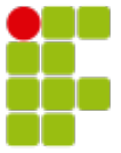


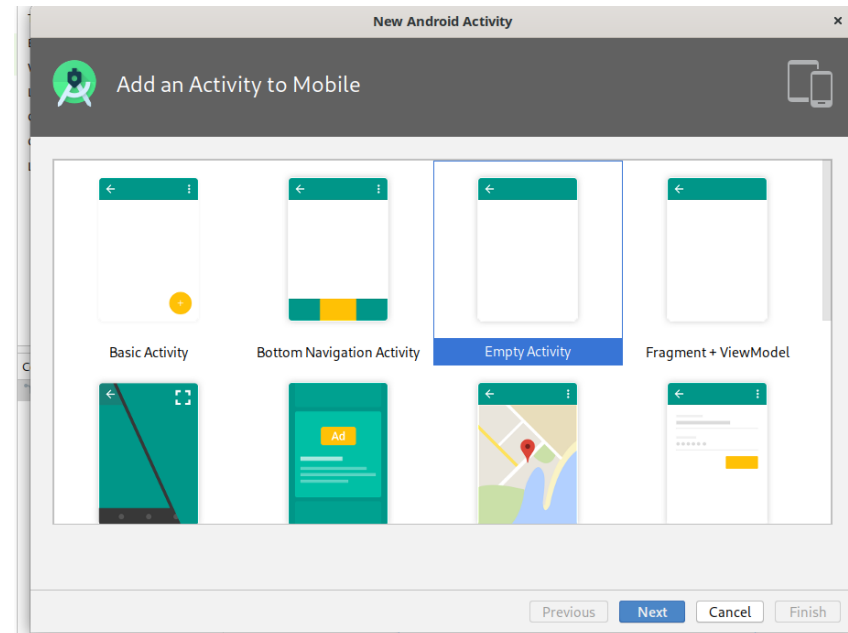
Activity

Professor Msc Romulo Beninca



Activity

Activity é uma classe da plataforma *android*, que desempenha a função de criar uma tela, e serve também como ponto de entrada para execução da aplicação ou de funções dela. As *activitys* podem interagir entre si, solicitando e passando informações.

