Tutorial Android #20 – Alarmes

[12/04/2012](http://www.rafaeltoledo.net/tutorial-android-20-alarmes/)[Android](http://www.rafaeltoledo.net/category/android/), [Linguagem Java](http://www.rafaeltoledo.net/category/linguagem-java/), [Programação Móvel](http://www.rafaeltoledo.net/category/programacao-movel/)[adt](http://www.rafaeltoledo.net/tag/adt/), [alarme](http://www.rafaeltoledo.net/tag/alarme/), [android](http://www.rafaeltoledo.net/tag/android-2/), [eclipse](http://www.rafaeltoledo.net/tag/eclipse/),[java](http://www.rafaeltoledo.net/tag/java/), [xml](http://www.rafaeltoledo.net/tag/xml/)

Olá pessoal! Aqui estamos nós com mais um tutorial de Android para alegria geral da nação. No  tutorial de hoje, vamos ver como criar um alarme para nos avisar a hora do almoço (tudo a ver com o aplicativo de restaurante, não?).

O primeiro passo é criarmos uma forma para o usuário configurar o horário em que ele deseja ser avisado do almoço. Poderíamos definir isto em uma Activity, mas esta opção soa mais como uma configuração. Dessa forma, vamos criar a classe**PreferenciaHorario** no pacote **net.rafaeltoledo.restaurante**, estendendo a classe**DialogPreference**. Logo em seguida explicarei os conceitos principais dela.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95 | package net.rafaeltoledo.restaurante;    import android.content.Context;  import android.content.res.TypedArray;  import android.preference.DialogPreference;  import android.util.AttributeSet;  import android.view.View;  import android.widget.TimePicker;    public class PreferenciaHorario extends DialogPreference {        private int ultimaHora = 0;      private int ultimoMinuto = 0;      private TimePicker picker = null;        public static int obterHora(String tempo) {          String[] fragmentos = tempo.split(":");          return Integer.parseInt(fragmentos[0]);      }        public static int obterMinuto(String tempo) {          String[] fragmentos = tempo.split(":");          return Integer.parseInt(fragmentos[1]);      }        public PreferenciaHorario(Context contexto) {          this(contexto, null);      }        public PreferenciaHorario(Context contexto, AttributeSet atributos) {          this(contexto, atributos, 0);      }        public PreferenciaHorario(Context contexto, AttributeSet atributos, int estilo) {          super(contexto, atributos, estilo);            setPositiveButtonText("Definir");          setNegativeButtonText("Cancelar");      }        @Override      protected View onCreateDialogView() {          picker = new TimePicker(getContext());          return picker;      }        @Override      protected void onBindDialogView(View view) {          super.onBindDialogView(view);            picker.setCurrentHour(ultimaHora);          picker.setCurrentMinute(ultimoMinuto);      }        @Override      protected void onDialogClosed(boolean positiveResult) {          super.onDialogClosed(positiveResult);            if (positiveResult) {              ultimaHora = picker.getCurrentHour();              ultimoMinuto = picker.getCurrentMinute();                String tempo = String.valueOf(ultimaHora) + ":" + String.valueOf(ultimoMinuto);                if (callChangeListener(tempo)) {                  persistString(tempo);              }          }      }        @Override      protected Object onGetDefaultValue(TypedArray a, int index) {            return a.getString(index);      }        @Override      protected void onSetInitialValue(boolean restorePersistedValue,              Object defaultValue) {          String tempo = null;            if (restorePersistedValue) {              if (defaultValue == null) {                  tempo = getPersistedString("00:00");              } else {                  tempo = getPersistedString(defaultValue.toString());              }          } else {              tempo = defaultValue.toString();          }            ultimaHora = obterHora(tempo);          ultimoMinuto = obterMinuto(tempo);      }  } |

Bastante coisa, não? Vamos por partes.

