Universidade Estadual do Maranhão
Engenharia da Computação

Programação de Dispositivos Móveis

Thiago Nunes de Sousa
thiagonunes.tns@gmail.com

Componentes

- **Blocos básicos para construção de um aplicativo Android**
- **Cada tipo de componente desempenha uma função específica**
- **Cada componente tem seu próprio ciclo de vida**

Componentes

- **Atividades**
  - Activity
- **Serviços**
  - Service
- **Provedores de Conteúdo**
  - Content Providers
- **Receptores de Transmissão**
  - Broadcast Receivers

Activity

- **Representa uma tela única com uma interface do usuário**
  - Uma tela de envio de e-mail ou para navegar na internet ou visualizar uma imagem, etc.
- **Geralmente uma app é composta por várias activities**
- **A activity principal é identificada no arquivo manifesto e chamada quando a app é aberta**

Service

- **Componentes executados em segundo plano**
  - Operações longas ou trabalhos remotos
- **Não possui interface com o usuário**
  - Um serviço pode tocar uma música em segundo plano ou fazer um download de um arquivo enquanto o usuário utiliza outro app
- **Serviços são startados por outros componentes (ex.: activity)**

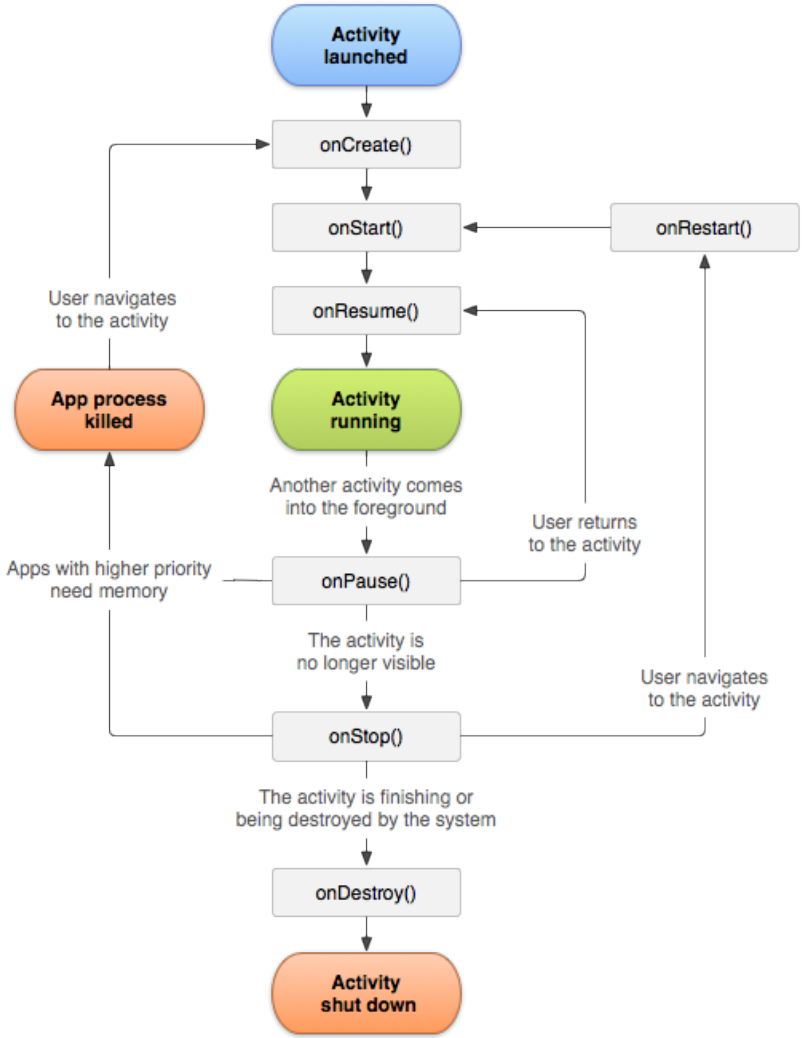
Content Providers

- **Gerenciam dados compartilhados**
- **Um aplicativo pode fornecer acesso à seus dados a terceiros utilizando este componente**
  - Ex.: Aplicativo de contatos
- **Outros apps**
  - Consultar ou alterar dados de terceiros
  - Desde que tenha permissão

Broadcast Receivers

- **Ficam “escutando” mensagens de todo o sistema**
  - Ex.: transmissão de uma mensagem informando que a bateria está baixa ou que foi tirado um print da tela ou que o download de um arquivo chegou ao fim
- **Não possuem interface com o usuário mas podem criar notificações na barra de status**

