

Desenvolvimento de aplicações móveis

Interação entre activities

Prof. Vanilson Burégio, PhD



Agenda

- ▶ Ambiente
 - ▶ Habilitando o uso de lambda
- ▶ Compatibilidade entre versões
- ▶ Salvando estado
- ▶ Mudando o layout de acordo com a orientação
- ▶ Interação entre activities
- ▶ Atividade

Ambiente

Habilitando o uso de lambda, especificação das versões do SDK

Habilitando o uso da notação lambda

► Exemplo

```
btnCalc.setOnClickListener(v) -> {
    try{
        double vgas = Double.parseDouble(edtGas.getText().toString());
        double valc = Double.parseDouble(edtAlc.getText().toString());

        if(valc/vgas <= 0.7){
            resultado = ALC;
        }else{
            resultado = GAS;
        }
    }catch (Exception e){
        resultado = SHIT;
    }
    atualizarTextoResultado();
};
```

Habilitando o uso da notação lambda

► Passos

1. Instalar e configurar o *build tool version 24*
2. Habilitar a opção *jack*

gradle.build

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 23
    buildToolsVersion "24.0.2"

    defaultConfig {
        applicationId "mobile.ufrpe.br.hello"
        minSdkVersion 15
        targetSdkVersion 23
        versionCode 1
        versionName "1.0"                                Importante saber a diferença
        jackOptions {
            enabled true
        }
    }
    compileOptions {
        targetCompatibility 1.8
        sourceCompatibility 1.8
    }
}
```

Prof. Vanilso

Exemplo

```
btnCalc.setOnClickListener((v) -> {
    try{
        double vgas = Double.parseDouble(edtGas.getText().toString());
        double valc = Double.parseDouble(edtAlc.getText().toString());

        if(valc/vgas <= 0.7){
            resultado = ALC;
        }else{
            resultado = GAS;
        }
    }catch (Exception e){
        resultado = SHIT;
    }
    atualizarTextoResultado();
});
```

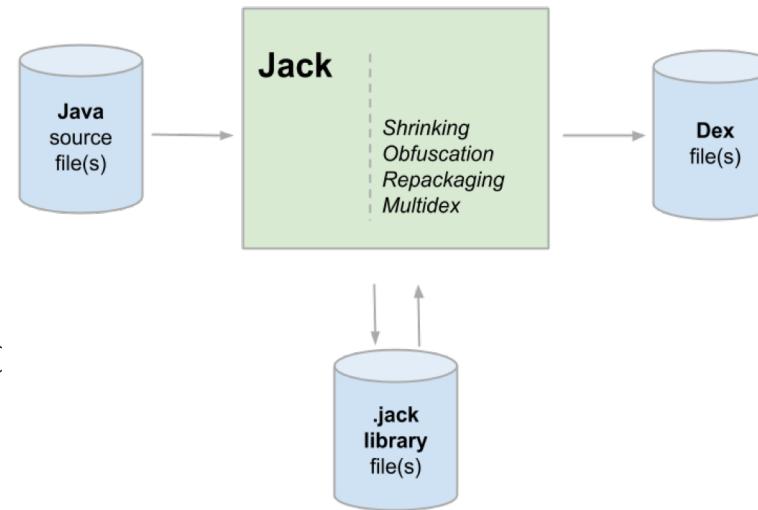
O que é
Jack?

Habilitando o uso da notação lambda

► Java Android Compiler Kit –

Jack

- Completamente **open source**
- Reduz tempo de **compilação**
- Manipula **obfuscation, repackaging**, etc



“Jack is a new Android toolchain that compiles Java source into Android dex bytecode. It replaces the previous Android toolchain, which consists of multiple tools, such as javac, ProGuard, jarjar, and dx.”

Fonte: <https://source.android.com/source/jack.html>

Exemplo

```
btnCalc.setOnClickListener((v) -> {  
    try{  
        double vgas = Double.parseDouble(edtGas.getText().toString());  
        double valc = Double.parseDouble(edtAlc.getText().toString());  
  
        if(valc/vgas <= 0.7){  
            resultado = ALC;  
        }else{  
            resultado = GAS;  
        }  
    }catch (Exception e){  
        resultado = SHIT;  
    }  
    atualizarTextoResultado();  
});
```

gradle.build

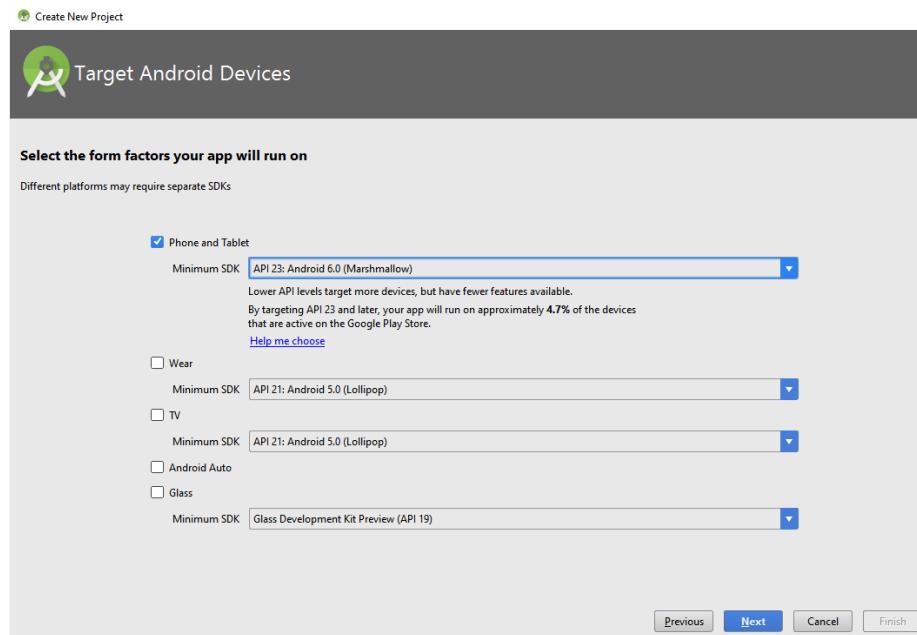
```
apply plugin: 'com.android.application'  
  
