



PROGRAMAÇÃO
PARA DISPOSITIVOS
MÓVEIS

Ana Karina

Conteúdo da
Aula

Activity

Ciclo de Vida

Exemplo

Programação para Dispositivos Móveis

Aula 4 - Activity

Ana Karina D. Salina de Oliveira

Faculdade de Computação
Universidade Federal de Mato Grosso do Sul

Programação para Dispositivos Móveis
2018



Roteiro

PROGRAMAÇÃO
PARA DISPO-
SITIVOS
MÓVEIS

Ana Karina

Conteúdo da
Aula

Activity

Ciclo de Vida

Exemplo

- 1 Activity
- 2 Ciclo de Vida de uma Activity
- 3 Métodos de retorno de chamada do ciclo de vida
- 4 Exemplo com 2 Activitys

Adaptação do material de Marcio Silva e Lucas Bedogoni

- A classe Activity fornece uma tela com a qual os usuários podem interagir para fazer algo, como:
 - discar um número no telefone,
 - tirar uma foto,
 - enviar um e-mail,
 - ver um mapa.
- é responsável por controlar os eventos da tela,
- herda atributos e métodos da classe `android.app.Activity`,
- deve sobrescrever o método `onCreate(bundle)`,
 - responsável por realizar a inicialização necessária para executar a aplicação,
 - ex: chamar o método `setContentView(view)` para definir a interface do usuário.

- Cada Activity deve obrigatoriamente ser declarada no arquivo *AndroidManifest.xml*
 - O atributo `android:name` é o único necessário — ele especifica o nome da classe da atividade.
 - O elemento `<action>` especifica que esse é o “principal” ponto de entrada do aplicativo.
 - Só uma atividade deve ter a ação "main" e a categoria "launcher".

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.ana.segundatela">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="SegundaTela"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="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=".SegundaTela"></activity>
    </application>

