**Conhecimento necessário: Iniciante.**

Se você tem um aparelho à base de Android, você já deve ter visto vários tipos de notificações. São usadas para mostrar ao usuário erros, avisos ou informar que alguma tarefa terminou e a melhor forma é colocá-la em notificação, como uma caixa de alerta, que permite comunicar-se com o usuário fora da interface do programa.

O Android SDK possui três tipos de notificações, o suficiente para cobrir todas necessidades:

Caixas de Alerta

Toasts

Notificações

Nesse tutorial, veremos todas elas.

**Caixas de Alerta**

Criar uma caixa de alerta é bem simples: você pode criar uma AlertDialog usando sua classe, que retorna a mensagem naturalmente. Por exemplo:

*new AlertDialog.Builder(this).setTitle("ScriptBrasil").setMessage("Olá Mundo !").setNeutralButton("OK", null).show();*

"ScriptBrasil" é o título da mensagem, "Olá Mundo !" seu corpo, e "OK" é o texto do botão de confirmação.

setNeutralButton(String, OnClickListener) cria um botão com o título específico e o OnClickListener lida com tudo que queira fazer com o clique.

Você pode dar um valor null porque o AlertDialog sempre fecha quando clica-se no botão; seu OnClickListener poderia cuidar de instruções adicionais somente.

Agora você deve estar pensando o que vem a ser o “neutral” no código, certo? Você pode criar três tipos de botões no total:

setPositiveButton

Mostra uma mensagem que entende-se como confirmação, ex. “Salvar” or “Sim”.

setNeutralButton

Mostra um botão neutro, como "Fechar".

setNegativeButton

Mostra uma mensagem que entende-se como cancelamento ou um simples “Não”.

É importante notar que se você adicionar todos os botões eles se manterão na ordem em que foram adicionados, exemplo:

*new AlertDialog.Builder(this).setTitle("ScriptBrasil").setMessage("Olá Mundo !").setNegativeButton("Cancelar", null).setPositiveButton("OK", null).setNeutralButton("Não", null).show();*

Lembre-se sempre: Crie mensagens com ao menos um botão, do contrário o programa fica travado por não conseguir fechar o aviso.

**Toasts**

Toasts são uma ótima forma de enviar mensagens não-obstrutivas ao usuário, porque diferente das Caixas de Alerta elas não tomam o foco da Activity.

São perfeitas para informar quando não se precisa de muita atenção, como informar que um download terminou.

Como Toasts somem depois de um tempinho e são tímidas por natureza, não é garantido que o usuário a veja. Em bom português: Não use Toasts para mensagens críticas!

Uma linha simples de código é o que é preciso para criar uma Toast:

*Toast.makeText(this, "Olá mundo !", Toast.LENGTH\_LONG).show();*

Novamente, o primeiro parâmetro define em qual contexto ela aparecerá. O segundo é o corpo da mensagem e o terceiro define por quanto tempo ela ficará na tela. Somente Toast.LENGTH\_LONG ou Toast.LENGTH\_SHORT são valores válidos – você não pode especificar em segundos ou milisegundos, use um handler para isso (que já é outro assunto).

**Notificações**

Notificações são as mensagens que ficam no topo da tela no lado contrário ao relógio. São uma mão-na-roda para irritar o usuário sobre algo novo, um e-mail não lido em background, mesmo que ele não esteja no aplicativo de e-mail (para que a notificação seja criada quando fora da Activity é necessário utilizar uma classe Service). O seguinte código é necessário para mostrar uma notificação:

*final int NOTIF\_ID = 1111;*

NotificationManager notifManager = (NotificationManager) getSystemService(NOTIFICATION\_SERVICE);

Notification note = new Notification(R.drawable.icon, "Novo post", System.currentTimeMillis());

PendingIntent intent = PendingIntent.getActivity(this, 0, new Intent(this, Activity.class), 0);

note.setLatestEventInfo(this, "Novo post", "Há um post novo no ScriptBrasil.", intent);

notifManager.notify(NOTIF\_ID, note);

Na linha 1 vê-se o ID da notificação. Quando utilizar mais de uma use ID's diferentes.

Na linha 3 você pode verificar o título, que aparecerá simultaneamente com a notificação, no topo da tela.

Na linha 4 é decidido qual Activity será chamada quando o usuário clicar na notificação (Activity.class nesse caso)

Na linha 5 você pode personalizar o texto que ela exibirá.

Enfim, a linha 6 executa a notificação.

Para fazê-la sumir sem que o usuário precise clicar em "Limpar" usa-se o código:

*notifManager.cancel(NOTIF\_ID);*

Por hoje é só, até mais, pessoal !