android {  
    compileSdkVersion 23  
    buildToolsVersion "24.0.2"  
  
    defaultConfig {  
        applicationId "mobile.ufrpe.br.hello"  
        minSdkVersion 15  
        targetSdkVersion 23  
        versionCode 1  
        versionName "1.0"  
        jackOptions {  
            enabled true  
        }  
        compileOptions {  
            targetCompatibility 1.8  
            sourceCompatibility 1.8  
        }  
    } ...
```

Compatibilidade entre versões

Compile SDK, min SDK e target SDK

Sobre versão do Android

- ▶ Uma aplicação compilada na versão 4.0.3 executa em um dispositivo com Android 6.0?
- ▶ E o contrário? Seria possível utilizar classes do SDK mais novo e a app executar em aparelhos com versões anteriores?



30/10/2018

Sobre versões do Android

- ▶ Como utilizar **classes de um SDK mais novo** e ainda assim a **app executar em aparelhos com versões anteriores?**

gradle.build

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 23
    buildToolsVersion "24.0.2"

    defaultConfig {
        applicationId "mobile.ufrpe.br.hello"
        minSdkVersion 15
        targetSdkVersion 22
        versionCode 1
        versionName "1.0"
        jackOptions {
            enabled true
        }
    }
    compileOptions {
        targetCompatibility 1.8
        sourceCompatibility 1.8
    }
} ...
```

Existem classes que só executam em uma versão específica, entretanto, outras podem ser disponibilizadas pelo compilador e serem executadas em versões anteriores.

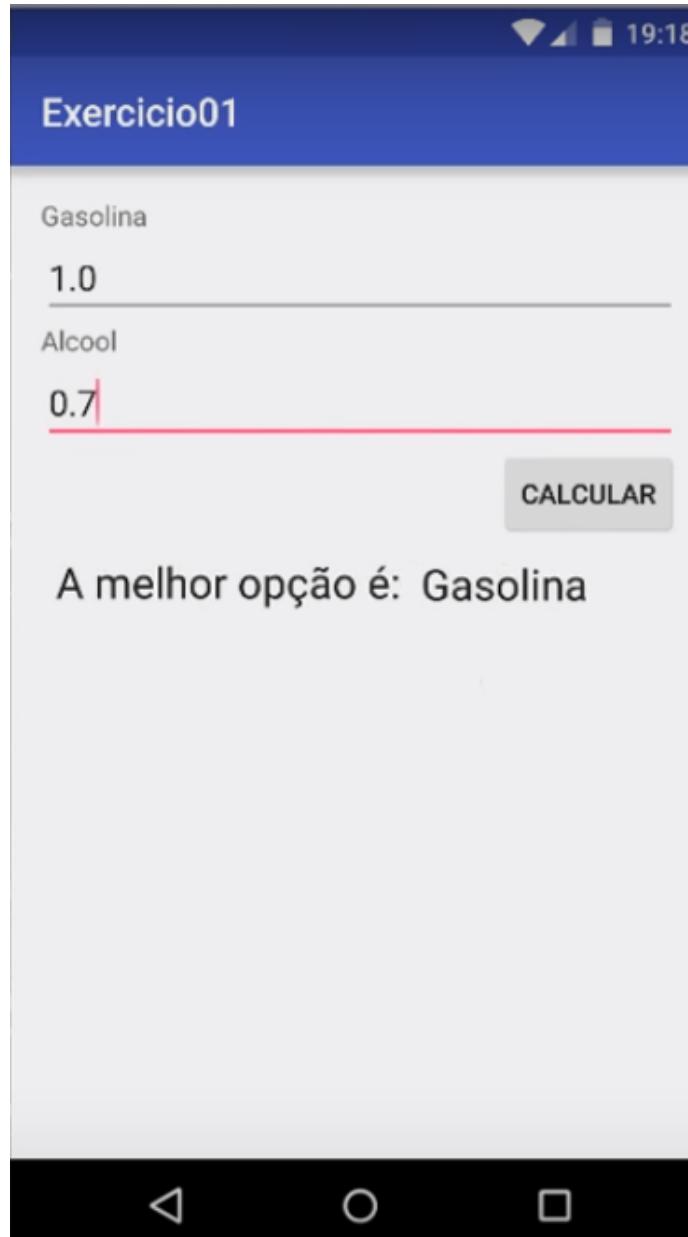
- ▶ **Compile SDK** – determina a versão do android em que você escreve o seu código.
- ▶ **Min SDK** – determina a versão mínima a partir da qual a aplicação pode executar.
- ▶ **Target SDK** – determina a versão na qual estou testando minha aplicação. Em alguns casos, determina com será o comportamento da aplicação.
 - ▶ Ex.: target 22 e aparelho com API 23, o esquema de permissões da app será o definido pela API level 22)

Aula passada...

