

Android: Primeiros Projetos



Exercício 1

primeira aplicação

Estrutura do projeto

Class R

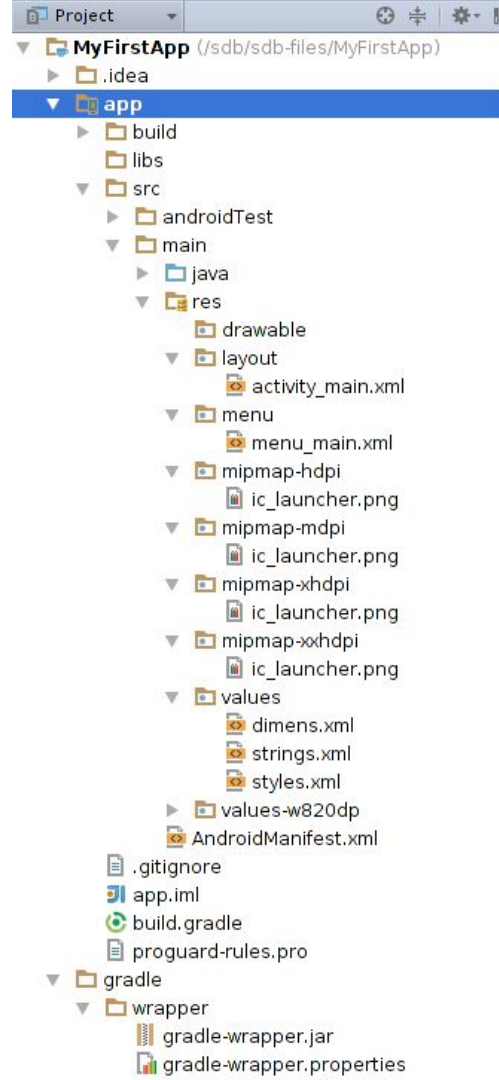
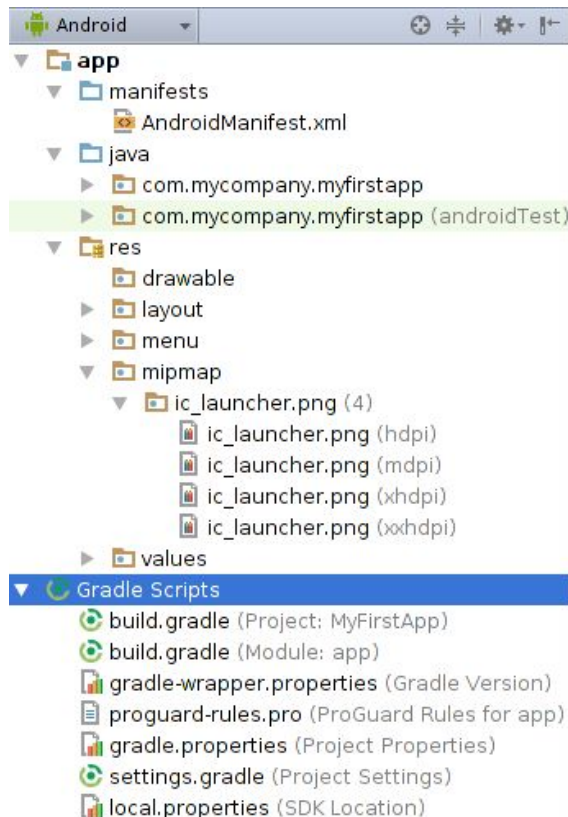
Drawable Resources

Layout Resources

Menu Resources

String Resources

Style Resource



A classe R

- Define constantes para acesso aos recursos do projeto.
- Ao utilizar um recurso, não precisamos colocar sua extensão.
- O nome dos arquivos adicionados, devem ser minúsculos e sem espaços, caso contrário, a classe R não será compilada corretamente.