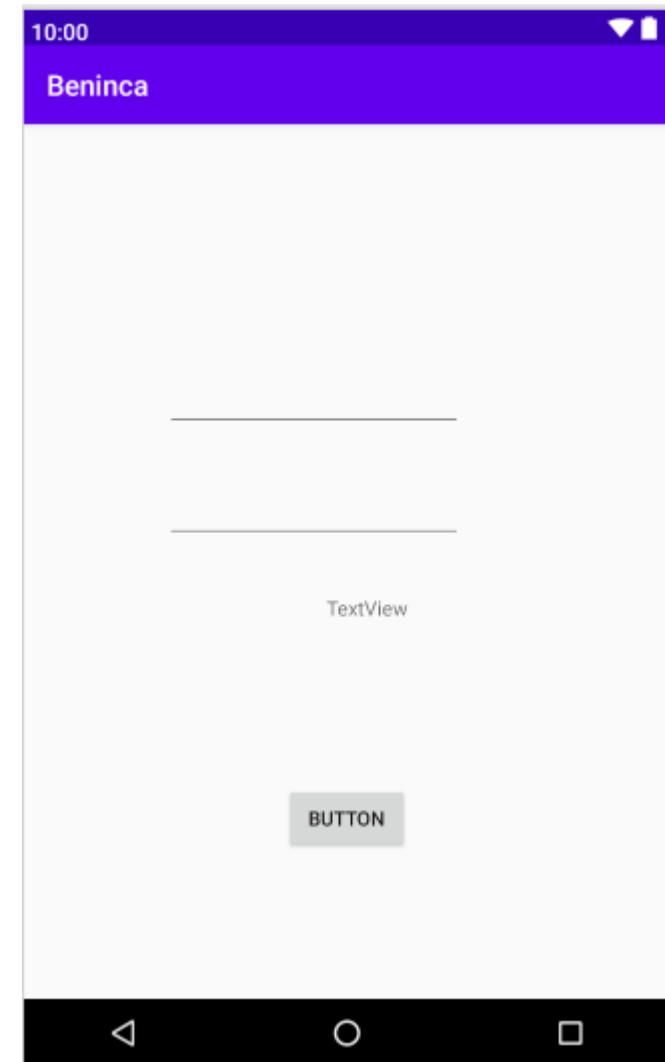


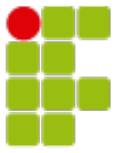


Activity

Ao iniciar uma aplicação uma activity é inicializada, ela pode ocupar toda tela ou parte dela.

- Dentro de uma aplicação *android* é comum que cada aplicação tenha uma função específica como login e configuração, listagem ...



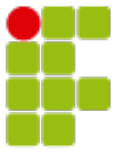


Activity no manifesto

Toda *Activity* é declarada no arquivo *manifest*, como explicito abaixo, a *activity*.

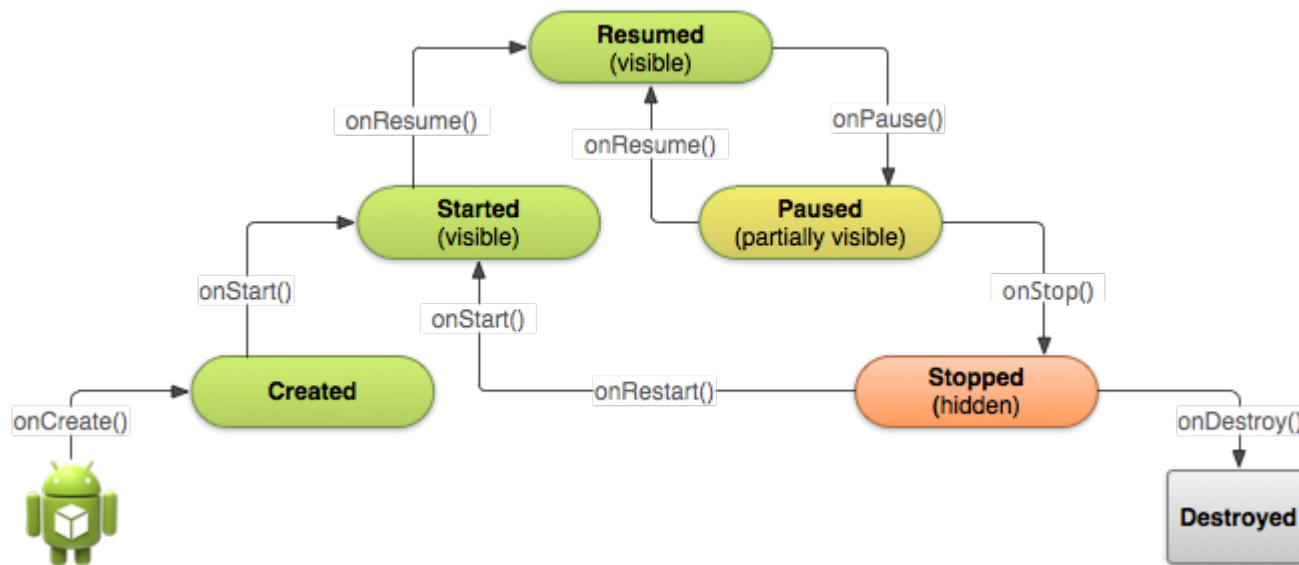
AndroidManifest.xml

```
<application
...
  <activity android:name=".MainActivity">
...
</activity>
...
/>
```



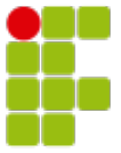
Lifecycle activity Android

O ciclo de vida de uma *activity* são os estados que uma *activity* passa durante seu tempo de vida e são vistos pelo programador como métodos no código, chamados no início de cada estado.