Rotação (demonstração), Activity/Intent e ciclo de vida de uma aplicação

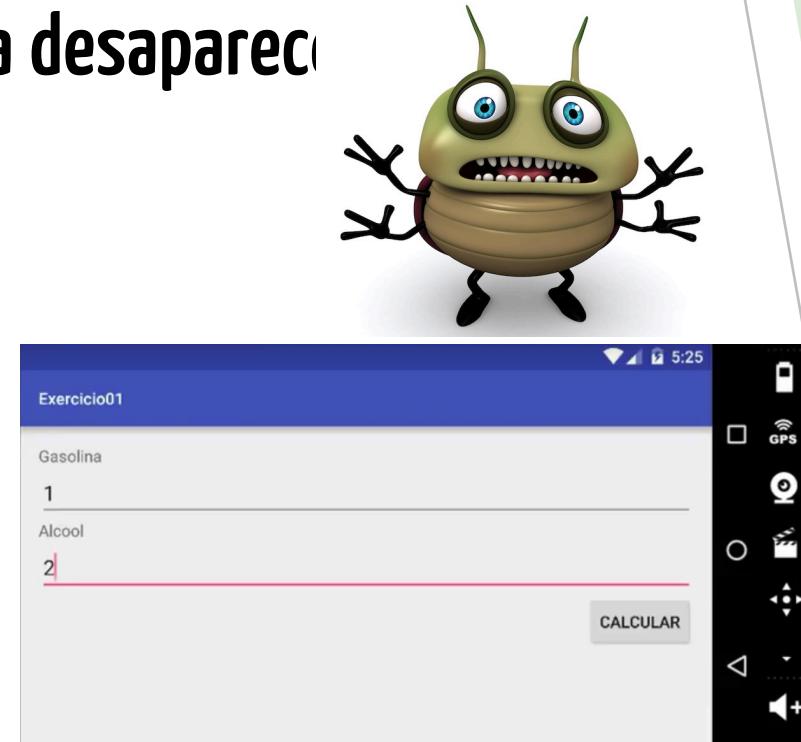
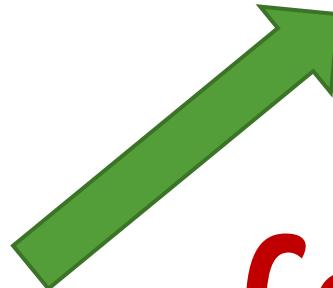
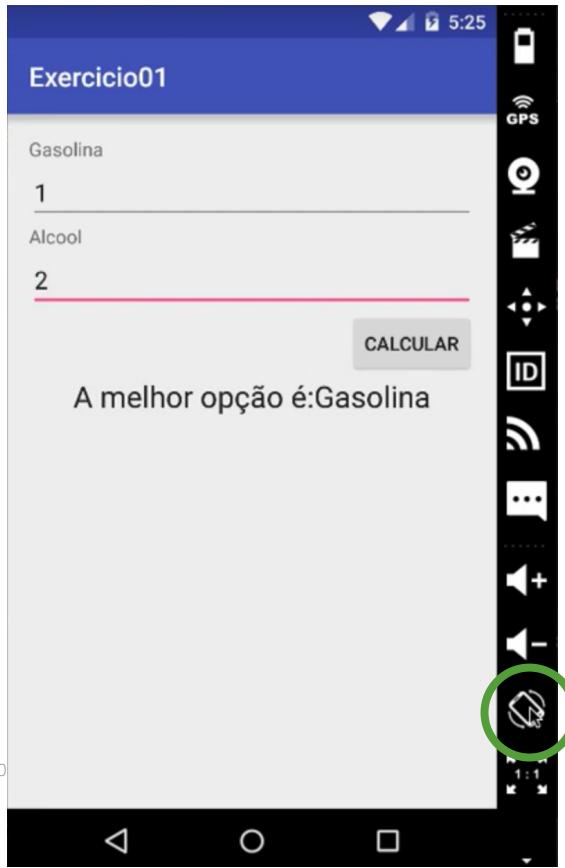
Dojo

- Atividade passada na última aula



Rotação

- Ao rotacionar o dispositivo, a resposta desaparece



Como resolvemos?

Salvando estado

Activity :: onSaveInstanceState

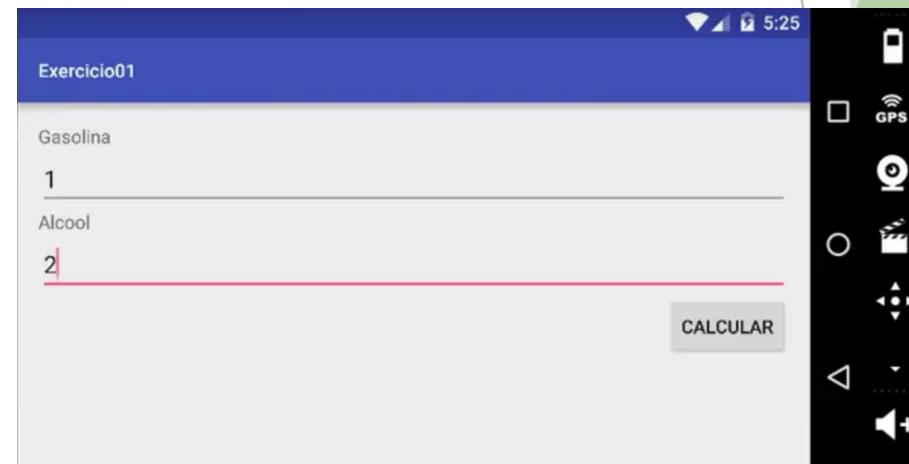
```
@Override  
public void onSaveInstanceState(Bundle bundle)  
{  
    super.onSaveInstanceState(bundle);  
    bundle.putInt("resultado", resultado);  
}
```

