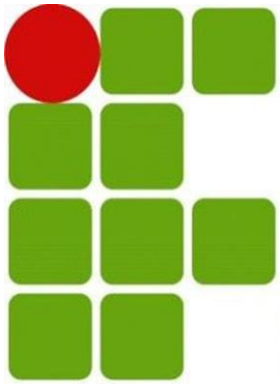


INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
MARANHÃO

**Prof. MSc. Flávio Barros**
*flavioifma@gmail.com*
**www.flaviobarros.com.br**

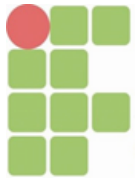
# Programação para Dispositivos Móveis

## Aula - Service

Caxias - MA

# Roteiro

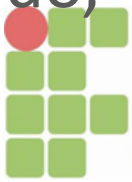
## Service



# Service

## Fundamentos

- Componente responsável por rodar tarefas em background;
- São tarefas mais longas, rodadas em background para não prejudicar a responsividade da aplicação;
- Um componente poderá se vincular a um serviço para interagir com ele e até estabelecer comunicação entre processos (IPC); Ex. um serviço pode lidar com transações de rede, reproduzir música, executar E/S de arquivos, ou interagir com um provedor de conteúdo, tudo a partir do segundo plano.

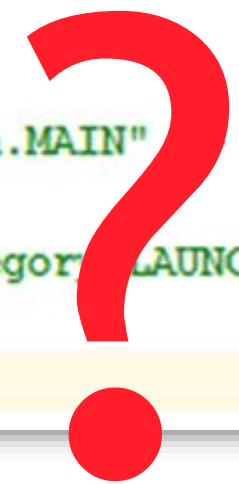


# Service

## Configuração no AndroidManifest.xml

- Para criar um serviço é preciso declarar o nome da classe no Manifest;
- O serviço pode ser utilizado por qualquer aplicação através de um Intent.

```
<activity android:name=".MainActivity">
    <intent-filter>
        <action android:name="android.intent.action.MAIN"

        <category android:name="android.intent.categor LAUNCHER" />
    </intent-filter>
</activity>
```

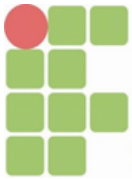


# Service

## Configuração no AndroidManifest.xml

- Para criar um serviço é preciso declarar o nome da classe no Manifest;

```
<application ... >
  <service android:name="ServiceName">
    <intent-filter>
      <action android:name="com.example.serviceapp.ACTION" />
      <category android:name="android.intent.category.DEFAULT"/>
    </intent-filter>
  </service>
</application>
```



# Service

## Manifesto

- É necessário explicitar que o serviço é privado no Manifest, caso o serviço seja útil apenas para a aplicação que o contém.

```
<application ... >
  <service android:name="ServiceName">
    android:exported ="false "
    <intent-filter>
      <action android:name="com.example.serviceapp.ACTION" />
      <category android:name="android.intent.category.DEFAULT"/>
    </intent-filter>
  </service>
</application>
```

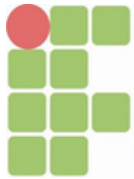
Para iniciar o serviço de outra aplicação sem um **<intent-filter>** deve-se usar **android:exported = "true"**

# Service

## Formas de iniciar um Serviço

- Um serviço pode essencialmente ter duas formas:
  - Iniciado
  - Vinculado

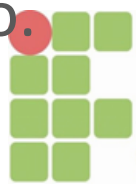
```
startService (Intent)

bindService (Intent, ServiceConnection, flags)
```



# Service

## Iniciado

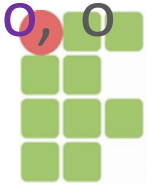
- Um serviço é "iniciado" quando um componente do aplicativo (como um atividade) inicia-o chamando startService();
- Quando iniciado, um serviço pode ficar em execução em segundo plano indefinidamente, mesmo que o componente que o iniciou, seja destruído;
- Um serviço iniciado realiza uma única operação e não retorna um resultado para o autor da chamada; Ex. Fazer download ou upload de um arquivo pela rede; Quando a operação for concluída, o serviço deverá ser interrompido.



