

EI, PSIU: CRIANDO SUAS PRÓPRIAS NOTIFICAÇÕES



Por Gabriel, Guilherme e Reinaldo

Desenvolvimento de Android - TP



Usando Notificações

As **notificações** destina-se a **informar** os usuários sobre **eventos** no seu aplicativo, tais para:

- Comunicação
- Lembretes de tarefas cronometradas
- Informativos
- etc. etc.. etc...

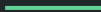


Notificações no android

Está em todos os tipos de apps:
Jogos, Comunicação, Fotografia,
Dados do Sistema e etc

Como as notificações
podem ser notadas

- Mostrando ícones
- Aparecendo na tela de bloqueio
- Piscando led na tela de bloqueio





Corpo da notificação android



App name

App name

Happy New Year!

- `.setSmallIcon()`
- `.setContentTitle()`
- `.setContentText()`

Para emitir a notificação, usa-se o objeto **NotificationManager** chamando **object.notify()**.



Exemplo.java



```
1. Notification.Builder builder =
2.     new Notification.Builder(activity.this)
3.         .setSmallIcon(R.drawable.notification_icon)
4.         .setContentTitle("Exemplo TP")
5.         .setContentText("Hello World!");
6.
7. NotificationManager nm = (NotificationManager)
8.     getSystemService(Context.NOTIFICATION_SERVICE);
9. nm.notify(id, builder.build());
```

Configurações e conteúdo opcionais



- Adicionar ações
- Configurando prévia, controle de mídia e outras funções
- Layout personalizado
- Grupos
- Definir prioridade
- Diferenças entre APIs

Configurações e conteúdo opcionais



- Documentação de referência do [Notification.Builder](#)
- Apesar de serem opcionais, é recomendado adicionar pelo menos uma ação à notificação, quando ela for clicada, direcionando para uma [Activity](#) no aplicativo.



Dentro de uma Notification, a ação é definida por uma PendingIntent que inicia uma Activity no aplicativo.



Exemplo.java

```
1.  ...
2.  Intent rIntent = new Intent(this, AlgumaActivity.class);
3.  TaskStackBuilder sBuilder = TaskStackBuilder.create(this);
4.  sBuilder.addParentStack(AlgumaActivity.class);
5.  sBuilder.addNextIntent(rIntent);
6.  PendingIntent rPendingIntent = sBuilder.getPendingIntent(0,
    PendingIntent.FLAG_UPDATE_CURRENT);
7.  mBuilder.setContentIntent(rPendingIntent);
8.  ...
```

Remoção de notificações



É interessante remover as notificações quando elas não são mais necessárias

- Chamar [setAutoCancel\(\)](#) quando a notificação é criada.
- Chamar [cancelAll\(\)](#), que remove todas as notificações anteriores ou [cancel\(\)](#) com um id para remover uma específica.

Mas não faz sentido usar
notificações se elas somente
aparecem quando o app estiver
sendo usado

Serviços



Qualquer função que seja **executada por um aplicativo** quando ele não está sendo usado é feito por um **processo de segundo plano**, chamado de **serviço**.

Serviços fogem do tema do nosso seminário, mas para que seja possível exibir notificações sem o usuário estar vendo o aplicativo deve-se criar um serviço e passar a ele essa função.

Dúvidas?