Atributo que criamos para armazenar o estado

Activity :: onCreate

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
  
    setContentView(R.layout.activity_main);  
  
    // inicialização dos elementos...  
  
    if (savedInstanceState != null) {  
        resultado = savedInstanceState.getInt("resultado");  
        atualizarTextoResultado();  
    }  
    ...  
}
```

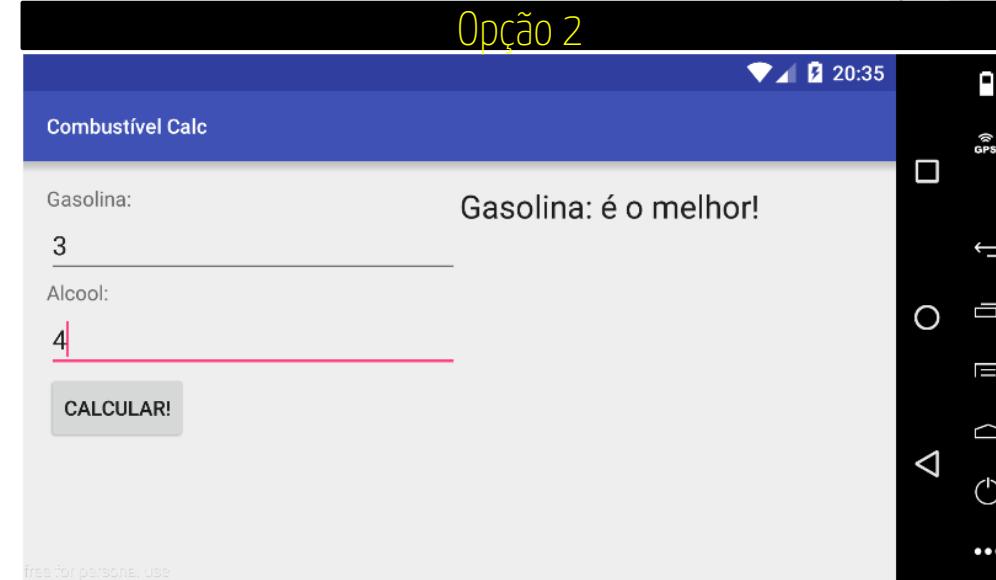
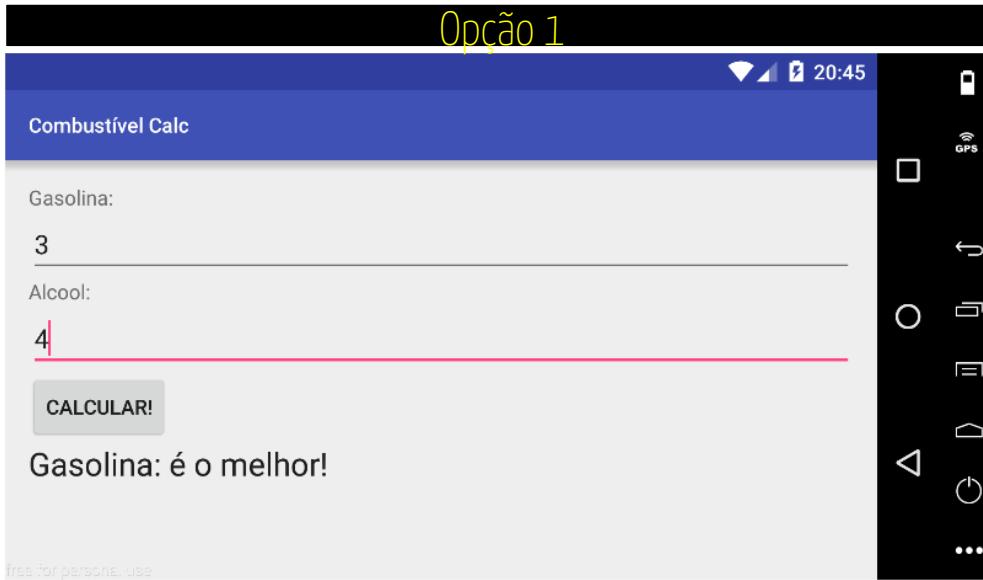
Se não chamar o super o que for digitado nos campos desaparece.



Verifica se há algum estado salvo e recupera, caso exista

Atividade - Orientação

- ▶ Como faríamos para, no modo paisagem, apresentarmos o layout conforme mostrado na opção 2?



0 que deve ser feito?

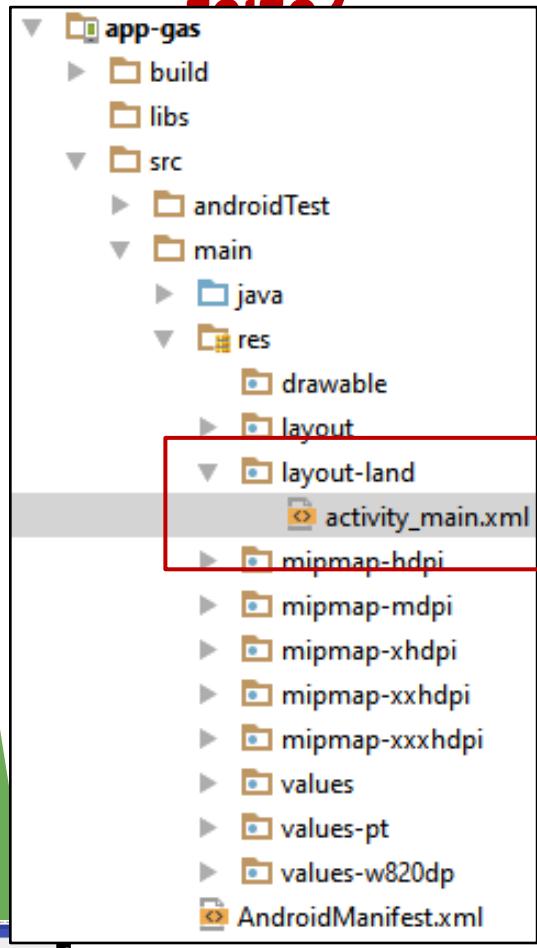
30/10/2018

Atividade - Orientação

- ▶ Como faríamos para, no modo paisagem, apresentarmos o layout conforme mostrado na opção 2?

O que deve ser

Opção 2