Activity


Activity launched
onCreate()
onStart()
onRestart()
onResume()
User navigates to the activity
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

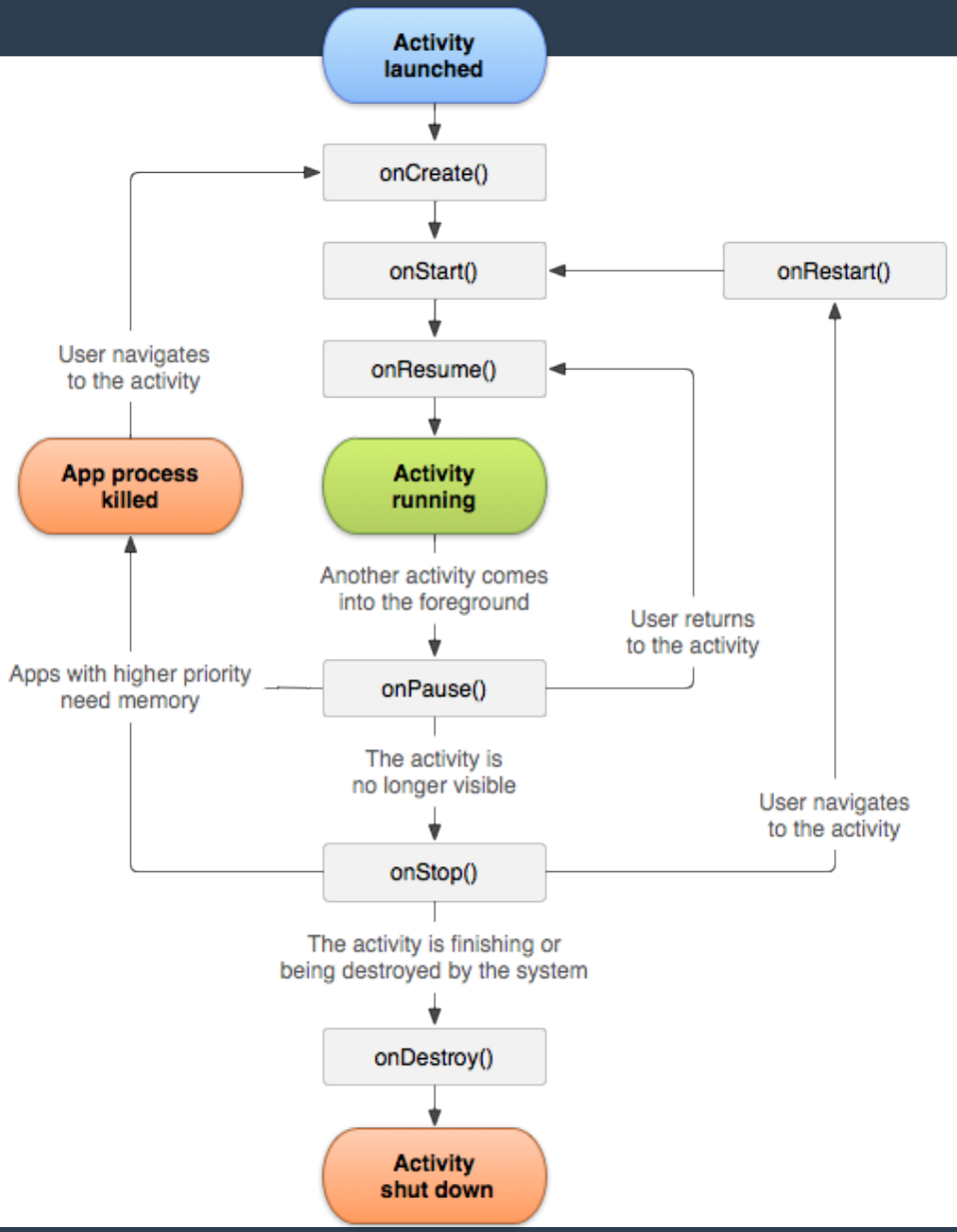
Activity

- **Tela**
- **Atividade Principal**
  - Iniciada junto com o app, pode iniciar outras
- **Pilha de Atividades**
  - Ativ1
  - Ativ1 → Ativ2
  - Ativ1 → Ativ2 → Ativ3
  - Ativ1 → Ativ2
  - Ativ1

Estados de uma Activity

- **Resumed**
  - A atividade está em primeiro plano na tela e tem o foco do usuário
- **Paused**
  - A atividade ainda está visível, mas outra atividade está em primeiro plano e tem o foco. Ou seja, 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"). Uma atividade interrompida ainda está ativa (o objeto Activity está retido na memória, mantém todas as informações de estado e do membro, mas não está anexado ao gerenciador de janelas). No entanto, ela não fica mais visível para o usuário e pode ser eliminada pelo sistema se a memória for necessária em outro processo.