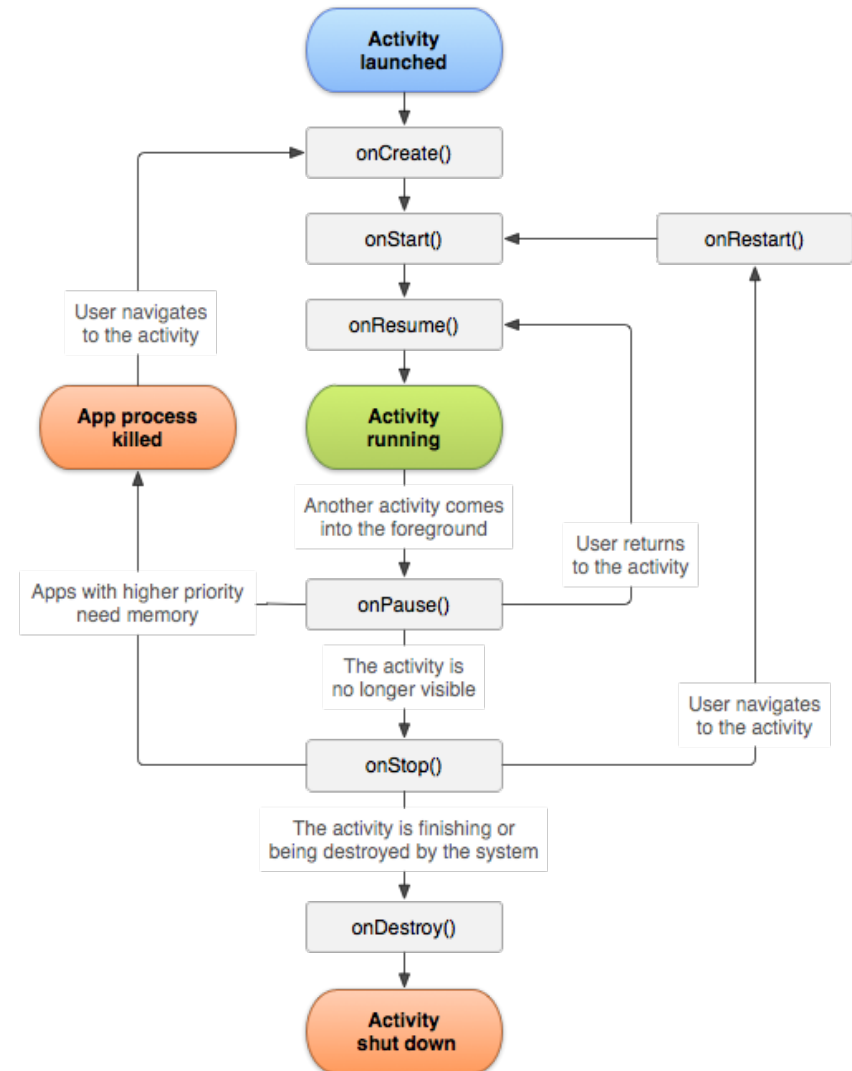
fonte e leitura recomendada:

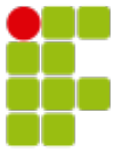
<https://developer.android.com/guide/components/activities/activity-lifecycle>



Métodos *callbacks* activity

Os métodos onCreate, onStart, onResume, onPause, onStop, onRestart, onDestroy são *callbacks* que possibilitam a aplicação executar uma ação durante a transição dos estados.





Acompanhando o ciclo de vida pelo log

Para utilizar o logCat na sua aplicação basta incluir a classe Log e utilizar um dos métodos estáticos a seguir:

Log.v("TAG", "Value")

Log.d("TAG", "Value")

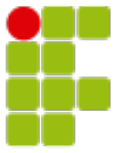
Log.i("TAG", "Value")

Log.w ("TAG", "Value")

Log.e("TAG", "Value")

Aplicação

Crie uma aplicação que faça a chamadas dos métodos callbacks.