</manifest>
```



Android

Início de uma atividade

PROGRAMAÇÃO
PARA DISPOSITIVOS
MÓVEIS

Ana Karina

Conteúdo da
Aula

Activity

Ciclo de Vida

Exemplo

- Para iniciar outra atividade, é possível chamar `startActivity()` passando uma `Intent`
- a `intent` especifica a atividade exata que deve ser iniciada (que pode ser até de um outro aplicativo).
- Um `intent` também pode portar pequenas quantidades de dados a serem usados pela atividade iniciada.
 - `Intent intent = new Intent(this, SegundaTela.class);`
 - `startActivity(intent);`



Android

Activity

PROGRAMAÇÃO
PARA DISPOSITIVOS
MÓVEIS

Ana Karina

Conteúdo da
Aula

Activity

Ciclo de Vida

Exemplo

Encerramento de uma atividade

- Para encerrar uma atividade

- chame o método `finish()`;
- Também é possível encerrar uma atividade separada iniciada anteriormente chamando `finishActivity()`.

- Uma atividade pode existir essencialmente em três estados:
 - **Resumed:** A atividade está em primeiro plano na tela e tem o foco do usuário (em geral, chama-se esse estado de “em execução”).
 - **Paused:** A atividade ainda está visível, mas outra atividade está em foco (em primeiro plano). outra atividade está visível por cima desta e está parcialmente transparente ou não cobre inteiramente a tela.
 - **Stopped:** A atividade está totalmente suplantada por outra (a atividade passa para "segundo plano"). ela não fica mais visível para o usuário e pode ser eliminada pelo sistema se a memória for necessária em outro processo.

Android

Ciclo de Vida

PROGRAMAÇÃO
PARA DISPOSITIVOS
MÓVEIS

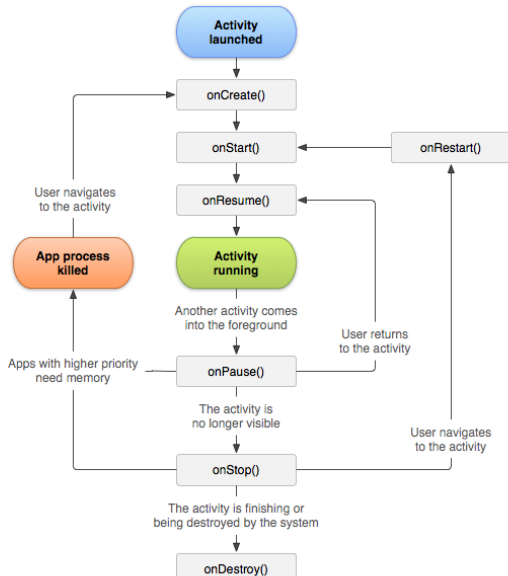
Ana Karina

Conteúdo da
Aula

Activity

Ciclo de Vida

Exemplo



● onCreate()

- Chamado quando a atividade é criada pela primeira vez.
- É onde se deve fazer toda a configuração estática normal — criar exibições, vincular dados a listas etc.
- Esse método recebe um objeto Bundle (pacote) contendo o estado anterior da atividade, caso esse estado tenha sido capturado.
Sempre seguido de onStart().

● onStart()

- Chamado logo antes de a atividade se tornar visível ao usuário.
- Seguido de onResume().

Android

Ciclo de Vida

PROGRAMAÇÃO
PARA DISPOSITIVOS
MÓVEIS

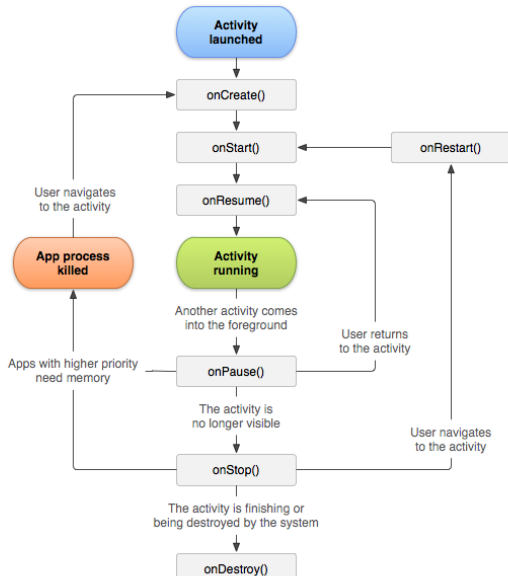
Ana Karina

Conteúdo da
Aula

Activity

Ciclo de Vida

Exemplo





Android

Métodos de retorno de chamada do ciclo de vida

PROGRAMAÇÃO
PARA DISPO-
SITIVOS
MÓVEIS

Ana Karina

Conteúdo da
Aula

Activity

Ciclo de Vida

Exemplo

- **onResume()**
 - Chamado logo antes da Atividade iniciar a interação com o usuário.
 - Nesse ponto, a atividade estará no topo da pilha de atividades com a entrada do usuário direcionada a ela.
 - Sempre seguido de onPause().

Android

Ciclo de Vida

PROGRAMAÇÃO
PARA DISPOSITIVOS
MÓVEIS

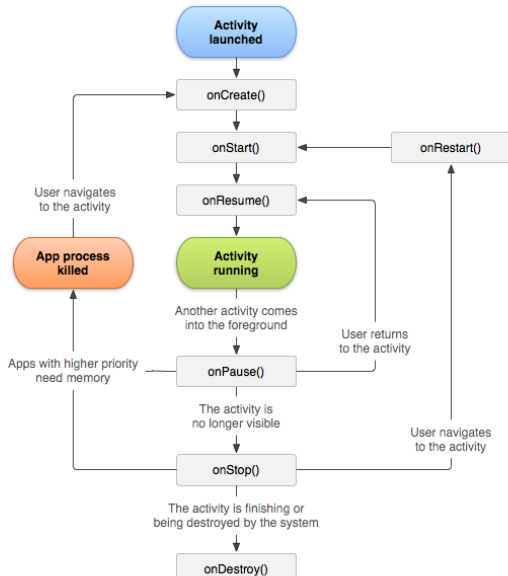
Ana Karina

Conteúdo da
Aula

Activity

Ciclo de Vida

Exemplo





Android

Métodos de retorno de chamada do ciclo de vida

PROGRAMAÇÃO
PARA DISPO-
SITIVOS
MÓVEIS

Ana Karina

Conteúdo da
Aula

Activity

Ciclo de Vida

Exemplo

● onPause()

- Chamado quando o sistema está prestes a retomar outra atividade.
- Esse método normalmente é usado para confirmar alterações não salvas a dados persistentes, animações interrompidas e outras coisas que talvez estejam consumindo CPU.
