

A04 Conceitos Básicos de Android

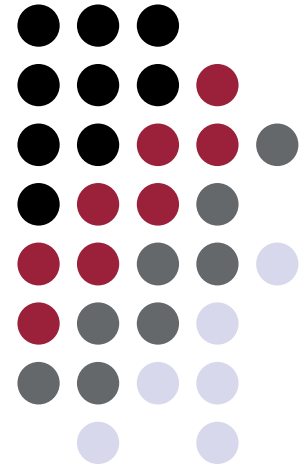


UFOP

Universidade Federal
de Ouro Preto

CSI401 – Programação para Dispositivos Móveis

Prof. Dr. George H. G. Fonseca
Universidade Federal de Ouro Preto





- Arquivo R.java
- Gerada automaticamente e contém identificadores para os recursos do projeto
 - Imagens
 - Arquivos XML de telas
 - Strings
- Pastas de recursos
 - drawable
 - layout
 - mipmap
 - values



- Quando você copia uma imagem para a pasta drawable um identificador para a imagem, com o nome do arquivo, é criado automaticamente na classe R
- Ids podem ser acessados no código por `R.pasta.id` e no XML por “@pasta/id”
 - Ex.: `R.drawable.smile` / “@drawable/smile”

AndroidManifest.xml



- Cada aplicação tem que ter um arquivo AndroidManifest.xml
- Apresenta informações sobre o aplicativo ao sistema Android
 - Activities
 - Permissions
 - Versão Android

```
<?xml version="1.0" encoding="utf-8" ?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="br.ufop.george.studentmanager">

    <uses-permission android:name="android.permission.CALL_PHONE" />

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="25"/>
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".StudentList" />
        <activity android:name=".StudentEdit" />
        <activity android:name=".StudentAdd" />
        <activity android:name=".StudentCall"></activity>
    </application>

</manifest>
```

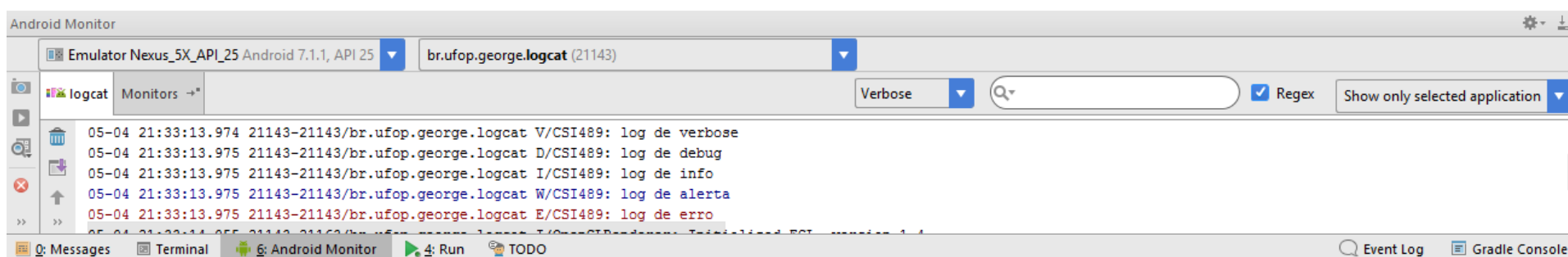


- Onde está a saída (antigo System.out.println())?
 - Usa a classe android.util.Log invece
- Exceptions (erros) também são informados aqui!!