Acompanhando o ciclo de vida pelo log

MainActivity.java

```
protected void onCreate(Bundle savedInstanceState) {  
    ...  
    Log.d("Ciclo de Vida Activity ", "onCreate ");  
}  
@Override  
protected void onStart() {  
    super.onStart();  
    Log.d("Ciclo de Vida Activity ", "onStart ");  
}  
@Override  
protected void onResume() {  
    super.onResume();  
    Log.d("Ciclo de Vida Activity ", "onResume ");  
}  
@Override  
protected void onPause() {  
    super.onPause();  
    Log.d("Ciclo de Vida Activity ", "onPause ");  
}  
@Override  
protected void onRestart() {
```

logcat

Emulador Pixel_3a_API_30

com.ifsc.minhasactivities (86!

Debug

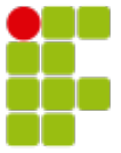
Ciclo

Regex

Show only selected applicatio

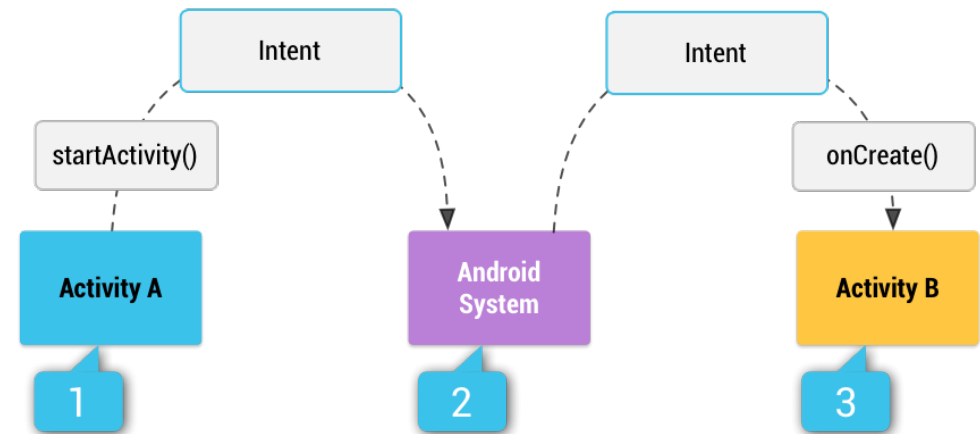
logcat

```
2021-02-17 22:46:24.081 8652-8652/com.ifsc.minhasactivities D/Ciclo de Vida Activity: onCreate  
2021-02-17 22:46:24.088 8652-8652/com.ifsc.minhasactivities D/Ciclo de Vida Activity: onStart  
2021-02-17 22:46:24.125 8652-8652/com.ifsc.minhasactivities D/Ciclo de Vida Activity: onResume  
2021-02-17 23:10:51.606 8652-8652/com.ifsc.minhasactivities D/Ciclo de Vida Activity: onPause  
2021-02-17 23:10:52.529 8652-8652/com.ifsc.minhasactivities D/Ciclo de Vida Activity: onStop
```