Ciclo de Vida



Ciclo de Vida

```java
public class ExampleActivity extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        // The activity is being created.
    }
    @Override
    protected void onStart() {
        super.onStart();
        // The activity is about to become visible.
    }
    @Override
    protected void onResume() {
        super.onResume();
        // The activity has become visible (it is now "resumed").
    }
    @Override
    protected void onPause() {
        super.onPause();
        // Another activity is taking focus (this activity is about to be "paused").
    }
    @Override
    protected void onStop() {
        super.onStop();
        // The activity is no longer visible (it is now "stopped")
    }
    @Override
    protected void onDestroy() {
        super.onDestroy();
        // The activity is about to be destroyed.
    }
}
```

Intent

- **Objeto de mensagem**
- **Utilizado para solicitar uma ação de outro componente**
  - Iniciar uma Activity
  - Iniciar um Service
  - “Disparar” uma transmissão

Ciclo de Vida

- **Projeto Novo**
- **Testar Ciclo de Vida das Activities**
- **Aprender a iniciar e destruir uma activity**

Enviando dados entre activities

- **Activity 1**

```
Intent intent = new Intent(this, OutraActivity.class);
intent.putExtra("CHAVE", “valor”);
startActivity(intent);
```

- **Activity 2**

```
Bundle extras = getIntent().getExtras();
if (extras != null) {
    String value = extras.getString("CHAVE");
}
```

Exemplo

- **Criar uma activity com um um campo de texto e um botão**
- **Ao clicar no botão, capturar o valor digitado e enviar para a segunda activity**
- **Na segunda activity, receber o texto digitado e exibir na tela**

Esperar por resultado

- **Activity1**

```
Intent i = new Intent(this, OutraActivity.class);
startActivityForResult(i, NUM_RETORNO);
```

- **Activity 2**

```
Intent i = new Intent();
i.putExtra("resposta", “Minha Resposta”);
setResult(Activity.RESULT_OK, i);
finish();
```

Esperar por resultado

- **Activity1 (Tratar Resposta)**

```
void onActivityResult(int requestCode, int resultCode, Intent data){
    if (requestCode == NUM_RETORNO) {
        if(resultCode == Activity.RESULT_OK){
            String resposta=data.getStringExtra("resposta");
        }
        if (resultCode == Activity.RESULT_CANCELED) {
            String resposta=”Cancelado pelo usuário”;
        }
    }
}
```

Exemplo

- **Criar duas activities**
- **A primeira deve ter um botão que ao ser clicado inicia a segunda e fica aguardando por uma resposta. Ao receber a resposta, exibir na tela.**
- **A segunda deve possuir um campo de texto e dois botões (Cancelar e Enviar). Ao clicar nestes botões retornar para a activity anterior**

Gravar Estado

- **onSaveInstanceState**
  - Chamado imediatamente antes do sistema destruir uma activity
  - Utilizado para salvar o estado atual da activity
    - Para ser utilizado em casos onde próprio Android não consiga recuperar a Activity
      - Ex.: Mudança de Configuração do Dispositivo, mudar orientação da tela e etc, ações que necessitam a recriação da activity
    - Valores de Views são salvos pelo android (?)
  - Salva os dados no formato chave-valor em um objeto da classe Bundle

Gravar Estado

```
public void onSaveInstanceState(Bundle savedInstanceState) {
    savedInstanceState.putInt(“Chave”, VALOR);
    super.onSaveInstanceState(savedInstanceState);
}

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    if (savedInstanceState != null) {
        meuValor = savedInstanceState.getInt(“Chave”);
    }
    (...)
}
```

Boa Práticas

- **Não coloque códigos pesados no onCreate. Sua App irá demorar a exibir a UI, parecendo que ela está travada.**
- **Utilize onStart e onResume para inicialização e configuração**
- **Desaloque recursos no método onDestroy (Ex.: conexão a BD)**

Universidade Estadual do Maranhão
Engenharia da Computação

Programação de Dispositivos Móveis

Thiago Nunes de Sousa
thiagonunes.tns@gmail.com