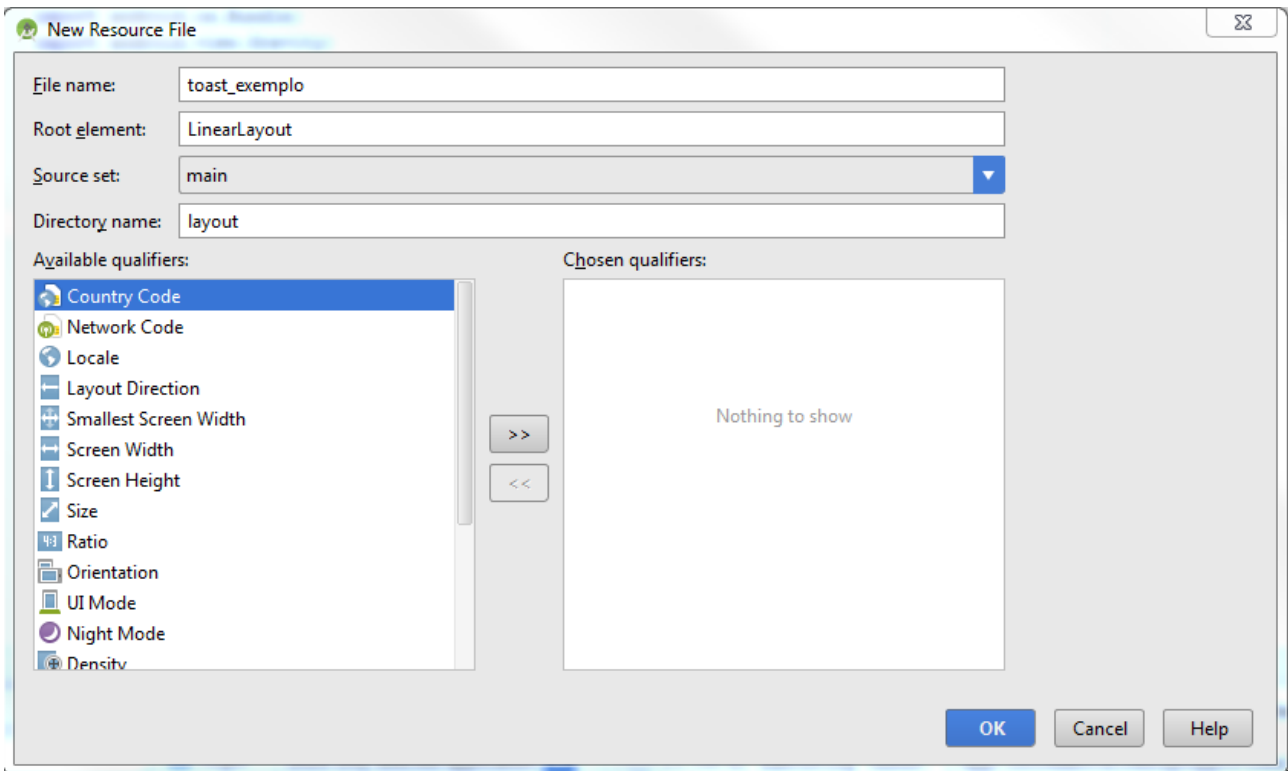
        //Mensagem rápida - Toda activity é um context
        Toast toast = Toast.makeText(MainActivity.this, "Teste de Exemplo app 1.0",
        Toast.LENGTH_SHORT);
        //set x e y - valores positivos ou negativos
        toast.setGravity(Gravity.CENTER, 0, 0);
        toast.show();
    }
}
```

Executar a app no AVD

Será exibida uma mensagem no centro da tela

Customizando a mensagem

Na pasta res/layout > New > Layout Resource File



Será criado o arquivo toast_exemplo.xml

Adicionar a TextView e id no Layout

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

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:id="@+id/layout_toast"
    android:layout_height="match_parent">
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/txtToast"/>
```

```
</LinearLayout>
```