Suporte a múltiplas resoluções

MyProject/

res/

drawable-xhdpi/

awesomeimage.png

drawable-hdpi/

awesomeimage.png

drawable-mdpi/

awesomeimage.png

drawable-ldpi/

awesomeimage.png

res/

layout/

main.xml

layout-land/

main.xml

layout-large/

main.xml

layout-large-land/

main.xml

Layout

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

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!" />

</RelativeLayout>
```

Android Manifest

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="firstapp.unibratec.edu.br.firstapp">
    <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">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

strings.xml

```
<resources>
```

```
    <string name="app_name">My First App</string>
```

```
    <string name="edit_message">Enter a message</string>
```

```
    <string name="button_send">Send</string>
```

```
    <string name="action_settings">Settings</string>
```

```
</resources>
```


strings.xml

/values/strings.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="title">My Application</string>
    <string name="hello_world">Hello World!</string>
</resources>
```

/values-es/strings.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="title">Mi Aplicación</string>
    <string name="hello_world">Hola Mundo!</string>
</resources>
```

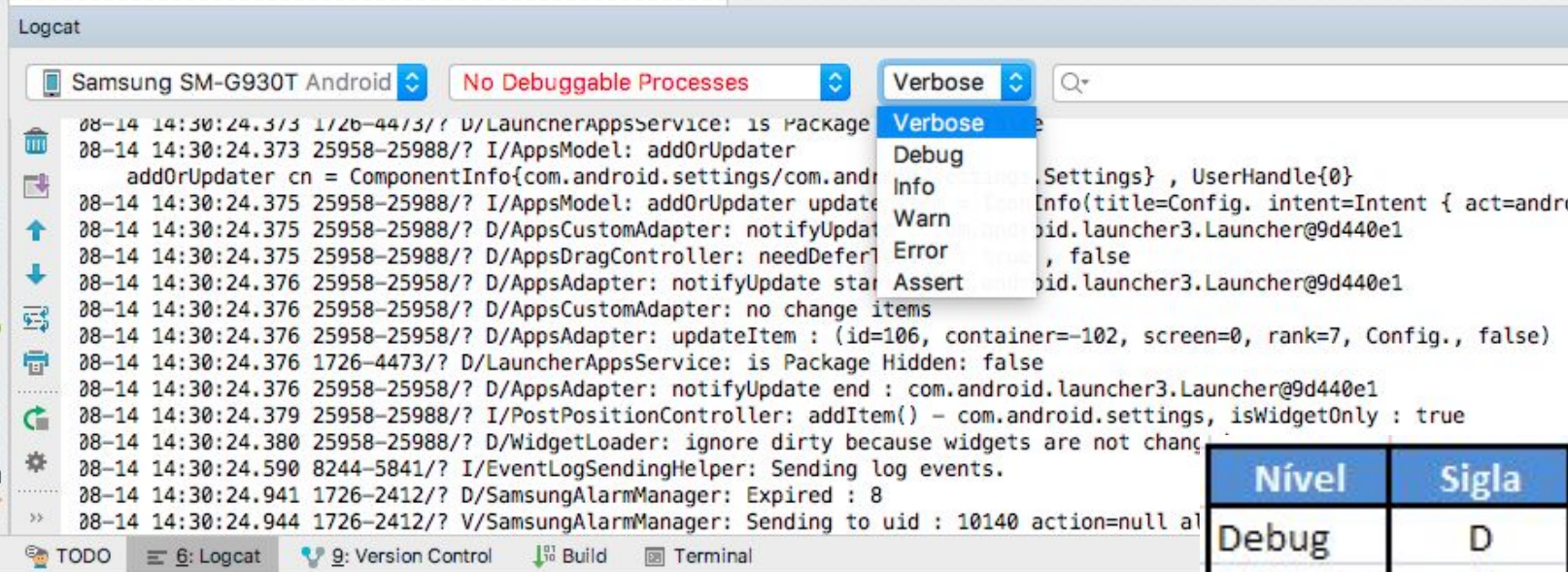
Gradle

- Sistema para automação de Builds.
- Utiliza o Groovy como linguagem.
- Utiliza tasks como diretivas.
- Possibilidade de baixar dependências dinamicamente.

build.gradle

```
android {  
    compileSdkVersion 23  
    buildToolsVersion "23.0.3"  
  
    defaultConfig {  
        applicationId "firstapp.unibratéc.edu.br.firstapp"  
        minSdkVersion 22  
        targetSdkVersion 23  
        versionCode 1  
        versionName "1.0"  
    }  
}  
  
dependencies {  
    compile fileTree(dir: 'libs', include: ['*.jar'])  
    testCompile 'junit:junit:4.12'  
    compile 'com.android.support:appcompat-v7:23.4.0'  
}
```

Logcat



The screenshot shows the Logcat window in Android Studio. The top bar indicates the device is a Samsung SM-G930T Android. The filter is set to 'No Debuggable Processes'. A dropdown menu is open, showing log levels: Verbose (selected), Debug, Info, Warn, Error, and Assert. The log messages below show various system events, including package updates, app model changes, and launcher actions.

Nível	Sigla
Debug	D
Verbose	V
Info	I
Warn	W
Error	E

Log