# Service

## Vinculado

- Um serviço é "vinculado" quando um componente do aplicativo chama bindService() para vinculá-lo;
- Este tipo de serviço oferece uma interface servidor-cliente que permite que os componentes interajam com a interface, enviem solicitações, obtenham resultados, mesmo em processos com comunicação interprocessual (IPC);
- Um serviço vinculado permanece em execução somente enquanto outro componente do aplicativo estiver vinculado a ele; Vários componentes podem ser vinculados ao serviço de uma só vez, mas quando todos desfizerem o vínculo, o serviço será destruído.

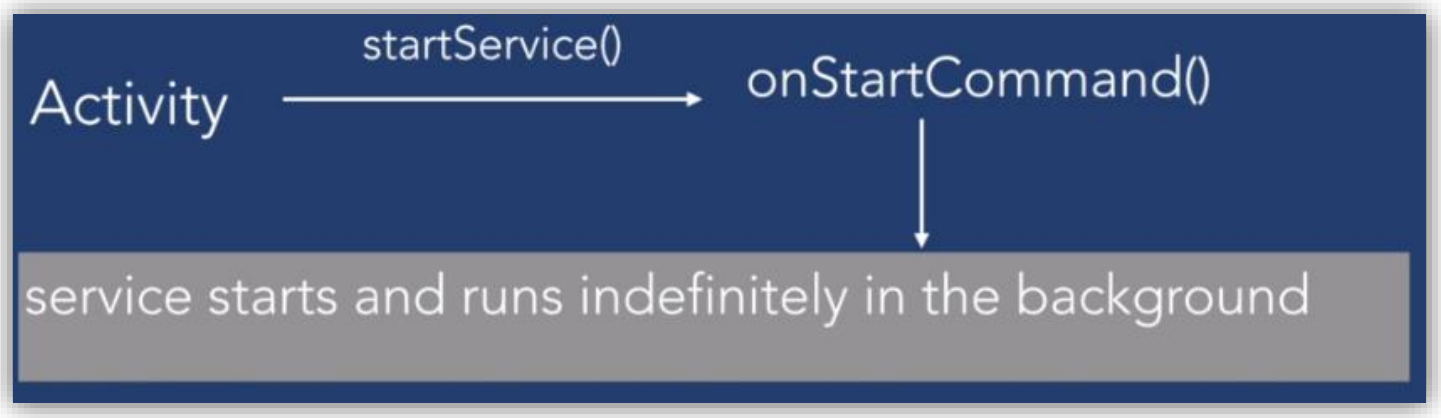


# Classe Service

Para se criar um serviço é preciso implementar uma extensão da classe Service e sobrescrever alguns métodos de callback.

## onStartCommand()

- Inicia um serviço indefinidamente; O serviço apenas termina quando o método stopSelf() é executado a partir do próprio serviço ou quando o método stop-Service() é executado a partir de outra aplicação.



# Classe Service

## onBind()

- Chamado pelo sistema para associar o serviço a uma aplicação; Deve prover uma interface de comunicação entre ambos; Deve ser implementado; Se o serviço não for projetado para suportar Bind então o método onBind deve devolver null.

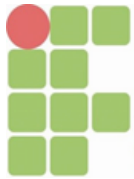
# Classe Service

## onCreate()

- Chamado pelo sistema no momento da criação do serviço e pode ser utilizado para realizar pré configurações.