Os métodos **obterHora()** e **obterMinuto()** servem para extrair a parte inteira do horário que será armazenado como uma string “**00:00**“. Temos três versões do construtor da classe, que no final sempre referenciam o terceiro. Isso é devido à superclasse. Ainda no construtor, definimos os nomes dos botões na janela de configuração de horário.**onCreateDialogView()** devolve um objeto **View** com a tela criada. Poderíamos aqui definirmos um layout, mas simplesmente devolvemos um widget **TimePicker**. O método**onBindDialogView()** é chamado após o **onCreateDialogView()** é encarregado de preencher a caixa de diálogo. **onDialogClose()**, como o próprio nome diz, é chamado quando a janelinha é encerrada. Caso o usuário pressione o botão de confirmar (condição**positiveResult**), o valor é armazenado em **SharedPreferences**. O método**onGetDefaultValue()** é utilizado para a conversão interna do Android para o tipo do objeto. Por último, o método **onSetInitialValue()**, como o próprio nome diz, atribui um valor padrão. Ele verifica se há algum valor já salvo, ou padrão ou então atribui **00:00**.

O próximo passo é adicionar a opção para a configuração do alarme pelo usuário. Dessa forma, edite o arquivo **preferencias.xml** da seguinte forma:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | <PreferenceScreen xmlns:android="<http://schemas.android.com/apk/res/android>">      <ListPreference          android:key="listagem"          android:title="Modo de Listagem"          android:summary="Escolha o modo de listagem a ser utilizado"          android:entries="@array/nomes\_ordenacao"          android:entryValues="@array/opcoes\_ordenacao"          android:dialogTitle="Escolha o modo de listagem" />      <CheckBoxPreference          android:key="alarme"          android:title="Tocar Alarme no Almoço"          android:summary="Marque se deseja ser informado sobre a hora do almoço"/>      <net.rafaeltoledo.restaurante.PreferenciaHorario          android:key="horario\_alarme"          android:title="Horário do Alarme do Almoço"          android:defaultValue="12:00"          android:summary="Configure seu horário desejado para o alarme"          android:dependency="alarme"/>  </PreferenceScreen> |

A primeira opção adicionada, do tipo **CheckBoxPreference** não tem muito segredo… a segunda, foi a que definimos na classe **PreferenciaH0rario**. Configuramos seu valor padrão para **12:00** e definimos que ela depende da opção **alarme**, ou seja, ela só estará habilitada caso **alarme** também esteja habilitada.

Neste projeto vamos utilizar o **AlarmManager** para gerenciar o nosso alarme. Porém, ele tem uma falha: toda vez que o celular é desligado, ao ligar novamente os alarmes não são configurados. Para resolver isso, vamos criar a classe **ReceptorBoot** para realizar essa configuração toda vez que o sistema for ligado. Crie-a no pacote**net.rafaeltoledo.restaurante**.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | package net.rafaeltoledo.restaurante;    import android.content.BroadcastReceiver;  import android.content.Context;  import android.content.Intent;    public class ReceptorBoot extends BroadcastReceiver {        @Override      public void onReceive(Context context, Intent intent) {        }  } |

A tarefa do **ReceptorBoot** será realizado no método **onReceive()**. Por enquanto, coloque-o pra descansar. Já já voltamos nele.

Prosseguindo, precisamos adicionar o nó **<receiver>** no arquivo**AndroidManifest.xml** para que ele possa atuar no boot. Adicione-o ao final do nó**application**.

|  |  |
| --- | --- |
| 39  40  41  42  43  44 | <receiver android:name=".ReceptorBoot"      android:enabled="false">      <intent-filter>          <action android:name="android.intent.action.BOOT\_COMPLETED"/>      </intent-filter>  </receiver> |

Além disso, adicione também a permissão para obter o sinal de boot completo do sistema.

|  |  |
| --- | --- |
| 12 | <uses-permission android:name="android.permission.RECEIVE\_BOOT\_COMPLETED"/> |

Precisamos agora tratar as preferências do usuário para configurar o alarme. Quando o usuário ativar o checkbox do alarme, precisamos ativar o alarme no tempo selecionado. Quando o usuário modificar o alarme (por exemplo, para **11:00**), devemos criar um novo alarme com o **AlarmManager**. Se ele desativar, precisamos cancelar o alarme existente. E, por fim, em um processo de boot, se o alarme estiver selecionado, precisamos criá-lo.