- Deve fazer tudo bem rapidamente porque a próxima atividade não será retomada até ela retornar.
- Seguido de onResume() se a atividade retornar para a frente ou de onStop() se ficar invisível ao usuário.

- **onStop()**

- Chamado quando a atividade não está mais visível ao usuário.
- Isso pode acontecer porque ela está sendo destruída ou porque outra atividade (uma existente ou uma nova) foi retomada e está cobrindo-a.
- Seguido de `onRestart()` se a atividade estiver voltando a interagir com o usuário ou `onDestroy()` se estiver saindo.

- **onRestart()**

- Chamado depois que atividade tiver sido interrompida, logo antes de ser reiniciada.
- Sempre seguido de `onStart()`.

● **onDestroy()**

- Chamado antes de a atividade ser destruída.
- É a última chamada que a atividade receberá.
- Pode ser chamado porque a atividade está finalizando (alguém chamou finish()) ou porque o sistema está destruindo temporariamente essa instância da atividade para poupar espaço.
- É possível distinguir entre essas duas situações com o método isFinishing().

<https://developer.android.com/guide/components/activities.html?hl=pt-br>

Activities podem utilizar Bundle(Pacotes)

- Bundle(Pacotes)
 - são objetos utilizados entre atividades com intenções passar informações entre componentes.
 - representa um conjunto de pares "chave/valor".
 - permite guardar valores de qualquer tipo primitivo.
- Os objetos Bundle devem ser usados:
 - pelos processos com transações IPC,
 - entre atividades com intents
 - e para armazenar o estado transitório nas alterações de configuração.

Activities podem utilizar Extras

- Pares de valor-chave que carregam informações adicionais necessárias para realizar a ação solicitada.
- Algumas ações usam determinados tipos de URIs de dados e outras usam determinados extras.
- É possível adicionar dados extras com diversos métodos *putExtra()*, cada um aceitando dois parâmetros: o nome principal e o valor.
- Também é possível criar um objeto Bundle com todos os dados extras e, em seguida, inserir o Bundle na Intent com *putExtras()*.



Parcelables and Bundles (Parcelables e pacotes)

Exemplo: Main

```
String nome = edtNome.getText().toString();  
Intent it = new Intent(this, SegundaTela.class);  
Bundle param = new Bundle();  
param.putString("chave_nome", nome);  
it.putExtras(param);  
startActivity(it);
```

Exemplo: Segunda Tela

```
Bundle param = itRecebedora.getExtras();  
if(param != null){  
    String nome = param.getString("chave_nome");  
    txtNome.setText(nome);  
}
```

Criando Duas Activities

- Main Activity - Dados do aluno
 - Nome do aluno
 - Curso
 - Disciplina
- Segunda Activity
 - Traz o nome do aluno da primeira Activity
 - Recebe
 - porcentagem de presenças
 - Prova 1
 - Prova 2
 - Trabalho
 - Calcula
 - Média
 - Resultado

Segunda Activity - Calculando média e apresentando resultado ao aluno

- se (porcentagem de presenças $< 75\%$)
 - Resultado: aluno reprovado
- senão
 - se $((\text{Prova1} + \text{Prova2} + \text{Trabalho}) / 3 \geq 7)$
 - Resultado: aluno aprovado
 - senão
 - Resultado: aluno reprovado