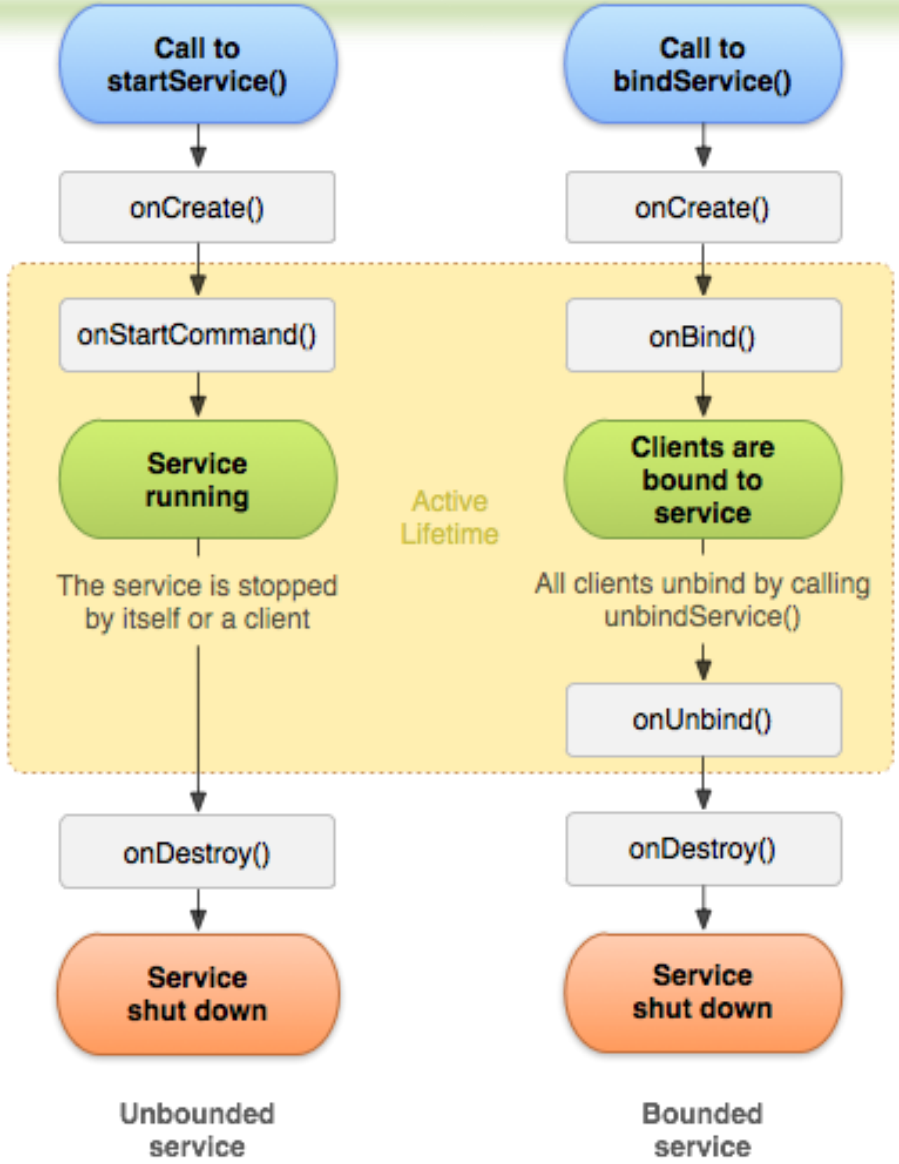
## onDestroy()

- Chamado pelo sistema quando o serviço for destruído e pode ser utilizado para liberar recursos utilizados.



# Service

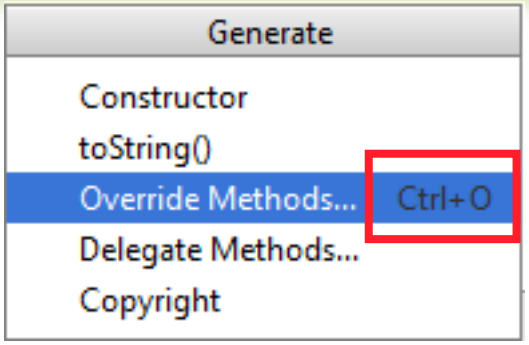
https://developer.android.com/guide/components/services.html?hl=pt-br