MainActivity.java

```
package com.example.guarino.exemplo1;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.Gravity;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

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

        getSupportActionBar().setTitle(R.string.app_title);

        //infla o layout - permite adicionar informações ao layout
        LayoutInflater inflater = getLayoutInflater();
        View layout_toast = inflater.inflate(R.layout.toast_exemplo, (ViewGroup)
findViewById(R.id.layout_toast));
        TextView txtToast = (TextView) layout_toast.findViewById(R.id.txtToast);
        txtToast.setText("Exemplo de app 1.0");

        //Mensagem rápida - Toda activity é um context
        //Toast toast = Toast.makeText(MainActivity.this, "Teste de Exemplo app 1.0",
Toast.LENGTH_SHORT);
        Toast toast = new Toast(this);
        toast.setView(layout_toast);
        //set x e y - valores positivos ou negativos
        toast.setGravity(Gravity.CENTER, 0, 0);
        toast.show();
    }
}
```

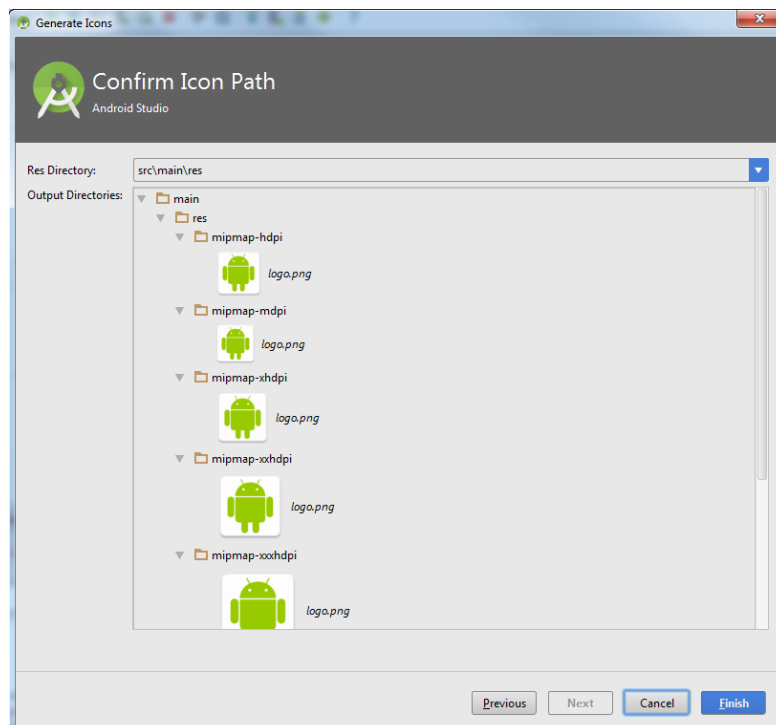
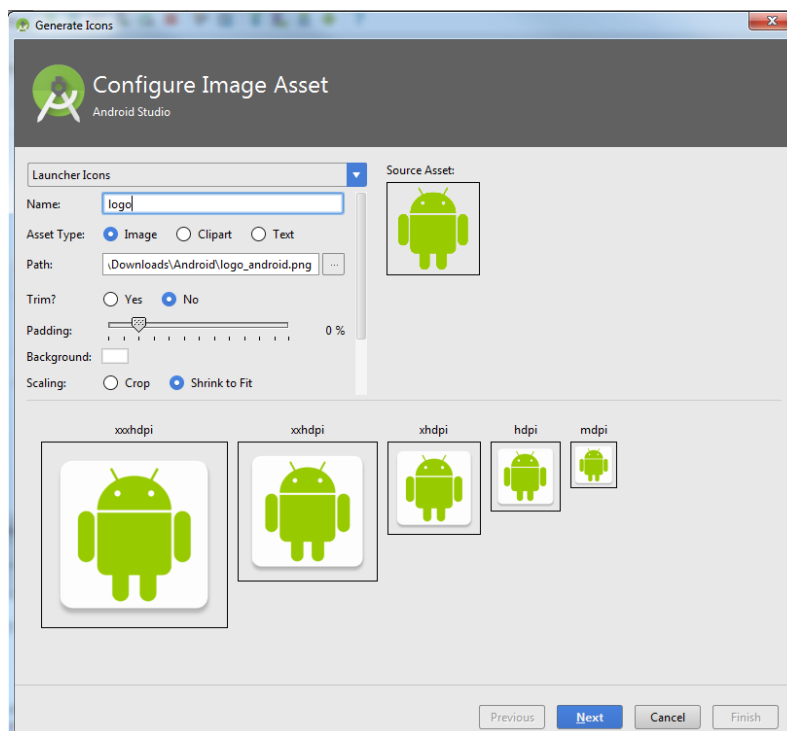
Executar a app no AVD