```
Logger.java x
1 package br.edu.unibratec.firstapp.firstapp;
2
3 import android.util.Log;
4
5 public final class Logger {
6
7     private Logger() {
8
9     }
10
11     public static void v(String message) {
12         if (BuildConfig.DEBUG) {
13             Log.v(Constants.APP_LOG_TAG, message);
14         }
15     }
16 }
17
```

Activity

- Uma Activity representa uma tela da aplicação
- É responsável por gerenciar seu ciclo de vida
- Gerenciadas pela pilha de Activity
- Tem controle sobre os componentes presentes na tela
- Uma Activity deve ser declarada obrigatoriamente no AndroidManifest.

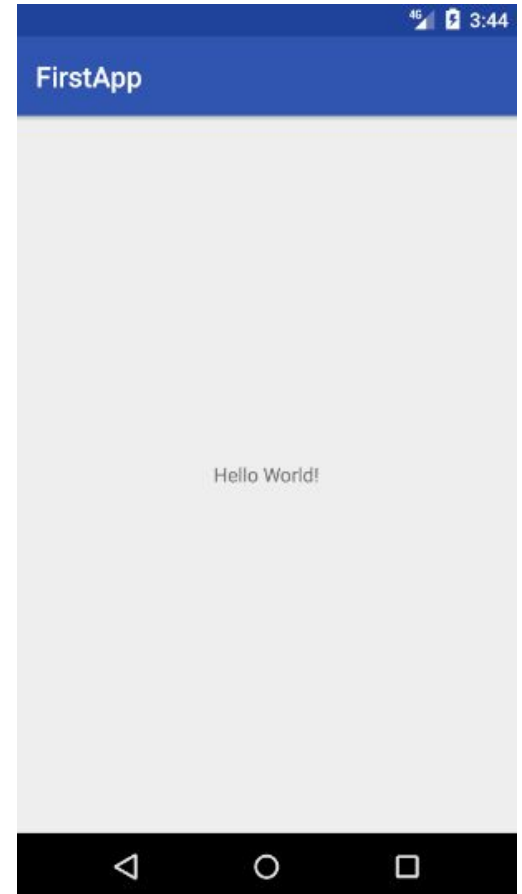
Activity



+



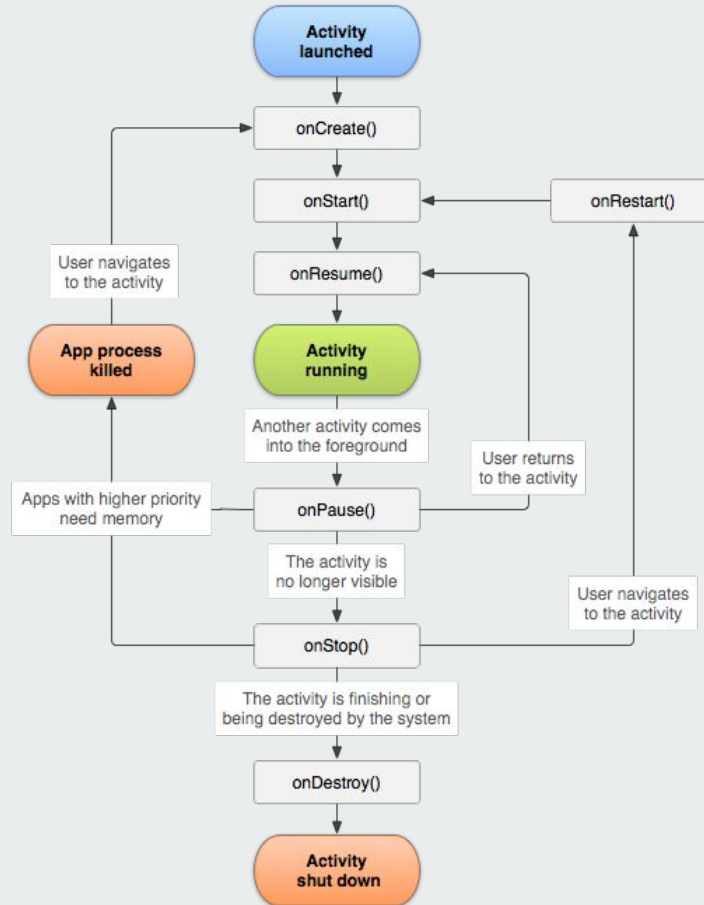
=



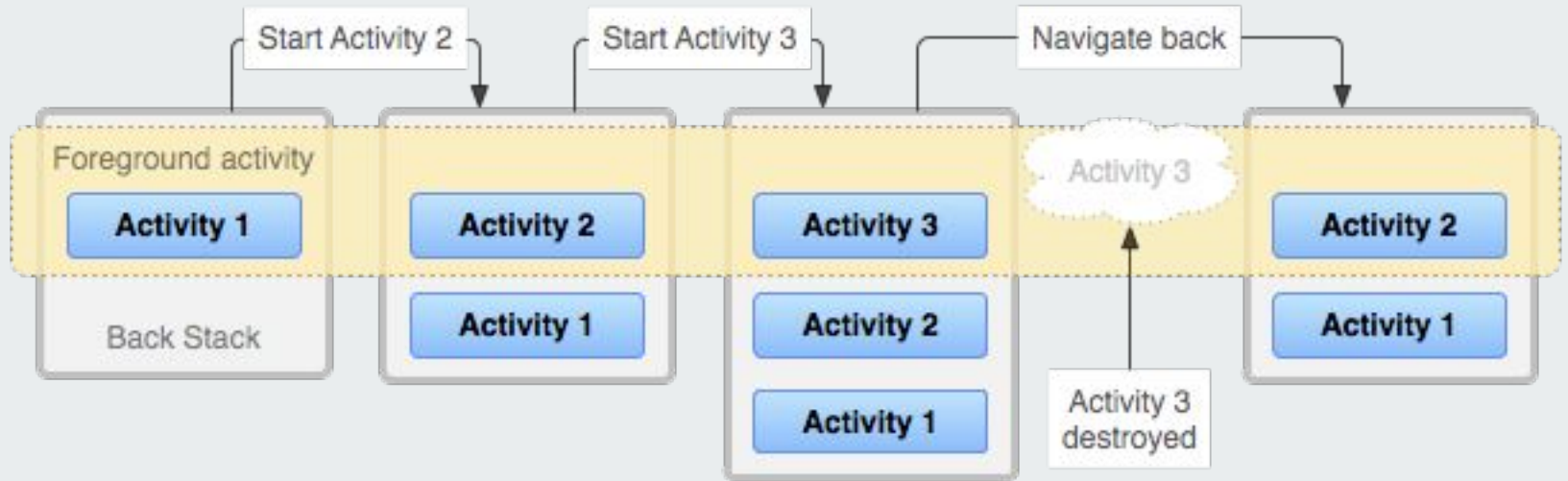
Activity no AndroidManifest

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


Ciclo de vida da Activity



Pilha de Activity



Encerrando uma Activity

- Para encerrar um Activity utilizamos o método **finish()**
- Ao utilizá-lo, o **onDestroy()** é chamado

Exercício 2

ciclo de vida

Exercício 3

conceitos básicos

Desafio 1

Álcool ou Gasolina

Navegação entre telas

- O método **startActivity(Intent)** é usado para iniciar uma nova Activity)
- O método sempre deve receber por parâmetro uma **Intent**
- Ao inicializar, a nova Activity ficará no **topo** da pilha

Exercício 4

navegação entre telas

Passando parâmetros para outras telas

- Passamos parâmetros para outras telas através das **Intent's**
- O método **putExtra** adiciona novos valores a Intent que iniciará a nova Activity

Exercício 5

passando parâmetros
para outras telas

Recebendo parâmetros de outras telas

- Às vezes necessitamos obter um resultado de uma Activity quando ela termina sua execução
- Para isso, utilizamos o método `startActivityForResult(Intent, Data)`
- Você receberá o retorno da execução no método `onActivityResult(Intent, Data)`

Exercício 6

recebendo parâmetros
de outras telas

Desafio 1

Calculadora