Intents no Android

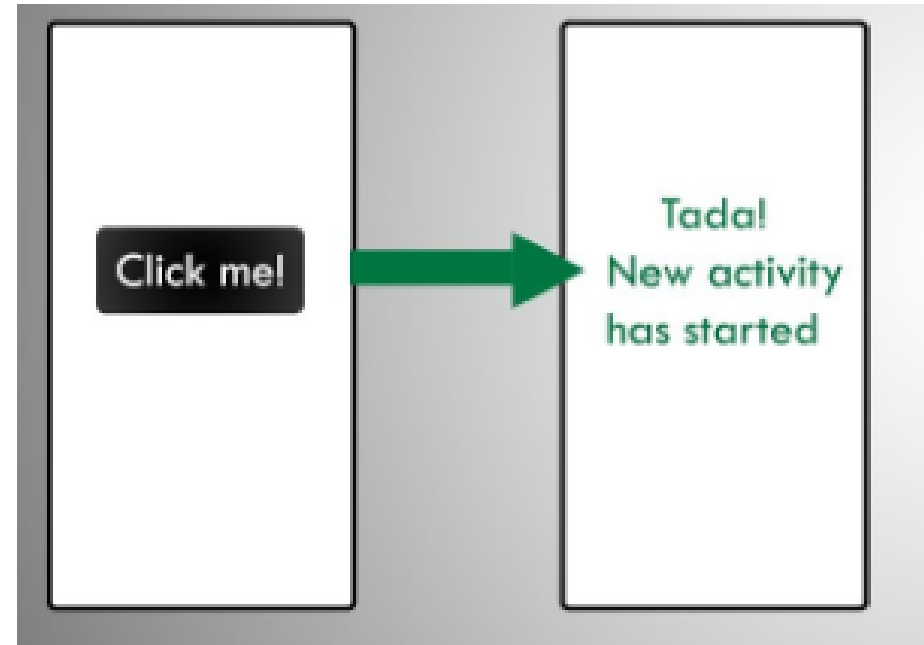
A comunicação entre diferentes Activitys é feita por meio de Intents que são encaminhadas ao sistema operacional que as trata encaminhando as ações necessárias.



Iniciando nova *Activity*

Os filtros permitem iniciar uma *activity* não apenas com base em uma solicitação explícita do usuário, mas também a depender de ações do sistema e outras aplicações.

- Por meio das *intents* também é possível passar dados entre duas *activities*.



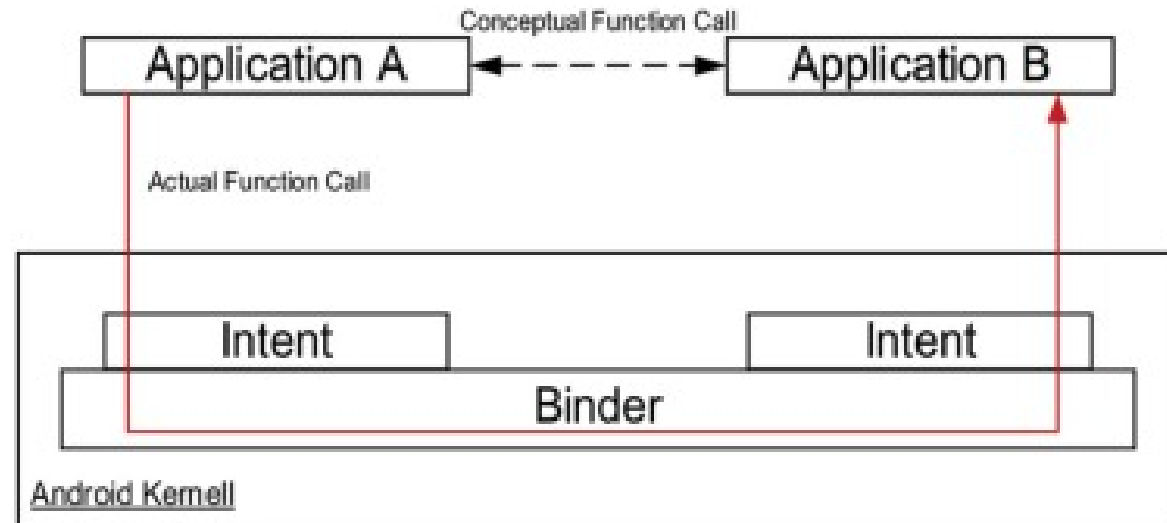
```
public void iniciaNovaActivity (View v){  
    Intent i = new Intent(getApplicationContext(),IMC.class);  
    startActivity(i);  
}
```