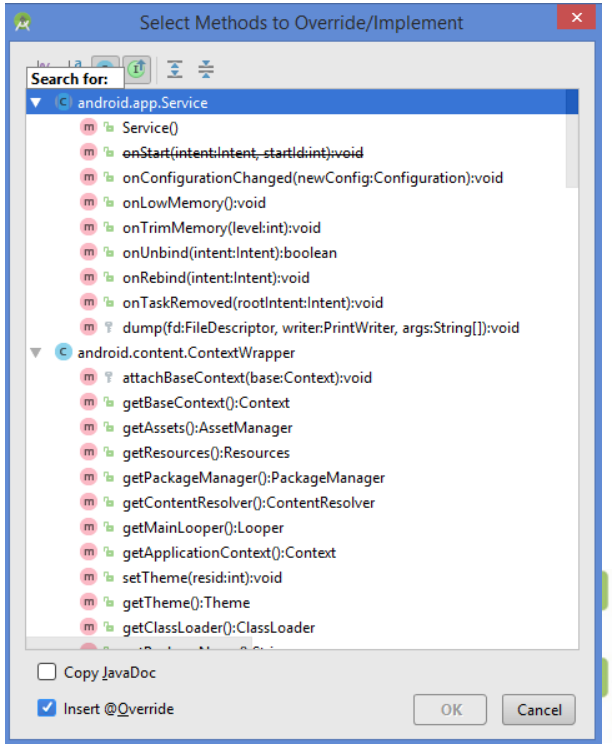


# Service

```
import android.app.Service;
import android.content.Intent;
import android.os.IBinder;
import android.support.annotation.Nullable;

public class ExService extends Service {

    @Override
    public void onCreate() {
        // Método excutado no momento em que o serviço é criado
        super.onCreate();
    }
    @Override
    public int onStartCommand(Intent intent, int flags, int startId) {
        // Executa o Serviço
        return super.onStartCommand(intent, flags, startId);
    }
    @Nullable
    @Override
    public IBinder onBind(Intent intent) {
        // Sem suporte a Binding
        return null;
    }
    @Override
    public void onDestroy() {
        super.onDestroy();
        // Método executado no momento em que o Serviço é Destruido
    }
}
```

**ALT + INSERT**



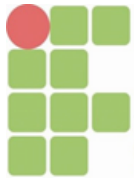
# Classe Service

- O método onStartCommand() devolve um inteiro;
- Este valor indica como o sistema deve continuar o serviço caso o sistema o destrua;
- Existem 3 valores possíveis:

START_NOT_STICKY

START_STICKY

START_REDELIVER_INTENT



# Classe Service

## START_NOT_STICKY

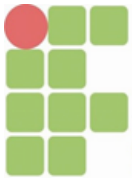
- Não reinicia o serviço a menos que hajam Intents a serem entregues;

## START_STICKY

- Reinicia o serviço mas não continua a partir do Intent que estava em execução mas apenas para os que estavam pendentes;

## START_REDELIVER_INTENT

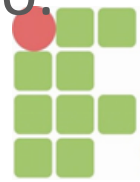
- Reinicia o serviço retomando a partir do Intent que estava em execução.



# Classe Service

## Método **startService** (Intent)

- Inicia o serviço determinado pela intente;
- O serviço roda independente do chamador e por tempo indeterminado;
- Se o contexto chamador terminar, o serviço continua rodando normalmente;
- Pode ser chamado inúmeras vezes:
  - Apenas na primeira o callback onCreate() é chamado;
  - O método onStart/onStartCommand sempre é chamado.



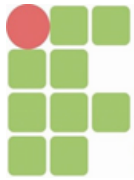
# Classe Service

## Método **stopService** (Intent)

- Termina o serviço chamando onDestroy();
- Não importa quantas vezes startService() foi chamado, uma única chamada a este encerra o serviço.

## Método stopSelf()

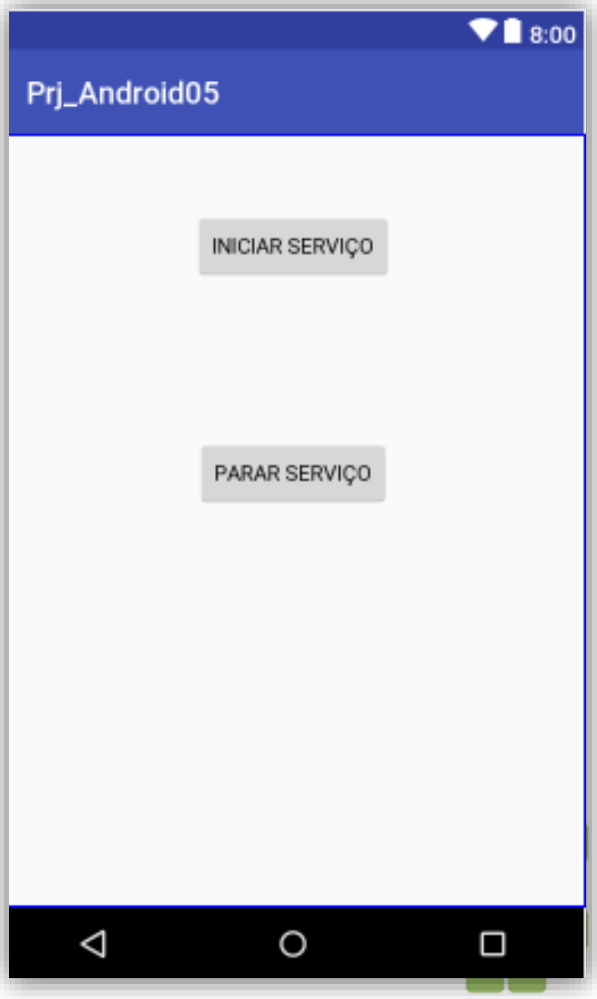
- Chamar quando serviço terminar o processamento;
- Boa prática para evitar consumo de bateria.