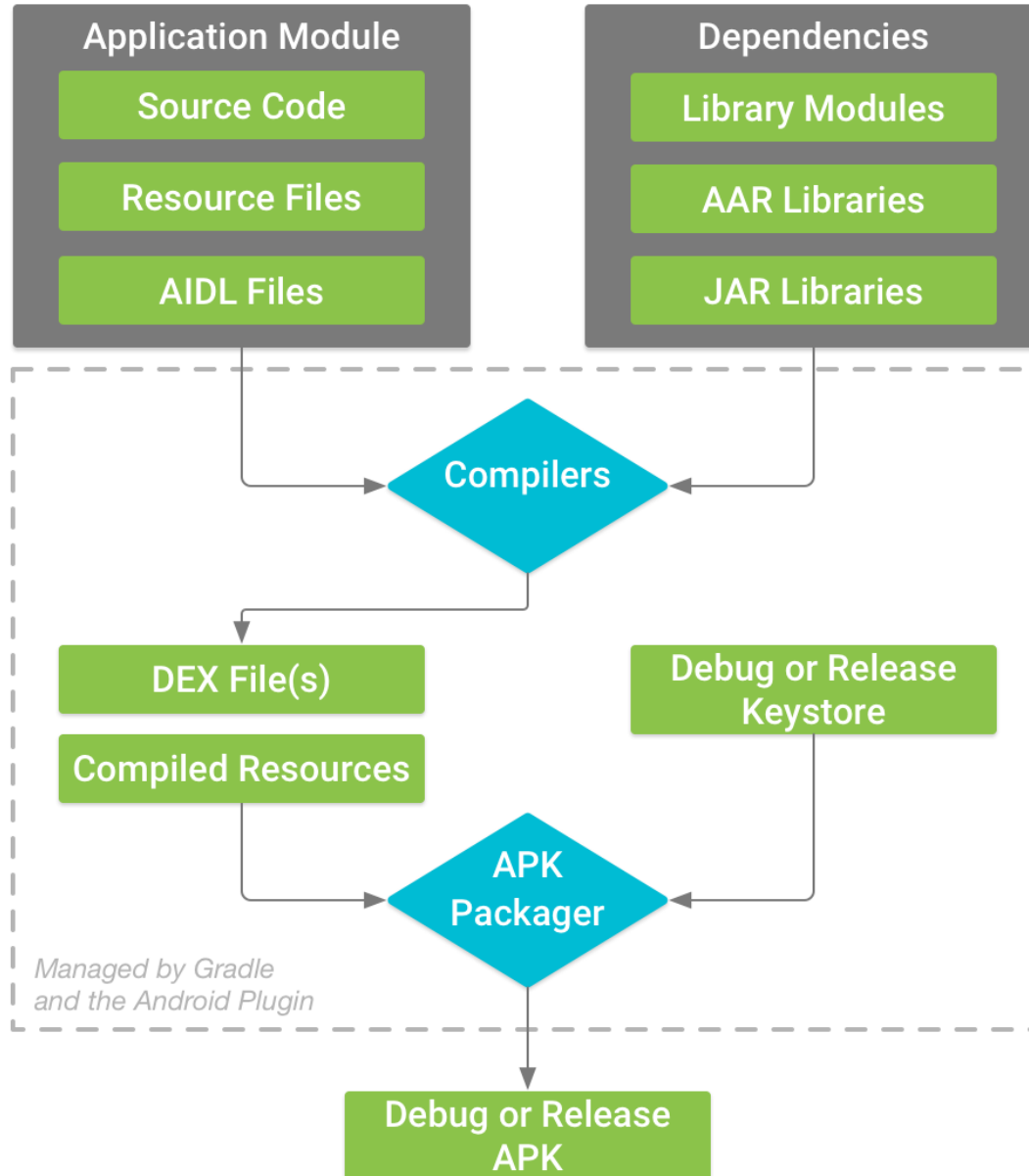
A screenshot of the Android Studio interface, specifically the Android Monitor window. The window title is "Android Monitor". The top bar shows the emulator name "Emulator Nexus_5X_API_25 Android 7.1.1, API 25" and the package name "br.ufop.george.aluno (4036)". Below this, there are tabs for "logcat" and "Monitors ->". The "logcat" tab is active, showing a list of log messages. The messages include timestamps, process IDs, package names, and log levels (I for info, W for warning). The messages are: "05-04 21:13:26.917 2338-2701/com.google.android.gms I/Icing: Usage reports 0 indexed 0 rejected 0 imm upload true", "05-04 21:13:26.935 2244-4069/com.google.android.googlequicksearchbox:search W/LanguagePackUpdateContr: Querying download manager failed for ID : 3", "05-04 21:13:26.939 1296-1296/? W/SurfaceFlinger: couldn't log to binary event log: overflow.", "05-04 21:13:26.939 1296-1296/? W/SurfaceFlinger: couldn't log to binary event log: overflow.", "05-04 21:13:26.939 2338-2701/com.google.android.gms I/Icing: Usage reports 0 indexed 0 rejected 0 imm upload true", and "05-04 21:13:26.964 2244-4069/com.google.android.googlequicksearchbox:search W/LanguagePackUpdateContr: Querying download manager failed for ID : 4". The bottom of the window shows a toolbar with icons for "Run", "TODO", "Android Monitor", "Messages", and "Terminal". On the right side of the toolbar, there are indicators for "Event Log" and "Gradle Console".



```
public class MainActivity extends Activity {  
  
    private static final String CATEGORY = "CSI489";  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        //Different types of log (check on LogCat)  
        Log.v(CATEGORY, "verbose log");  
        Log.d(CATEGORY, "debug log");  
        Log.i(CATEGORY, "info log");  
        Log.w(CATEGORY, "warning log");  
        Log.e(CATEGORY, "error log");  
    }  
}
```