Activity no manifesto filtros de *intent*

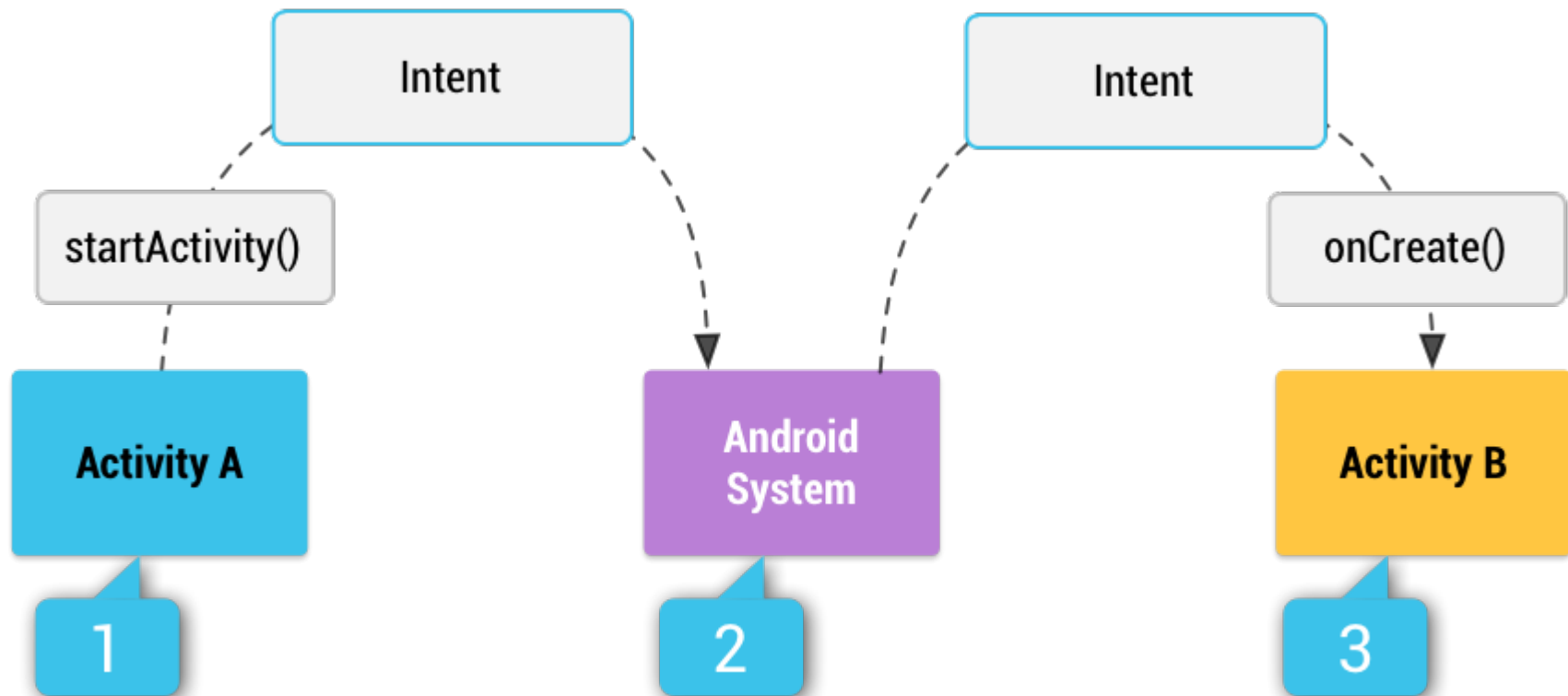
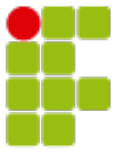


Os filtros permitem iniciar uma *activity* não apenas com base em uma solicitação explícita do usuário, mas também a depender de ações do sistema e outras aplicações.

- Por meio das *intents* também é possível passar dados entre duas *activities*.

