

# Notificações

- User Notifications
  - Toast
  - User Notification Area

# Referências

- DEITEL, Harvey, DEITEL, Paul, DEITEL, Abbey, MORGANO, Michael. Android para programadores: uma abordagem baseada em aplicativos. Porto Alegre: Bookman. 2013.
- Programming Handheld Systems – Prof. Adam Porter of University of Maryland (<https://www.coursera.org/course/android>)
- <http://developer.android.com>
- <https://developers.google.com/university/courses/mobile>
- <http://oreilly.com/training/androidapps>

# Notificações de usuários

- Mensagens para o usuário fora da interface normal
- São destinadas para:
  - Dar feedback para o usuário: *Toasts* e *Dialogs*
  - Notificar a ocorrência de um evento

# *Toast*

- Mensagens transitórias que aparecem na tela do usuário para notifica-lo que algo aconteceu
- Aparecem e desaparecem automaticamente sem a interação do usuário

# Criando uma *Toast Notification*

- Instanciar um objeto *Toast*:

```
Toast.makeText(context, text, duration)
```

- Exibir a mensagem:

```
Toast.show()
```

```
@Override  
public void onClick(View v) {  
    Toast.makeText(getApplicationContext(), "You're toast!", Toast.LENGTH_LONG).show();  
}
```

# *Toast* personalizada

- Pode-se criar *Toasts* personalizadas com layouts em XML e configura-las com: `Toast.setView()`

```
@Override
public void onClick(View v) {

    Toast toast = new Toast(getApplicationContext());

    toast.setGravity(Gravity.CENTER_VERTICAL, 0, 0);
    toast.setDuration	Toast.LENGTH_LONG);

    toast.setView(getLayoutInflater().inflate(R.layout.custom_toast,null));

    toast.show();
}
```

# Notificações na *Notification Area*

- O Android provê uma área de notificação para alertar os usuários acerca da ocorrência de um evento
- Ele também provê uma área que os usuários podem deslizar para baixo para ver informações mais detalhadas dessas notificações

# Arquitetura da Notificação

- Notification
  - Title, Detail, Small Icon
- Notification Area
  - Ticker Text, Small Icon
- Notification Drawer
  - View
  - Action



# *Notification Manager*

- Serviço do sistema que gerencia as notificações
- Envia e cancela as notificações

# Exemplo de código

```
mNotificationIntent = new Intent(getApplicationContext(),
    NotificationSubActivity.class);
mContentIntent = PendingIntent.getActivity(getApplicationContext(), 0,
    mNotificationIntent, Intent.FLAG_ACTIVITY_NEW_TASK);

final Button button = (Button) findViewById(R.id.button1);
button.setOnClickListener(new OnClickListener() {

    @Override
    public void onClick(View v) {

        // Define the Notification's expanded message and Intent:

        mContentView.setTextViewText(R.id.text, contentText + " ("
            + ++mNotificationCount + ")");

        // Build the Notification

        Notification.Builder notificationBuilder = new Notification.Builder(
            getApplicationContext())
            .setTicker(tickerText)
            .setSmallIcon(android.R.drawable.stat_sys_warning)
            .setAutoCancel(true)
            .setContentIntent(mContentIntent)
            .setSound(soundURI)
            .setVibrate(mVibratePattern)
            .setContent(mContentView);

        // Pass the Notification to the NotificationManager:
        NotificationManager mNotificationManager = (NotificationManager) getSystemService(Context.NOTIFICATION_SERVICE);
        mNotificationManager.notify(MY_NOTIFICATION_ID,
            notificationBuilder.build());
    }
});
```