```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_weight="1"
    android:orientation="horizontal">

    <LinearLayout
        android:layout_width="0dp"
        android:layout_weight="1"
        android:orientation="vertical"
        android:layout_height="match_parent">

        <TextView
            ... />

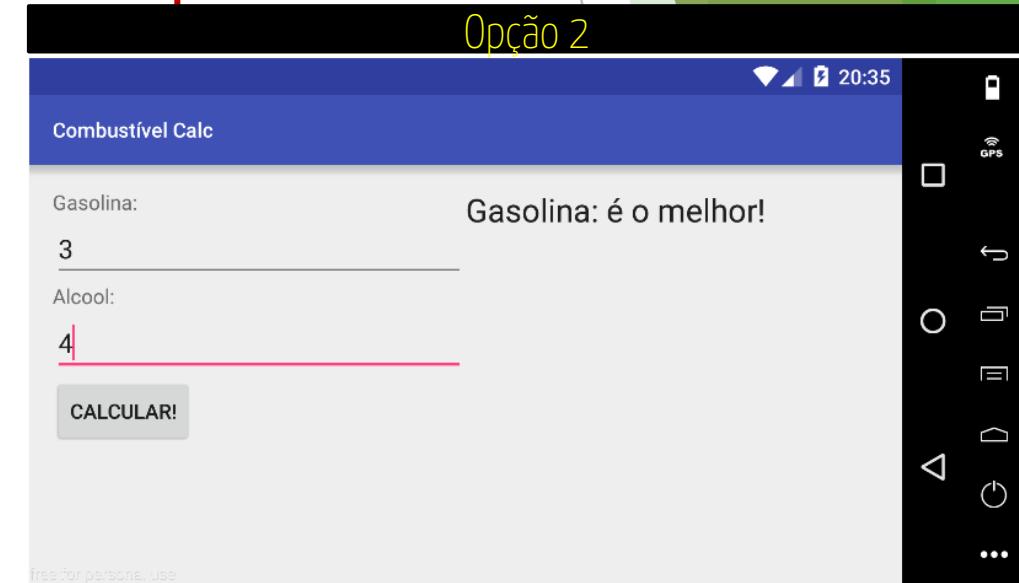
        <EditText
            ... />

        <TextView
            ... />

        <EditText
            ... />

        <Button
            ... />
    </LinearLayout>

    <TextView
        android:id="@+id/tvResultado"
        android:layout_width="0dp"
        android:layout_weight="1"
        android:layout_height="match_parent"
        android:textAppearance="?android:attr/textAppearanceLarge" />
</LinearLayout>
```

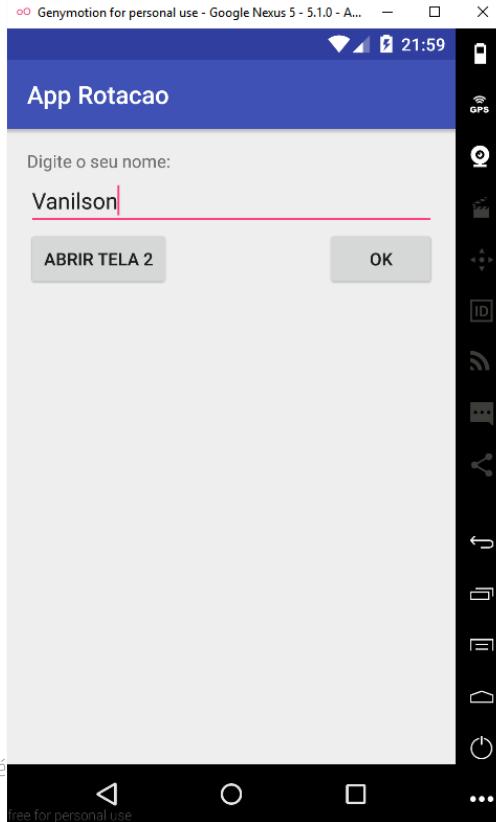


Interação entre activities

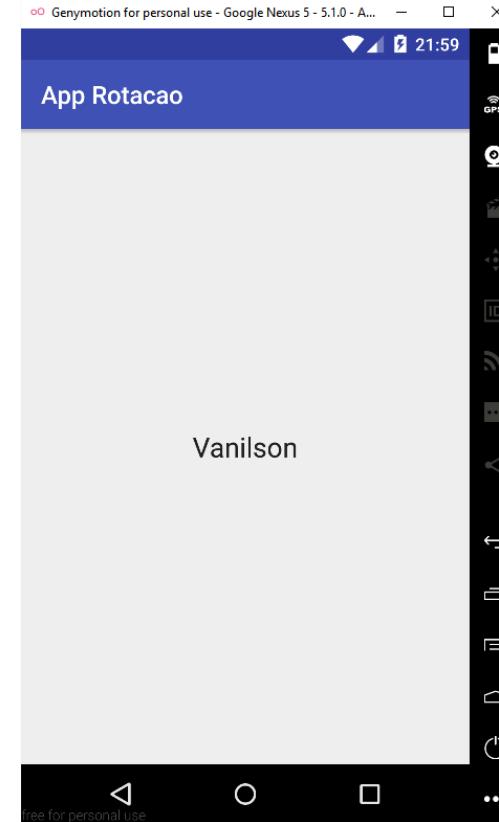
Enviando e obtendo dados

Exercício - Enviando dados para outra activity

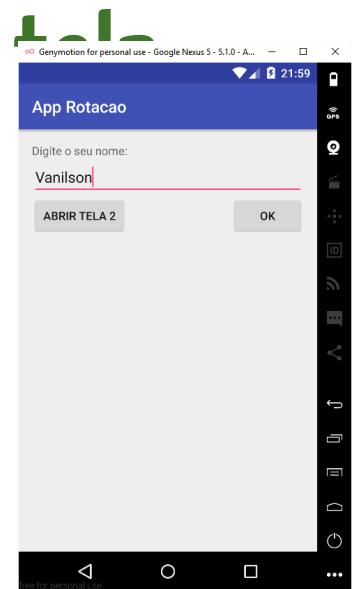
- ▶ Crie essas duas telas e passe como parâmetro o texto digitado na Tela 1



Passar parâmetro para a Tela 2



Exercício – Enviando informação para outra