Para fazer todo esse trabalho, adicione os seguintes métodos na classe **ReceptorBoot**. Para corrigir os **imports**, só lembrar do **Ctrl + Shift + O**.

|  |  |
| --- | --- |
| 20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47 | public static void configurarAlarme(Context contexto) {      AlarmManager gerenciador = (AlarmManager) contexto.getSystemService(Context.ALARM\_SERVICE);      Calendar cal = Calendar.getInstance();      SharedPreferences preferencias = PreferenceManager.getDefaultSharedPreferences(contexto);      String horario = preferencias.getString("horario\_alarme", "12:00");        cal.set(Calendar.HOUR\_OF\_DAY, PreferenciaHorario.obterHora(horario));      cal.set(Calendar.MINUTE, PreferenciaHorario.obterMinuto(horario));      cal.set(Calendar.SECOND, 0);      cal.set(Calendar.MILLISECOND, 0);        if (cal.getTimeInMillis() < System.currentTimeMillis()) {          cal.add(Calendar.DAY\_OF\_YEAR, 1);      }        gerenciador.setRepeating(AlarmManager.RTC\_WAKEUP, cal.getTimeInMillis(),              AlarmManager.INTERVAL\_DAY, obterIntentPendente(contexto));  }    public static void cancelarAlarme(Context contexto) {      AlarmManager gerenciador = (AlarmManager) contexto.getSystemService(Context.ALARM\_SERVICE);      gerenciador.cancel(obterIntentPendente(contexto));  }    private static PendingIntent obterIntentPendente(Context contexto) {      Intent i = new Intent(contexto, ReceptorAlarme.class);      return PendingIntent.getBroadcast(contexto, 0, i, 0);  } |

Também atualize o método **onReceive()**:

|  |  |
| --- | --- |
| 15  16  17  18 | @Override  public void onReceive(Context context, Intent intent) {      configurarAlarme(context);  } |

Bem, no código listado acima, primeiramente, ao receber o sinal do boot (método**onReceive()**), configuramos o alarme, através do método **configurarAlarme()**. Neste método, obtemos o **AlarmManager**, e obtemos as preferências do usuário para o alarme (se existirem), e a montamos em um objeto do tipo **Calendar**. Caso alarme seja anterior ao horário atual, adicionamos um dia a ele e configuramos para repeti-lo diariamente. Já no método **cancelarAlarme()**, cancelamos o alarme vinculado ao contexto, obtendo o**AlarmManager** e obtendo um objeto **PendingIntent** (como se fosse uma tarefa pendente) com o método **obterIntentPendente()**.

No código que temos até agora, o alarme só é armado na inicialização do sistema. Para que ele funcione da maneira como desejamos, precisamos adicionar alguns método a classe **EdicaoPreferencias**:

|  |  |
| --- | --- |
| 22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58 | @Override  protected void onResume() {      super.onResume();        preferencias = PreferenceManager.getDefaultSharedPreferences(this);      preferencias.registerOnSharedPreferenceChangeListener(onChange);  }    @Override  protected void onPause() {      preferencias.unregisterOnSharedPreferenceChangeListener(onChange);        super.onPause();  }    OnSharedPreferenceChangeListener onChange = new SharedPreferences.OnSharedPreferenceChangeListener() {        public void onSharedPreferenceChanged(SharedPreferences sharedPreferences,              String key) {          if ("alarme".equals(key)) {              boolean habilitado = preferencias.getBoolean(key, false);              int flag = (habilitado ? PackageManager.COMPONENT\_ENABLED\_STATE\_ENABLED : PackageManager.COMPONENT\_ENABLED\_STATE\_DISABLED);              ComponentName componente = new ComponentName(EdicaoPreferencias.this, ReceptorBoot.class);                getPackageManager().setComponentEnabledSetting(componente, flag, PackageManager.DONT\_KILL\_APP);                if (habilitado) {                  ReceptorBoot.configurarAlarme(EdicaoPreferencias.this);              } else {                  ReceptorBoot.cancelarAlarme(EdicaoPreferencias.this);              }          } else if ("horario\_alarme".equals(key)) {              ReceptorBoot.cancelarAlarme(EdicaoPreferencias.this);              ReceptorBoot.configurarAlarme(EdicaoPreferencias.this);          }      }  }; |