# Roteiro

# Prj_Android05

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="20dp">
    <Button
        android:id="@+id/btn_start"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Iniciar Serviço"
        android:layout_marginTop="50dp"
        android:layout_centerHorizontal="true" />
    <Button
        android:id="@+id/btn_stop"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Parar Serviço"
        android:layout_marginTop="200dp"
        android:layout_centerHorizontal="true" />
</RelativeLayout>
```

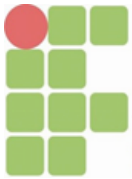



activity_main.xml

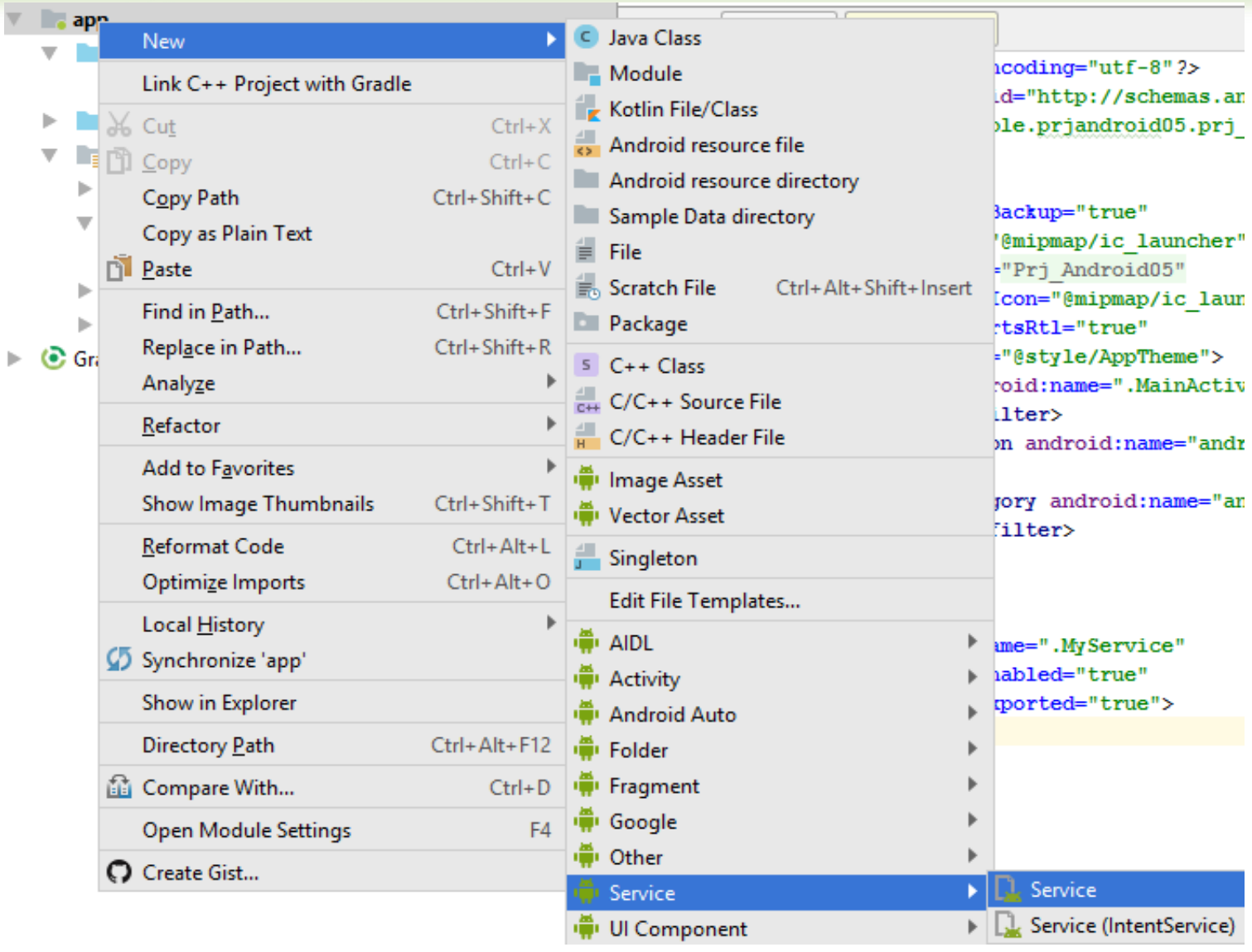
# Prj_Android05

MainActivity.java