```
private View.OnClickListener tratadorEvento = (v) -> {
    Intent it = new Intent(Tela1Activity.this, Tela2Activity.class);
    it.putExtra("nomeDigitado", edtNome.getText().toString());
    startActivity(it);
};
```

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_tela2);
    Log.i("Tela2", "onCreate");

    txtView = (TextView) findViewById(R.id.txtNome);
    String nome = getIntent().getStringExtra("nomeDigitado");
    txtView.setText(nome);

}
```

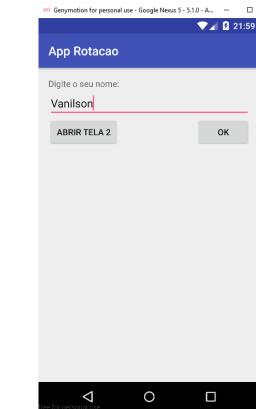
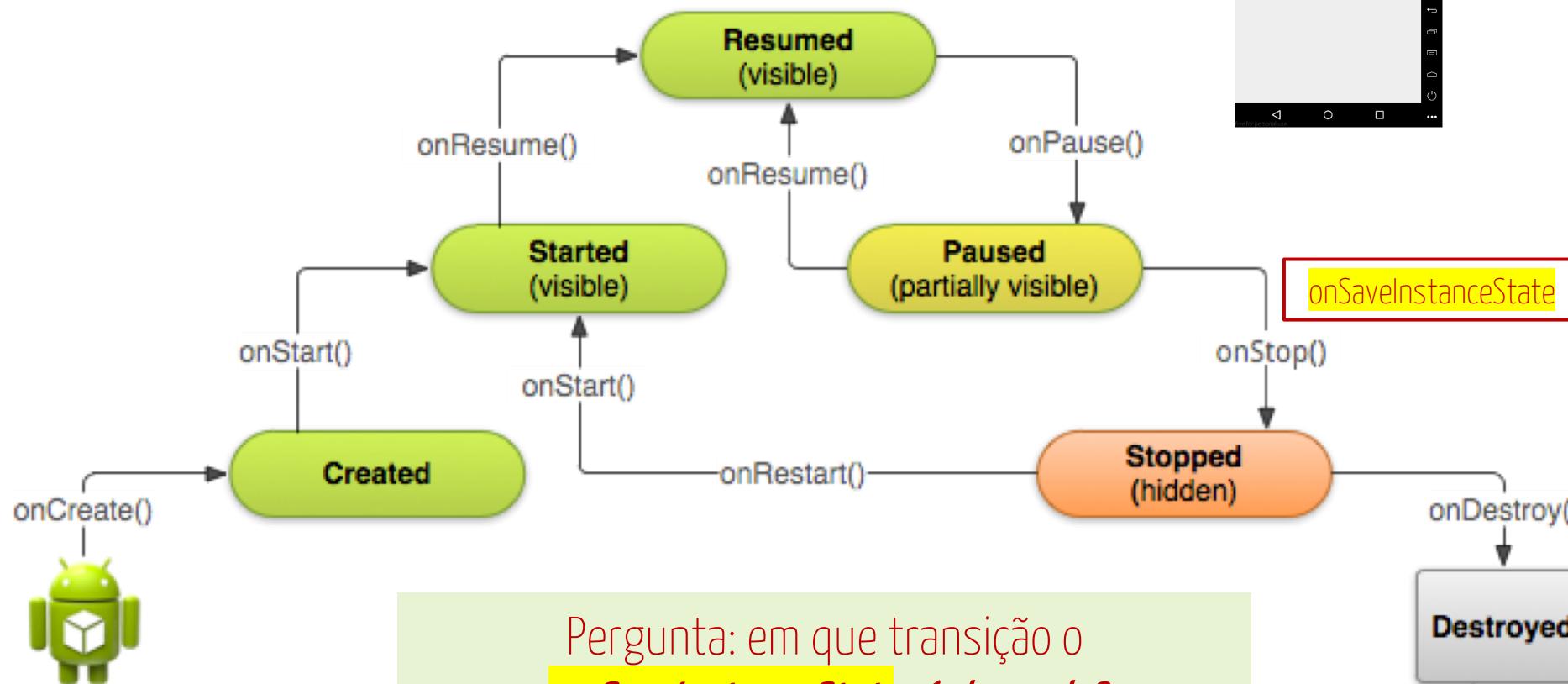
Relembrando o ciclo de vida de uma activity

Logando os métodos do ciclo de vida

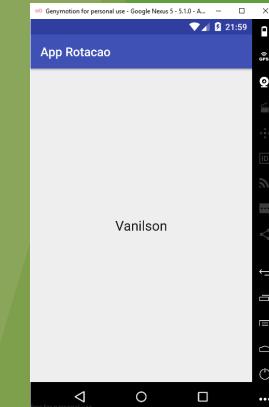
Prof. Vanilson Burégio



Ciclo de vida de uma activity



abre



Ciclo de vida de uma activity

► Quando saber isso é importante?

