

# Desenvolvimento de Apps para Mobile

Fabrício Tonetto Londero

Aula 10 – Notification Manager



# Notification Manager

- é o sistema que o Android usa para exibir alertas visuais e sonoros para o usuário, informando sobre eventos importantes que aconteceram no seu aplicativo ou no sistema.

- Vamos criar um serviço:

```
public static class BackgroundService extends Service { }
```

- Adicionar o IBinder

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



# PendingIntent

- Um PendingIntent é um token que você pode fornecer a outro aplicativo ou ao sistema Android para executar uma ação em seu nome. É frequentemente usado em cenários em que você deseja delegar uma ação para ser executada no futuro, mesmo que seu aplicativo não esteja em execução.



```
private void showNotification() {  
    Intent intent = new Intent(this, MainActivity.class);  
    PendingIntent pendingIntent = PendingIntent.getActivity(this, 0, intent, FLAG_MUTABLE);  
  
    Notification notification = new NotificationCompat.Builder(this, "default")  
        .setContentTitle("Notificação de Evento")  
        .setContentText("Algo aconteceu!")  
        .setSmallIcon(R.drawable.person_pin_circle_24px)  
        .setContentIntent(pendingIntent)  
        .build();  
  
    NotificationManager notificationManager = (NotificationManager)  
    getSystemService(Context.NOTIFICATION_SERVICE);  
    notificationManager.notify(1, notification);  
}
```



- O sinalizador `FLAG_IMMUTABLE` indica que o `PendingIntent` não pode ser modificado após sua criação, enquanto o sinalizador `FLAG_MUTABLE` indica que o `PendingIntent` pode ser modificado após sua criação. No seu caso, você deve usar o sinalizador `FLAG_IMMUTABLE` ao criar o `PendingIntent`.



```
@Override
public int onStartCommand(Intent intent, int flags, int startId) {
    new Thread(new Runnable() {
        @Override
        public void run() {
            try {
                Thread.sleep(5000);
                showNotification();
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }).start();

    return START_STICKY;
}
```

# OnCreate de uma Activity ou em um click de botão

```
Intent serviceIntent = new Intent(this, BackgroundService.class);  
startService(serviceIntent);
```



# NotificationChannel

- Precisamos criar uma classe a ser executada junto com a aplicação que define um canal de comunicação
- Isso é obrigatório para que as notificações de fato ocorram
- Criamos então, uma classe Java

```
public class AppCanalComunicacao extends Application
{
    @Override
    public void onCreate() {
        super.onCreate();
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
            int importance = NotificationManager.IMPORTANCE_DEFAULT;
            NotificationChannel channel = new NotificationChannel("default", "default", importance);
            channel.setDescription("Canal de Notificação");

            NotificationManager notificationManager = getSystemService(NotificationManager.class);
            notificationManager.createNotificationChannel(channel);
        }
    }
}
```



# AndroidManifest

- Adicionamos a propriedade NAME ao Application (Canal de Comunicação)

```
<application  
    android:name=".AppCanalComunicacao"  
    android:allowBackup="true"
```

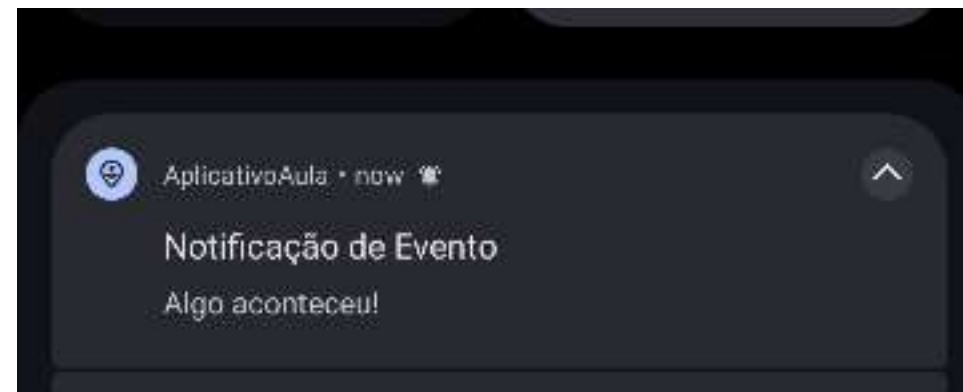
- Adicionamos o Service

```
<service android:name=".NotificationActivity$BackgroundService" />
```



# Notificação

- No nosso exemplo, ao executar, passado 5 segundos, uma notificação vai ser apresentada



# Exercício

- Crie um aplicativo com um botão de contador. Cada vez que o usuário clicar no botão, o contador é incrementado e exibido na tela.
- Quando o contador atingir 5 cliques, o aplicativo deve exibir uma notificação dizendo:
  - "Parabéns! Você clicou 5 vezes!"
- A notificação deve conter:
  - Um título de parabéns.
  - Um texto com a quantidade de cliques.
  - Ação que, ao clicar, abre a MainActivity.
- Dica: Use uma variável `int` contador para controlar os cliques. No quinto clique, chame o método que cria e dispara a notificação.