```
public class MainActivity extends AppCompatActivity {
    private Button btn_start;
    private Button btn_stop;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        btn_start = (Button)findViewById(R.id.btn_start);
        btn_stop = (Button)findViewById(R.id.btn_stop);

        btn_start.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                // INICIAR SERVIÇO
                Intent intent = new Intent(getApplicationContext(), MyService.class);
                startService(intent);
            }
        });
        btn_stop.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                // FINALIZAR SERVIÇO
                Intent intent = new Intent(getApplicationContext(), MyService.class);
                stopService(intent);
            }
        });
    }
}
```



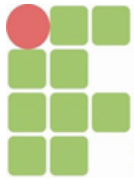
# Prj_Android05



# Prj_Android05

MyService.java

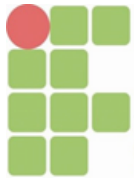
```
public class MyService extends Service {
    public MyService() {
    }
    @Override
    public IBinder onBind(Intent intent) {
        // TODO: Return the communication channel to the service.
        throw new UnsupportedOperationException("Not yet implemented");
    }
    @Override
    public void onCreate() {
        // Método executado no momento em que o serviço é criado
        Log.i( tag: "Log",  msg: "Criando Serviço!");
        super.onCreate();
    }
    @Override
    public int onStartCommand(Intent intent, int flags, int startId) {
        // Executa o Serviço
        Log.i( tag: "Log",  msg: "Executando Serviço!");
        return super.onStartCommand(intent, flags, startId);
    }
    @Override
    public void onDestroy() {
        // Método executado no momento em que o Serviço é Destruido
        Log.i( tag: "Log",  msg: "Finalizando Serviço!");
        super.onDestroy();
    }
}
```



# Prj_Android05

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.prjandroid05.prj_android05">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="Prj_Android05"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="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>

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

</manifest>
```

AndroidManifest.xml

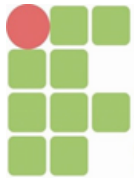


# Roteiro

# Praticando

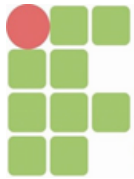
## Prj_Android_Extra04

- Desenvolver aplicativo Android que possua um serviço, onde este, ao clicar no botão tocar, deverá tocar uma música, e quando clicar no botão parar, deverá encerrar a música.



# Roteiro

# Referências



# Referências


- DEITEL, P.; DEITEL, H.; DEITEL, A. **Android 6 - Para Programadores - Uma Abordagem Baseada em Aplicativos**. 2.ed. Bookman, 2015.
- CORDEIRO, Fillipe. **Começando com Android Studio: o guia passo a passo.**1.ed.,1998.
- MONTEIRO, J. **Google Android - Crie Aplicações para Celulares e Tablets**. Editora Casa do Código, 2013.
- MAIA, Luís F. **Programação para dispositivos móveis**. IFMA/ Caxias. 2017.
- BACALÁ JR, Sílvio. **Computação Móvel**. FACOM – UFU.
- Site oficial "**Android Developer**". Disponível em: <http://developer.android.com/index.html>.