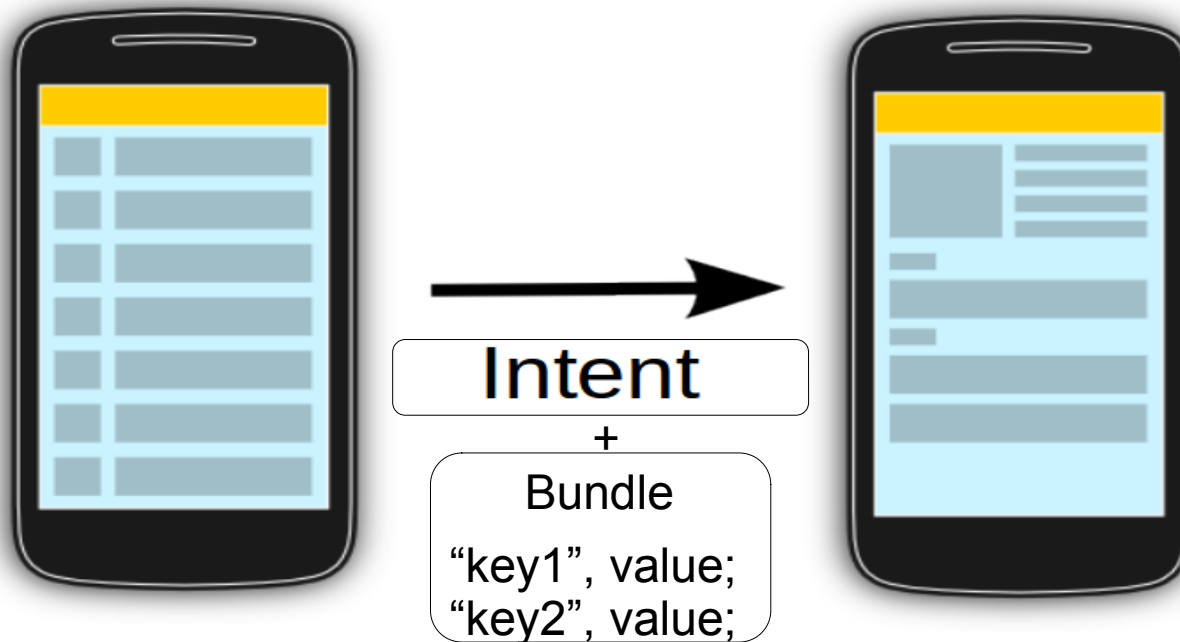


```
activity android:name=".ExampleActivity"  
android:icon="@drawable/app_icon">  
  <intent-filter>  
    <action android:name="android.intent.action.SEND" />  
    <category android:name="android.intent.category.DEFAULT" />  
    <data android:mimeType="text/plain" />  
  </intent-filter>  
</activity>
```



Passagem de dados entre *activitys*

Para passar dados entre duas *activitys* podemos utilizar a intenção de chamada passando os dados por meio do método `putExtra()`.



Passagem de dados entre *activitys*



MainActivity.java ×

```
public void iniciaNovaActivity(View v ){  
    Intent i = new Intent(this,Activity2.class);  
    String mensagem=editTextMsg.getText().getString();  
    i.putExtra("msg",mensagem);  
    startActivity(i);  
}
```

Activity2.java ×

```
{..  
    Bundle bundle= getIntent().getExtras();  
    String string= bundle.getString("msg");  
    ..  
}
```



INSTITUTO FEDERAL
SANTA CATARINA



Instituto Federal de Santa Catarina

Aplicação IMC

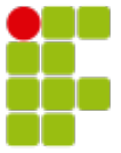
Agora que já sabemos :

- Criar um novo Activity no nosso projeto.
- Definir a *Activity* que será carregada com a aplicação.
- Chamar uma nova *Activity* no nosso projeto.
- Passar os dados informados pelo usuário de uma *Activity* para outra.
- Recuperar os dados informados em uma *Activity*