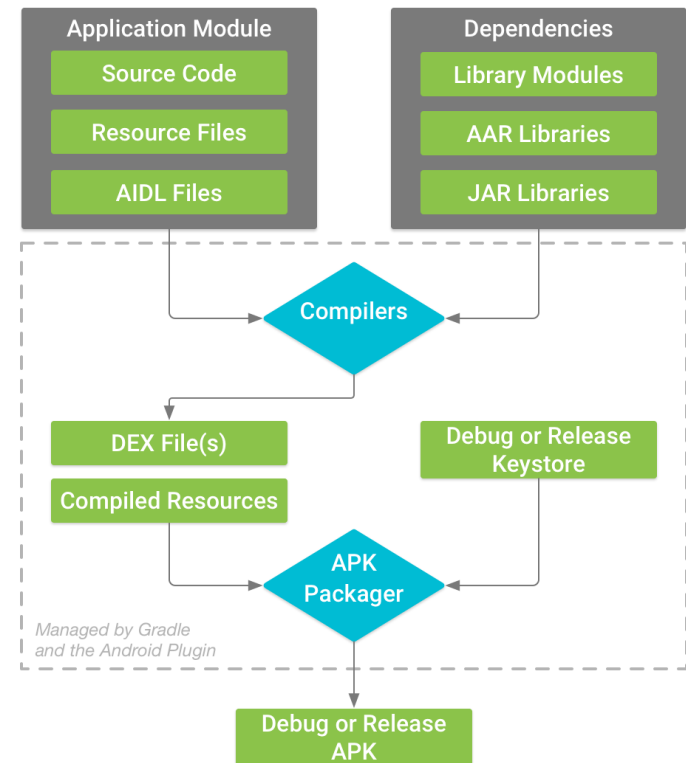
Processo de Compilação



Processo de Compilação



1. Os compiladores convertem o código-fonte em arquivos DEX (Dalvik Executable), que incluem o bytecode que é executado em dispositivos android e nos recursos
2. O APK Packager combina os arquivos DEX e os recursos compilados em um só APK.



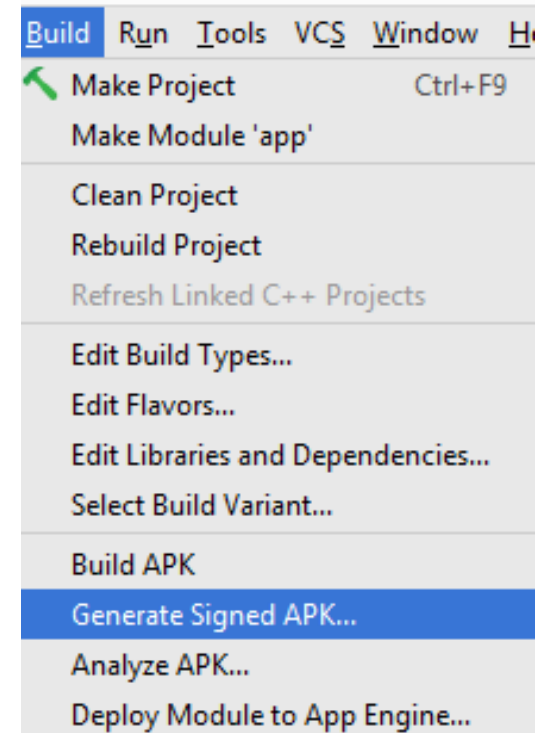
Processo de Compilação



3. O APK Packager assina o APK usando o repositório de chaves de lançamento ou de depuração:
 - a. Depuração: aplicativo apenas para teste, o Packager o assina com o repositório de chaves de depuração.