► Ex.1 – Download de arquivo em process a parte

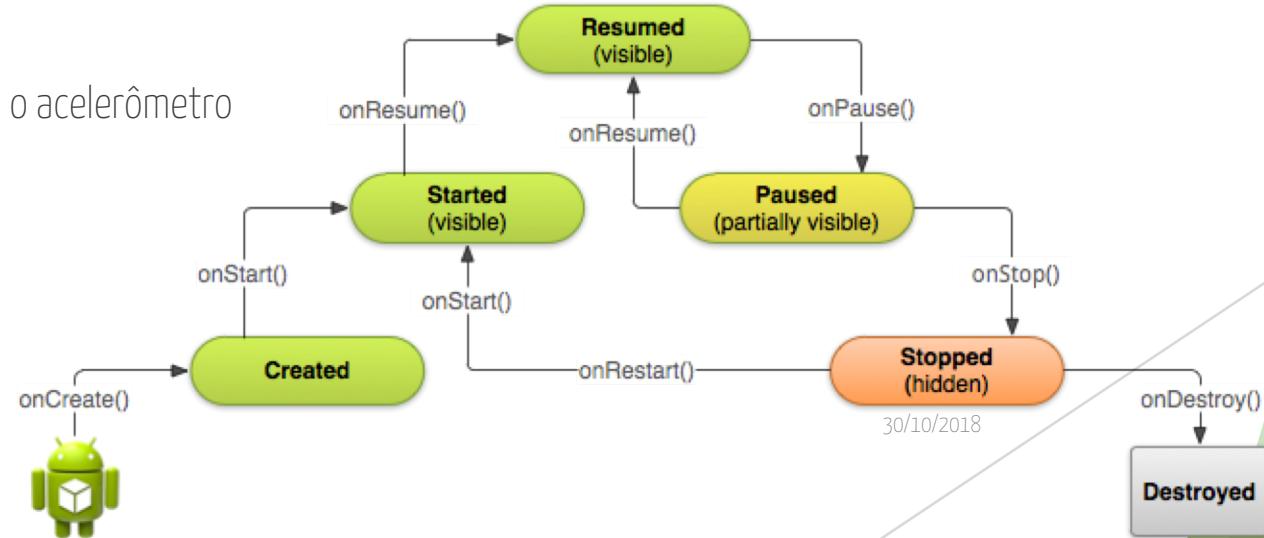
- Se o usuário sair da app, deve “matar” o processo que está baixando o arquivo,
- caso contrário dará erro pois, ao final do download, o processo tentará retornar para uma tela que já não mais existe
- `onPause` ou `onStop` deve ser utilizado nesse caso

► Ex.2 – Localização (GPS consome muita bateria)

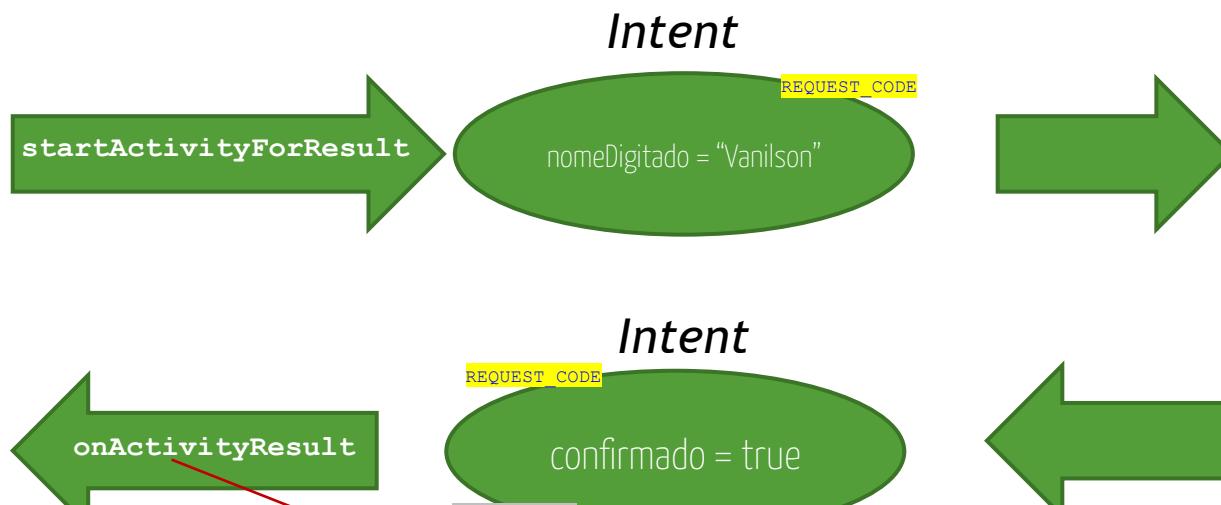
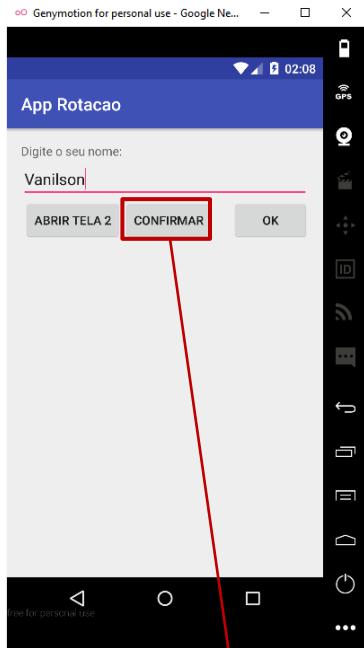
- Ao pausar a activity pare de receber a localização via GPS
- Quando voltar pega a localização novamente