Atividade

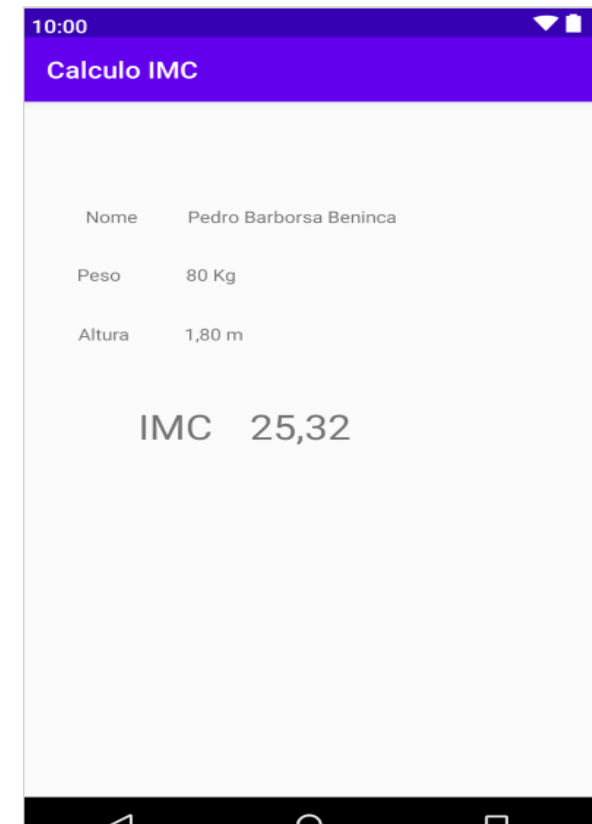
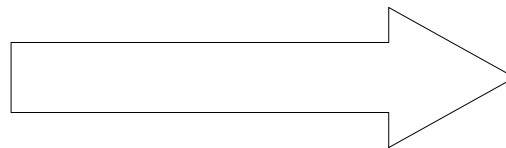
Crie o aplicativo IMC, que deve receber como parâmetro a altura do usuário em metros e o peso em Kg e exibir o valor do IMC.

$$\text{IMC} = \frac{\text{PESO}}{\text{ALTURA}^2}$$



Atividade para entregar

Desenvolver uma aplicação para realizar cálculo do Índice de massa corporal (IMC). A aplicação deve solicitar ao usuário na primeira tela o seu nome, altura e peso. Ao clicar no botão calcular IMC deve-se encaminhar o usuário para uma segunda tela em que será reexibido os valores informados e também o cálculo do IMC.



Gerenciadores de *layout*

Relative Layout - Atividade APP IMC Passo 9

10:00

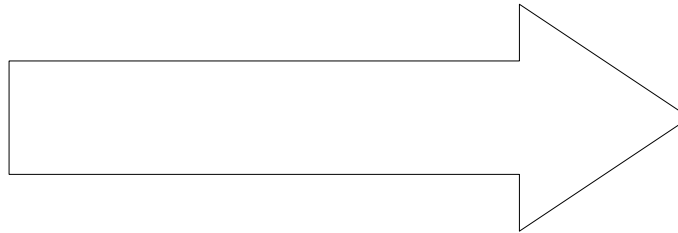
Aplicativo Calculo IMC

Nome _____

Peso _____

Altura _____

CALCULA IMC



10:00

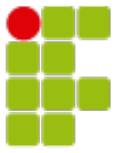
Calculo IMC

Nome Pedro Barborsa Beninca

Peso 80 Kg

Altura 1,80 m

IMC 25,32



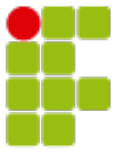
Lembrando como formatar valores

Para formatar valores no Java podemos utilizar as classes especializadas DecimalFormat do pacote java.text.

```
DecimalFormat decimalFormat = new DecimalFormat("###,###,###,##0.00");  
decimalFormat.format(2.333333333333333);
```

fonte:

<https://docs.oracle.com/javase/7/docs/api/java/text/DecimalFormat.html>



Referências

- Lecheta, R. Google Android: Aprenda a criar aplicações para dispositivos móveis. 3a Ed. Novatec, 2013.
- Androd Developer. Android Disponível em:<<https://developer.android.com>>
- Androd Developer, 2018. Relative Parameters Disponível em:<<https://developer.android.com/reference/android/widget/RelativeLayout.LayoutParams>>