Lembre-se também de adicionar o membro privado da classe chamado **preferencias**:

|  |  |
| --- | --- |
| 13 | SharedPreferences preferencias = null; |

O que nos falta fazer é criar um receptor que exiba o alarme na tela quando o alarme disparar. Para isso, primeiramente crie o arquivo **alarme.xml** na pasta **res/layout**:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | <?xml version="1.0" encoding="utf-8"?>  <TextView xmlns:android="<http://schemas.android.com/apk/res/android>"      android:layout\_width="wrap\_content"      android:layout\_height="wrap\_content"      android:text="Hora do almoço!"      android:textSize="30sp"      android:textStyle="bold" /> |

Bastante simples, ele simplesmente exibirá bem grande na tela **Hora do almoço!**. Agora vamos criar a Activity que exibirá o aviso propriamente dito. Crie a classe **AlarmeActivity**no pacote **net.rafaeltoledo.restaurante**:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | package net.rafaeltoledo.restaurante;    import android.app.Activity;  import android.os.Bundle;    public class AlarmeActivity extends Activity {        @Override      protected void onCreate(Bundle savedInstanceState) {          super.onCreate(savedInstanceState);          setContentView(R.layout.alarme);      }  } |

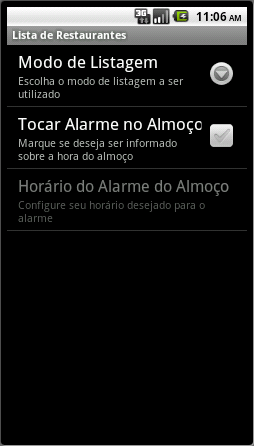
Crie também uma classe chamada **ReceptorAlarme** que será encarregada de iniciar a**AlarmeActivity**.

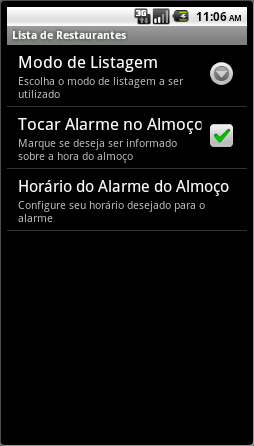
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | package net.rafaeltoledo.restaurante;    import android.content.BroadcastReceiver;  import android.content.Context;  import android.content.Intent;    public class ReceptorAlarme extends BroadcastReceiver {        @Override      public void onReceive(Context context, Intent intent) {          Intent i = new Intent(context, AlarmeActivity.class);          i.setFlags(Intent.FLAG\_ACTIVITY\_NEW\_TASK);          context.startActivity(i);      }  } |

Encerrando (ufa!), falta somente adicionarmos esse último receptor no**AndroidManifest.xml**. Adicione-o no fim do nó **application**.

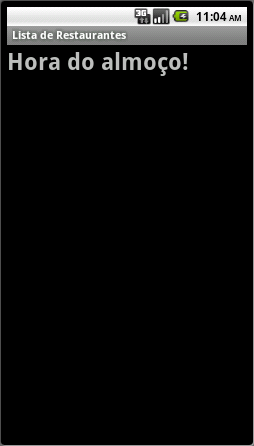
|  |  |
| --- | --- |
| 45  46 | <receiver android:name=".ReceptorAlarme">  </receiver> |

E é isso!

[](http://i0.wp.com/www.rafaeltoledo.net/wp-content/uploads/2012/04/android-tutorial-2.png)

[](http://i0.wp.com/www.rafaeltoledo.net/wp-content/uploads/2012/04/android-tutorial-3.png)

[](http://i2.wp.com/www.rafaeltoledo.net/wp-content/uploads/2012/04/android-tutorial-4.png)

[](http://i2.wp.com/www.rafaeltoledo.net/wp-content/uploads/2012/04/android-tutorial-1.png)