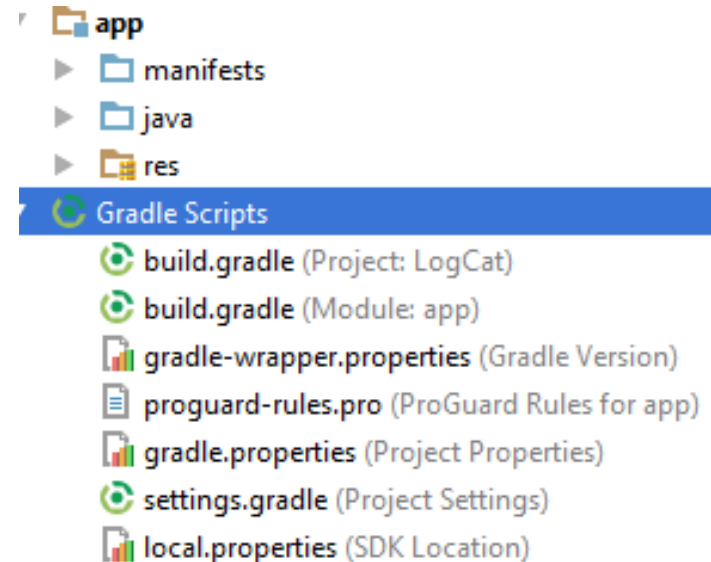
- b. Lançamento: o Packager o assina usando o repositório de chaves de lançamento.

[Como assinar seu aplicativo no Android Studio.](#)





- Kit avançado de ferramentas de compilação
 - Permite configurações personalizadas e flexíveis
 - Lida com a compatibilidade das várias versões níveis APK
 - Configurações personalizadas sem modificar os principais recursos do aplicativo
 - Android studio cria automaticamente arquivos .gradle





- Activity
 - Manipulação e tratamento das telas e eventos
- View
 - Elementos gráficos da aplicação
 - Layouts
 - Textos
 - Botões ...
- Intent
 - Intenção da aplicação
 - Abrir outra tela
 - Abrir outra aplicação
 - Passagem de parâmetros



- Activity
 - Manipulação e tratamento das telas e eventos
 - View
 - Elementos gráficos da aplicação
 - Layouts
 - Textos
 - Botões ...
 - Intent
 - Intenção da aplicação
 - Abrir outra tela
 - Abrir outra aplicação
 - Passagem de parâmetros
- Diagram illustrating relationships between Activity, View, and Intent:
- Activity calls `findViewById(int id)` to interact with View.
 - Activity calls `startActivity(Intent it)` and `startActivityForResult(Intent it, int code)` to interact with Intent.
 - Activity calls `putExtra(String id, Object data)` to pass data to Intent.

Interação entre Activity e View



activity_main.xml

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

    <TextView
        android:id="@+id/text1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!" />

</LinearLayout>
```

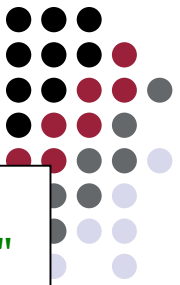
MainActivity.java

```
public class MainActivity extends Activity {

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

        //Changes the text in the layout file
        TextView textView = (TextView) findViewById(R.id.text1);
        textView.setText("Hello George!");
    }
}
```

Interação entre Activity e View



activity_main.xml

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

    <TextView
        android:id="@+id/text1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Type your name:" />

    <EditText
        android:id="@+id/editText2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:inputType="textPersonName" />

    <Button
        android:id="@+id/button"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Confirm"
        android:onClick="sendMessage" />

</LinearLayout>
```