► Ex. 3 – Acelerômetro

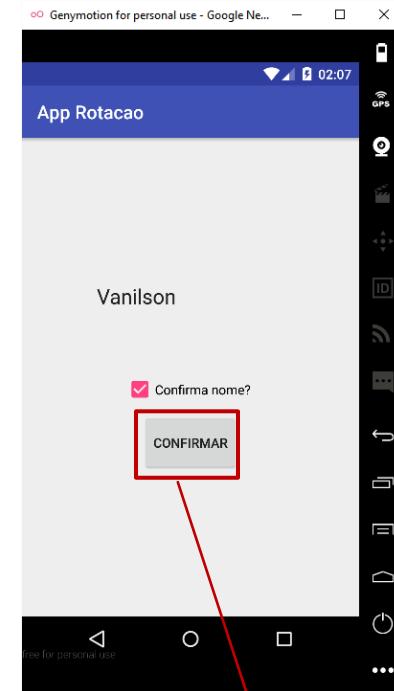
- Saiu da tela, pare de “ouvir” o acelerômetro



Obtendo dados de outra activity



```
Intent it = new Intent(Tela1Activity.this, Tela2Activity.class);
it.putExtra("nomeDigitado", edtNome.getText().toString())
startActivityForResult(it, MEU_REQUEST_CODE);
```



```
Intent it = new Intent();
it.putExtra("confirmado", chkConfirmado.isChecked());
setResult(RESULT_OK, it);
finish();
```

```
protected void onActivityResult(int requestCode, int resultCode, Intent data) { // chamado de volta sempre
    super.onActivityResult(requestCode, resultCode, data);

    if (requestCode == MEU_REQUEST_CODE && resultCode == RESULT_OK) { // poderia ser RESULT_CANCELED
        String nomeConfirmado = data.getBooleanExtra("confirmado", false) ? "Nome confirmado!" : "Nome não confirmado!";
        txtConfirmacao.setText(nomeConfirmado);
    }
}
```

Tratando mudança de orientação

Outras alternativas

Tratando mudança de orientação

Por padrão, quando algumas configurações do aparelho mudam, a **activity** atual é destruída e recriada.

► Opção 1: Forçar uma orientação

- android:screenOrientation

Opção 1 - AndroidManifest.xml

```
<activity android:name=".Tela2Activity"  
         android:screenOrientation="portrait"/>
```

► Opção 2: Tratar manualmente

- Para evitar que o Android destrua a activity ao girar o aparelho, podemos informar
traremos isso manualmente
- onConfigurationChange(Configuration) precisa ser implementado

Opção 2 - AndroidManifest.xml

```
<activity android:name=".Tela2Activity"  
         android:configChanges="orientation|keyboardHidden"/>
```

► Opção 3: Salvar o estado da activity

- **Boa prática**, pois nas opções acima há outros cenários em que o Android destrói a
activity (e.g., mudança de idioma)
- Método onSaveInstanceState(Bundle) chamado imediatamente antes do **onStop()**
- Método onRestoreInstanceState(Bundle) chamado após o **onStart()** e antes do
onResume()

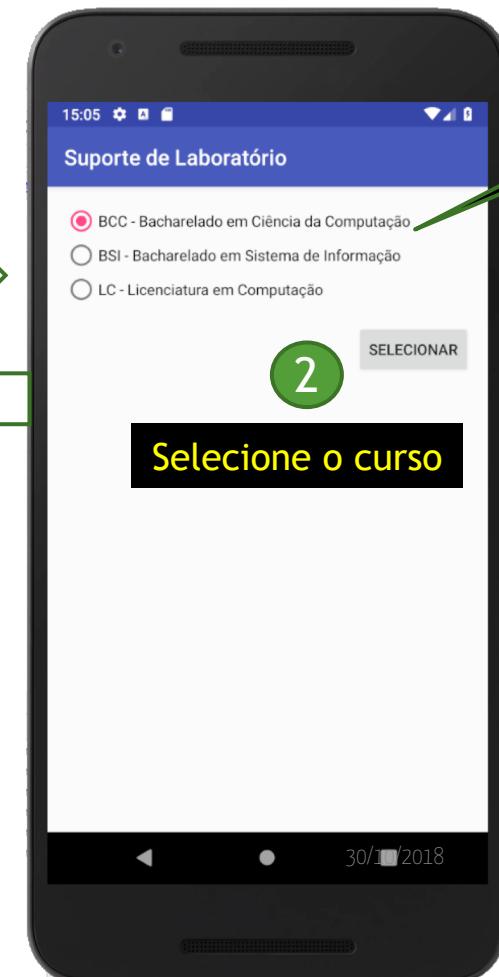
Atividade

Prof. Vanilson Burégio

30/10/2018

Atividade

- ▶ Obtendo dados de uma activity



Utilize Radio Group

Desenvolvimento de aplicações móveis

Interação entre activities

Prof. Vanilson Burégio, PhD