Será exibida uma mensagem no centro da tela.
Observe que a mensagem é sem estilo.

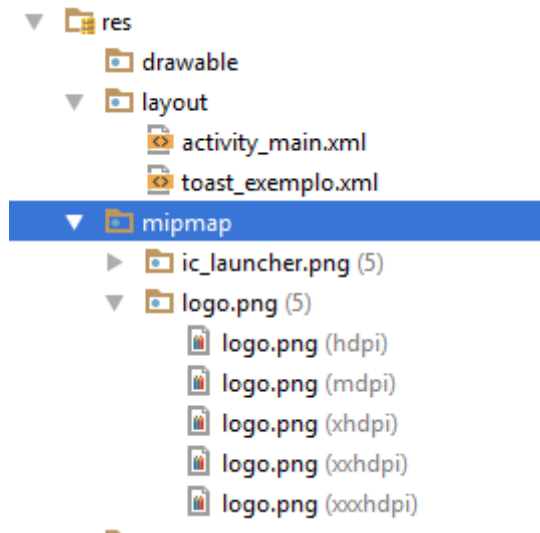
Adicionando uma imagem no Toast

Na pasta res/mipmap > New > Image Asset

Selecionar uma imagem do disco.



O projeto ficará assim:



Alterando o toast_exemplo.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:id="@+id/layout_toast"
    android:layout_height="match_parent">

    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@mipmap/logo"/>

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/txtToast"/>
</LinearLayout>
```

Executar a app no AVD

Será exibida uma mensagem no centro da tela com uma imagem.

Customizando as cores e espaçamentos do Toast

Adicionar cores em res/values/colors.xml

```
<?xml version="1.0" encoding="utf-8" ?>
<resources>
  <color name="colorPrimary">#3F51B5</color>
  <color name="colorPrimaryDark">#303F9F</color>
  <color name="colorAccent">#FF4081</color>
  <color name="branco">#FFFFFF</color>
  <color name="verdeEscuro">#004106</color>
</resources>
```

Alterar arquivos de dimensões res/values/dimens.xml

```
<resources>
  <!-- Default screen margins, per the Android Design guidelines. -->
  <dimen name="activity_horizontal_margin">16dp</dimen>
  <dimen name="activity_vertical_margin">16dp</dimen>
  <dimen name="margin_widgets">20dp</dimen>
</resources>
```

Alterar o toast_exemplo.xml

```
<?xml version="1.0" encoding="utf-8" ?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
  android:orientation="vertical" android:layout_width="match_parent"
  android:id="@+id/layout_toast"
  android:layout_height="match_parent"
  android:background="@color/verdeEscuro"
  android:padding="@dimen/margin_widgets">

  <ImageView
    android:layout_gravity="center"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:src="@mipmap/logo"/>

  <TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/txtToast"
    android:textColor="@color/branco"
  />
</LinearLayout>
```

Executar a app no AVD

Será exibida uma mensagem no centro da tela com uma imagem.