Interação entre Activity e View



MainActivity.java

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

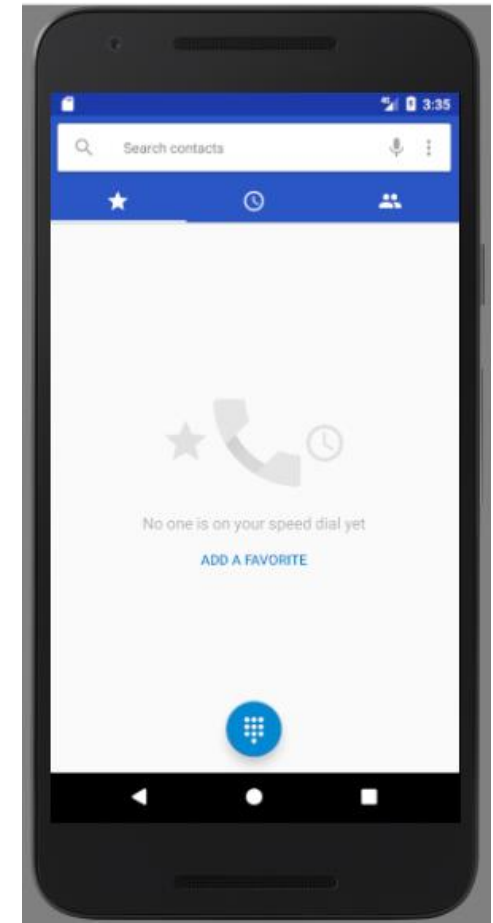
    //Method that handles the click on the button on the activity's layout
    public void sendMessage(View view) {
        //Recover the text typed in the editText component
        TextView editText = (TextView) findViewById(R.id.editText2);
        String name = editText.getText().toString();
        //Prints a brief message in the screen
        Toast.makeText(this, "Bem vindo " + name, Toast.LENGTH_SHORT).show();
    }
}
```


Iniciar nova Activity (Intent)



MainActivity.java

```
public class MainActivity extends Activity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        Intent it = new Intent(Intent.ACTION_DIAL);  
        startActivity(it);  
    }  
}
```

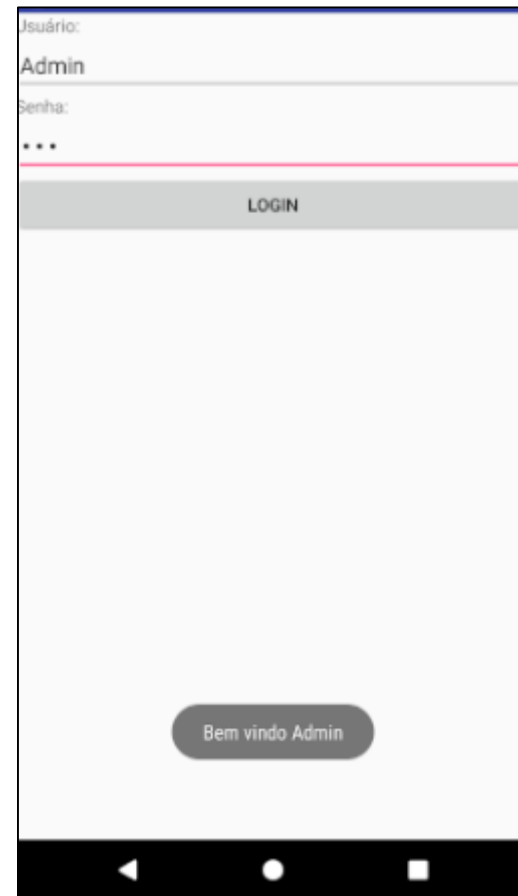


Exercício



Projeto A04 Login.rar

- Criar uma tela de login que aceite apenas usuário Admin e senha 123
- Uma mensagem de erro (Toast) deve ser exibida caso entrada seja inválida





- Google and Open Handset Alliance n.d. **Android API Guide**. <http://developer.android.com/guide/index.html>. Acessado em Maio de 2017.
- Google and Open Handset Alliance n.d. **Android training guide**. <http://developer.android.com/training/index.html>. Acessado em Maio de 2017.
- Lecheta, R. R. **Google Android: Aprenda a criar aplicações para dispositivos móveis com o Android SDK**. 3ª edição. São Paulo: Novatec Editora, 2013.

