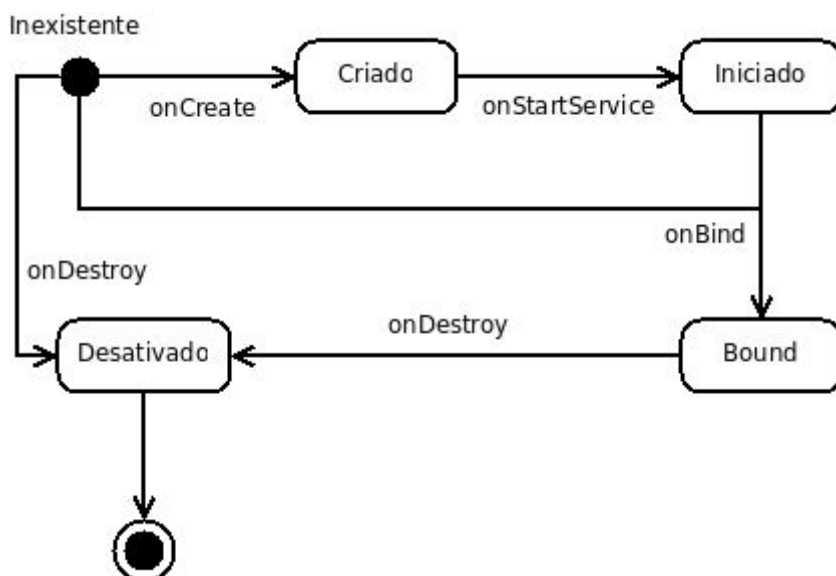


Aula Prática 3

Service

Um serviço, seja em qual plataforma for, é uma aplicação que executa tarefas em segundo plano, ou em "background". A plataforma Android já tem em sua API um modelo de componente específico para criar serviços.

Seu ciclo de vida e estados é ligeiramente diferente de uma Activity:



Quando o serviço não está sendo executado e alguém solicita sua criação (veremos como fazer isto mais adiante), o método "onCreate()" é invocado. Nele, você pode fazer inicializações. Quando o serviço é iniciado, o método "onStartService()" é invocado. Ele ficará executando suas atividades até que algum cliente (da mesma aplicação ou de outra), tente se conectar (bind) a ele, neste caso, o método "onBind()" é invocado. Sempre que alguém solicitar seu término, o método "onDestroy()" é invocado.

A classe que representa o serviço deve ser uma subclasse de `android.app.Service` e deve obrigatoriamente implementar o método `IBinder onBind(intent)`, e se necessário métodos para controlar o ciclo de vida do Serviço, como `onCreate()`, `onStart()` e `onDestroy()`.

Atividade 1:

Passo 1:

Crie um projeto que servirá como exemplo para seu primeiro serviço criado no Android.

Passo 2:

No projeto que foi criado, se ainda não existir, crie um novo layout com dois botões e uma Activity que permita controlar os elementos de View, conforme mostrado na imagem abaixo:



Passo 3:

Crie uma nova classe chamada `MyService` no projeto que servirá para controlar o processo que será executado via background (Serviço).

Passo 4:

As ações dos métodos `onClick` de cada botão deverão fazer as seguintes chamadas, respectivamente:

Botão Iniciar Serviço:

```
startService(new Intent(this, MyService.class));
```

Botão Parar Serviço:

```
stopService(new Intent(this, MyService.class));
```

Passo 5:

Na nova classe criada (Service) implemente o código abaixo:

```
public class MyService extends Service implements Runnable{

    private static final String TAG = "Servico";
    private boolean active;
    private int count;
    private static final int MAX = 50;

    @Override
    public IBinder onBind(Intent arg0) {
        return null;
    }

    @Override
    public void onCreate() {

        Toast.makeText(this, "Serviço Criado!", Toast.LENGTH_LONG).show();
        Log.d(TAG, "Método onCreate");
        active = true;
        new Thread(this).start();
    }
}
```

```

@Override
public void onStart(Intent intent, int startId) {
    Toast.makeText(this, "Serviço Iniciado!", Toast.LENGTH_LONG).show();
    Log.d(TAG, "Método onStart");
}

@Override
public void onDestroy() {
    Toast.makeText(this, "Serviço Finalizado!", Toast.LENGTH_LONG).show();
    Log.d(TAG, "Método onDestroy");
    active = false;
}

@Override
public void run() {
    while (active && count < MAX)
    {
        executeThread();
        Log.d(TAG, "Executando..." + count);
        count++;
    }
    Log.d(TAG, "FIM!!!");
    stopSelf();
}

private void executeThread() {
    try {
        Thread.sleep(1000);
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
}
}
}

```

No método run(), quando o valor do contador chega a 50, o loop da thread termina e o método stopSelf() é chamado, o que encerra o ciclo de vida do serviço, fazendo com que o próprio Android chame o método onDestroy, encerrando o processo para liberar memória e recursos utilizados.

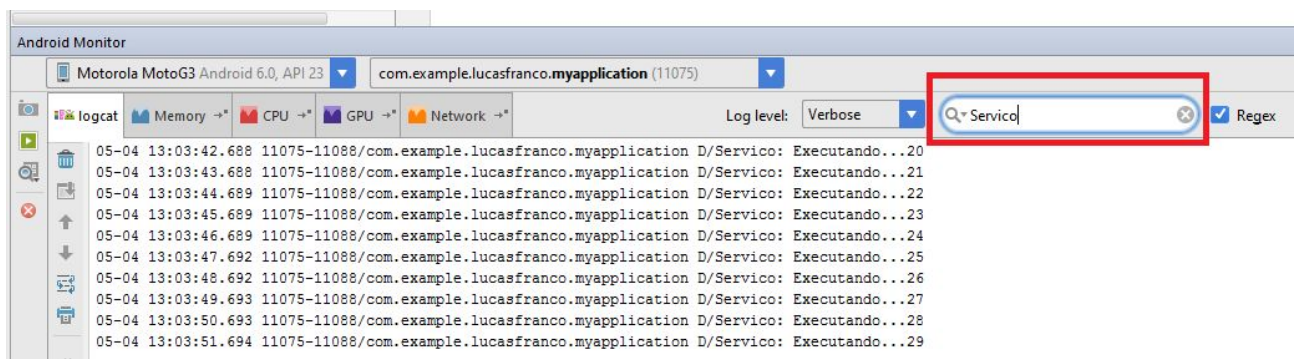
Passo 6:

Inclua no AndroidManifest o trecho que faz associação ao novo serviço utilizado:

```
<service android:enabled="true" android:name=".MyService" />
```

Passo 7:

Execute e teste sua aplicação. Para isso, utilize um filtro com a TAG utilizada no código ("Serviço") no LogCat para ver o processo funcionar:



A execução só será interrompida quando o loop chegar ao valor 50, ou se você entrar na aplicação e clicar no botão "Parar Serviço".

Atividade 2:

Altere o código acima para que a execução do Serviço grave em um arquivo local de texto (TXT), no Android, uma sequência numeral de 1 até 100000. Para a realização desta atividade utilize o método abaixo como exemplo, que permite a criação e gravação de arquivos textos no Android:

```
private void gravarDados(String texto) {

    String lstrNomeArq;
    File arq;
    byte[] dados;

    try {

        arq = new File(Environment.getExternalStorageDirectory(), "arqAndroid.txt");
        FileOutputStream fos;

        dados = texto.toString().getBytes();

        fos = new FileOutputStream(arq);
        fos.write(dados);
        fos.flush();
        fos.close();
        Log.d(TAG, "TEXTO SALVO COM SUCESSO!!!");

    } catch (Exception e) {
        Log.d(TAG, "FIM DA GRAVACAO!!!");
    }

}
```

Inclua no AndroidManifest o trecho que permite a gravação de arquivo texto em dispositivo de armazenamento externo:

```
<uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
```